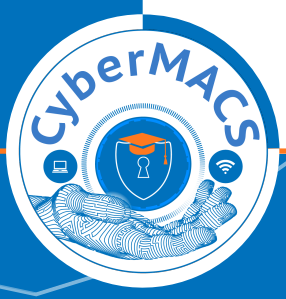
WEB PROCEEDINGS

International Applied Cybersecurity Conference

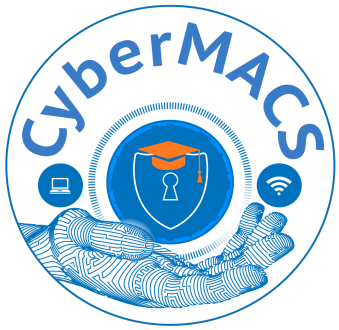
11-14 October 2025
Ohrid, Republic of N. Macedonia



International Applied Cybersecurity Conference
IACyC 2025

11 October – 14 October 2025
Ohrid, R. N. Macedonia

Web Proceedings



Editors

Prof. Dr. Vesna Dimitrova

Prof. Dr. Aleksandra Popovska - Mitrovikj

Assoc. Prof. Dr. Hristina Mihajloska Trpcheska

Asst. Prof. Dr. Boban Joksimoski

Technical Editors

Mila Dodevska

Zorica Karapancheva

Jovan Simonoski

Publisher: Faculty of Computer Science and Engineering in Skopje

Publisher's headquarters: st. Rugjer Boskovic 16, Skopje, R. N. Macedonia

Place of publication: Skopje, R. N. Macedonia

Date of publication: 2026

CIP - Каталогизација во публикација
Национална и универзитетска библиотека "Св. Климент Охридски", Скопје

004.056(062)

INTERNATIONAL applied cybersecurity conference IACyC (2025 ; Ohrid) Web Proceedings [[Електронски извор]] / International applied cybersecurity conference, IACyC 2025, 11 October–14 October 2025 Ohrid, R. N. Macedonia ; [editors Vesna Dimitrova ... и др.]. - Текст во PDF формат , содржи 190 стр. ; табели, графикони. - Skopje : Faculty of computer science and engineering, 2025. - 190 стр. : илустр. ; 25 см

Начин на пристапување (URL): https://proceedings.iacyc.finki.ukim.mk/ (Слободен пристап). - Наслов преземен од екранот. - Опис на изворот на ден 21.04.2026. - Други уредници: Aleksandra Popovska-Mitrovikj, Hristina Mihajloska Trpcheska, Boban Joksimoski. - Библиографија кон трудовите

ISBN 978-608-4699-23-1

а) Сајбер-безбедност – Собири

COBISS.MK-ID 68676613

Global Honorary Chairs

Prof. Dr. Biljana Angelova, Ss. Cyril and Methodius University in Skopje, North Macedonia

Prof. Dr. Sondan Durukanoğlu Feyiz, Kadir Has University, Istanbul, Turkey

Dr. rer. pol. Thorsten Bagschik, SRH Berlin University of Applied Sciences, Berlin, Germany

Global Chairs

Prof. Dr. Hasan Dağ, Kadir Has University, Istanbul, Turkey

Prof. Dr. Reiner Creutzburg, SRH Berlin University of Applied Sciences, Berlin, Germany

Prof. Dr. Vesna Dimitrova, Ss. Cyril and Methodius University in Skopje, North Macedonia

Steering Committee

Prof. Dr. Hasan Dağ, Kadir Has University, Istanbul, Turkey

Prof. Dr. Reiner Creutzburg, SRH Berlin University of Applied Sciences, Berlin, Germany

Prof. Dr. Vesna Dimitrova, Ss. Cyril and Methodius University in Skopje, North Macedonia

Prof. Dr. Ivan Chorbev, Ss. Cyril and Methodius University in Skopje, North Macedonia

Prof. Dr. Boro Jakimovski, Ss. Cyril and Methodius University in Skopje, North Macedonia

Geritt Tamm, SRH Berlin University of Applied Sciences, Berlin, Germany

Assoc. Prof. E. Fatih Yetkin, Kadir Has University, Istanbul, Turkey

Prof. Guy Gogniat, Université Bretagne Sud (UBS), Lorient, France

Prof. Tegawendé BIssyande, University of Luxembourg, Luxembourg

Prof. Jean-Michel Dricot, Université Libre de Bruxelles, Brussels, Belgium

Local Organizing Committee

Prof. Dr. Vesna Dimitrova, Ss. Cyril and Methodius University in Skopje, North Macedonia

Assoc. Prof. Dr. Hristina Mihajloska Trpcheska, Ss. Cyril and Methodius University in Skopje, North Macedonia

Prof. Dr. Aleksandra Popovska-Mitrovikj, Ss. Cyril and Methodius University in Skopje, North Macedonia

Asst. Prof. Dr. Boban Joksimoski, Ss. Cyril and Methodius University in Skopje, North Macedonia

Prof. Dr. Magdalena Kostoska Gjorcheska, Ss. Cyril and Methodius University in Skopje, North Macedonia

Asst. Prof. Dr. Aleksandar Stojmenski, Ss. Cyril and Methodius University in Skopje, North Macedonia

Dr. Elissa Mollakuqe, Technical University of Darmstadt Darmstadt, Germany

Ebru Dilan, Kadir Has University, Istanbul, Turkey

Mert İlhan Ecevit, Kadir Has University, Istanbul, Turkey

Igor Cvetanovski, Ss. Cyril and Methodius University in Skopje, North Macedonia

Technical Committee

Mila Dodevska, Ss. Cyril and Methodius University in Skopje, North Macedonia

Zorica Karapancheva, Ss. Cyril and Methodius University in Skopje, North Macedonia

Jovan Simonoski, Ss. Cyril and Methodius University in Skopje, North Macedonia

Preface

International Applied Cybersecurity Conference - IACyC 2025, held from October 11 to 14, 2025, at the Congress Center and Hotel Metropol in Ohrid, North Macedonia, is the first IACyC conference, organized within the framework of the CyberMACS project (Master's Programme in Applied Cybersecurity, Erasmus Mundus Joint Master Degree). The project is a collaboration between three academic partner institutions (Kadir Has University in Istanbul, Turkey, SRH University of Applied Sciences Heidelberg in Berlin, Germany and Ss. Cyril and Methodius University in Skopje, North Macedonia), including the Faculty of Computer Science and Engineering at Ss. Cyril and Methodius University in Skopje, which proudly hosts this event. The main mission of the project and the conference is to promote high-quality international education in the field of cybersecurity, attract talented students worldwide, and foster academic and industry collaboration.

IACyC 2025 is envisioned as a platform for knowledge exchange, enabling students, alumni, academic researchers, and industry professionals connected with Erasmus Mundus programs to share ideas, innovations, and experiences in applied cybersecurity.

The conference program includes a Summer School on October 13 – 14, featuring workshops and lectures by distinguished speakers from academia and industry. We are also honored to host the defense of master's theses by the first generation of CyberMACS students, who will be awarded a joint degree from UKIM (N. Macedonia) and Kadir Has University (Turkey). Their achievements will be celebrated during a special graduation ceremony held during the conference.

Comprising 24 distinct contributions, the Web Proceedings captures the innovative and collaborative drive of the international cybersecurity community. The featured papers span a broad spectrum of topics, highlighting new discoveries, real-world applications, and visionary strategies in the field.

We would like to express our sincere gratitude to all the authors for their valuable contributions, and for sharing their insights and expertise. Your work is instrumental in advancing the field and shaping the future of applied cybersecurity. On behalf of the organizing committee and project coordinators, we thank you for being part of IACyC 2025 and hope you find the conference enriching and inspiring.

October, 2025
Ohrid, N. Macedonia

Prof. Dr. Vesna Dimitrova
Chair of IACyC 2025

Web Proceedings

Table of Contents

A Study of Zero Trust Security Mechanisms in Microservices Architecture

1st Nitika Poudel
SRH Heidelberg
University of Applied Sciences
Berlin, Germany
nitika.professional@gmail.com

2nd Klaus Schwarz
SRH Heidelberg
Berlin, Germany
schwarz@posteo.de

3rd Reiner Creutzburg
SRH Heidelberg
Berlin, Germany
reiner.creutzburg@gmail.com

4th Oğuzhan Ceylan
Kadir Has University
Istanbul, Turkey
oguzhan.ceylan@khas.edu.tr

***Abstract*—This research explores the implementation of Zero Trust security using the Istio service mesh in a microservices-based application deployed on a resource-constrained distributed edge cluster. While Zero Trust implementations offers enhanced security, their performance impact in edge environments remains underexplored. To address this gap, three configurations were evaluated under controlled load: a baseline with no security, a setup with mTLS and JWT authentication, and a full Zero Trust configuration including Attribute Based Access Control with OPA. Performance and resource utilization metrics were analyzed under low, medium, and high load conditions. Results showed that while the baseline scaled well with minimal latency, introducing Zero Trust mechanisms particularly OPA-based authorization significantly increased performance overhead, especially under higher loads. The findings highlight the trade-off between security and performance in resource-constrained edge environments and underscore the need for balanced design decisions when adopting Zero Trust in microservices architectures at the edge.**

***Index Terms*—Zero Trust Architecture, Kubernetes, microservices security, mTLS, OPA, JWT, Istio service mesh, Attribute based access control, cybersecurity, Edge Computing, Distributed System**

I. Introduction

In recent years, the monolithic architectural paradigm has been replaced by microservices, driven by the demand for scalable, modular, and resilient cloud-native applications. The distributed nature of microservices expands the attack surface, exposing multiple entry points and increasing the risk of lateral movement by malicious actors. Traditional perimeter-based security is inadequate for such dynamic environments. Zero Trust Architecture (ZTA) offers a modern security paradigm that addresses this challenge by enforcing the principle of "never trust, always verify". Service mesh with its automation and traffic management functionality makes it easier to enforce zero trust policies in a microservice architecture which is not available by default with Kubernetes. While past research has focused largely on cloud deployments, the use of Zero Trust in edge computing remains underexplored. In resource-constrained edge environments, where computing is closer to the data source, the trade-off between security and performance becomes a critical consideration.

A. Research Questions

- How can zero trust security be implemented in a microservices based testbed edge computing environment?
- How does Zero Trust security implementation with mTLS, JWT authentication, and Attribute-Based Access Control (ABAC) impact the performance and resource utilization of a microservice-based application deployed on a testbed distributed edge cluster?
- How are the performance and resource utilization metrics impacted across various level of security configurations under varying load in a resource constrained microservices deployment on a testbed distributed edge cluster?

II. Literature Review

Traditional security systems operating on perimeter based security using firewalls and access control mechanisms rely mostly on implicit trust assumptions on various system components and resources [1]. To adapt to the dynamic attack surface of modern applications and services, zero trust tends to move defenses towards users, assets and resources unlike traditional security measures which focus on securing static attack surfaces such as network based enterprise perimeters. Zero trust focuses on evaluating trust explicitly rather than implicitly trusting an entity based on their physical location or network. Zero trust architecture has authentication and authorization at its core and is designed to prevent lateral movement through continuous trust evaluation. It performs authentication and authorization at a granular level i.e for every access request and provides least privilege needed to perform any transaction [2]. There is a need of secure and granular inter service authentication and authorization which can be achieved through the use of mTLS protocol and role based or attribute based access control mechanisms [3].According to NIST SP 800-204B [4], attribute-based access control (ABAC) is an authorization system that evaluates user access requests based on user, application, and environment attributes, as well as policies that incorporate these attributes. (Simone et al.,2021) implemented zero trust using a service mesh in a multi cloud environment and analyzed the performance overhead of implementation. It was found that the use of Istio could increase CPU and memory usage depending on the cloud environment and configurations when compared to a baseline multi cloud implementation without Istio [5]. Another study also analyzed the performance overhead of using ISTIO service mesh with microservices applications in the Kubernetes cluster which

demonstrated that the use of Istio can result in significant overhead in latency and CPU cycles [6]. Service mesh can be a promising solution for managing the complexities of microservice architectures through automation, security and effective traffic management. Service mesh in a multi cloud infrastructure can introduce significant overhead (latency, CPU, and memory), but programmable kernel tech like eBPF can mitigate some of it [7]. The study used role-based access control for authorization. (Viswanathan et al., 2024) applied zero trust security for web applications deployed in a Kubernetes cluster by using mTLS and JWT authentication. The authors introduced a zero trust security model which enforced zero trust in web applications with continuous identity verification and dynamic access control and found that zero trust implementation resulted in minimized unauthorized access and lateral movement [8]. Another study investigated about the various deployment policies for microservices on edge computing environments and provided insights on the performance aspect of those deployments [9]. (Hossain et al., 2023) performed a comprehensive analysis on the benefits, applications and role of microservices in edge computing environments. The study described and presented the benefits of using microservices in edge computing environments. Performance improvement, improved resource utilization and latency reduction were some of the listed benefits of using microservices in edge computing environments [10]. Another study was done to evaluate the various aspects of performance benchmarking while using a service mesh in edge computing based deployments but did not cover the zero trust security aspects. The study focused on performance impact and the complexity of architecture for the on-premise Kubernetes cluster with Istio service mesh. The study showed that a significant impact of envoy was observed on the Kubernetes stack in terms of performance and bottlenecks [11]. A study was done to analyze and evaluate the performance overheads of microservices deployments in a Kubernetes cluster with Istio Service Mesh using Linux-perf and wrk2. A comprehensive analysis was performed for the CPU overheads and performance metrics for a cloud computing based scenario and did not address the edge computing based scenarios [6]. A master thesis study was performed to evaluate the performance versus security tradeoff of zero trust implementation on Kubernetes using the service meshes: Istio, Linkerd and Cilium. It also compared the performance overheads with varying loads for the three services meshes implementing zero trust. The study was performed on an experiment setup built on the Google Kubernetes Engine (GKE) [12].

III. Methodology

One of the major principles of zero trust architecture is "never trust, always verify". It means that no user or entity should be trusted implicitly based on their physical location inside a network. Zero trust is based on the concept of developing explicit trust with continuous authentication and authorization. Service mesh like Istio, can be used by organizations to achieve zero trust for their microservices applications. Istio provides functionalities like peer authentication, request authentication, ingress gateway security and OPA external authorization for fine grained access control. Istio allows mutual authentication and encrypted communication between all services through mTLS (mutual Transport Layer Security). It ensures both communicating services are who they claim to be before starting communication. With mTLS STRICT mode enabled, any plain text communication between two services is rejected. Istio provides request authentication functionality which ensures that any entity trying to access a resource within a mesh has verified and valid credentials and it can be achieved through JWT validation per request. In addition to the available authorization functionality, istio also allows the use of external authorization services like OPA for access control policy evaluation. OPA can be deployed as a sidecar with an external authorization filter in the istio service mesh. Deploying OPA as a sidecar helps keep it close to the workload for low-latency decisions.

A. System Design

For the purpose of experimentation, a distributed edge cluster was formed over a tailscale network. Table I shows the hardware requirements for the experimental setup. The testbed distributed edge cluster comprised 3 Raspberry Pi machines and 2 Ubuntu Virtual Machines on a macOS host. An Ethernet switch connected the Raspberry Pi machines to a MacBook. The MacBook acted as a gateway router for the local Raspberry Pi network through the Mac's internet sharing capability. The local network of 3 Raspberry Pi machines and 2 Ubuntu Virtual Machines was connected over Tailscale to form a private mesh network. Figure 1, represents the network diagram for the cluster formation.

Device	RAM	Hard Disk	Role
Ubuntu VM on macOS host	8GB	50GB	K3s Control Plane Node
Raspberry Pi 4	8GB	32GB	K3s Worker Node
Raspberry Pi 4	4GB	32GB	K3s Worker Node
Ubuntu VM on macOS host	4GB	20GB	K3s Monitoring Node
Raspberry Pi 4	2GB	16GB	Load Testing Machine

TABLE I
DEVICE CONFIGURATION AND ROLES IN THE K3S CLUSTER

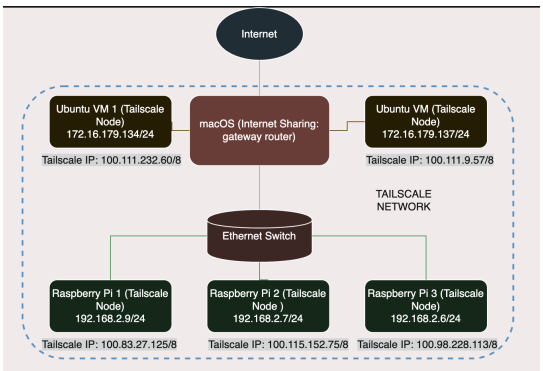


Fig. 1. System design diagram

B. Experiment Design

The research experiment setup was composed of a k3s cluster with a control node and 3 worker nodes. The experiment was designed to test the performance metrics of the k3s cluster over three different configurations and 3 varying loads (20RPS, 60RPS, 100RPS). The main goal was to test the performance of the system with respect to scalability and how additional layers of zero trust can impact the overall performance. Following are the three different configurations:

- **Baseline:** The baseline deployment configuration consisted of a simple deployment of the bookinfo application without istio-proxy sidecars and with no external request authentication. The bookinfo application was exposed by using Istio ingress gateway.
- **Istio + JWT:** The second configuration consisted of the bookinfo application deployed with istio-proxy sidecars and exposed by using Istio ingress gateway. This configuration also enabled peer authentication in STRICT mode using mTLS and request authentication using JWT validation. Keycloak was set up as an OpenID Connect (OIDC) provider which issued JWT tokens. Istio proxy sidecar on the application pod intercepts the HTTP request, checks the validity of the keycloak assigned JWT token and allows request into the mesh only if the token is valid.
- **Istio + JWT + OPA:** The third configuration consisted of the bookinfo application deployed with istio-proxy sidecars, JWT request authentication, mTLS peer authentication and an additional external authorization of policies with OPA. The OPA was injected as a sidecar in the application pod. The Http request is intercepted by the istio-proxy sidecar which validates the JWT token. If the JWT token is valid, the request is then forwarded to the OPA sidecar which either allows or denies the request based on the rego policy. In this configuration, attribute based access control policies are implemented.

C. Data Collection

The performance metrics were collected by performing a load test with k6 on all three configurations under varying loads (20RPS, 60RPS,100RPS). The same k6 script was used to test on each configuration to maintain reliability of data. Each k6 load test (low, medium and high) was run 5 times for each configuration to generate enough replicates. Following metrics were scraped from prometheus and grafana: P95 Latency Cluster CPU Usage(%), Cluster Network Throughput(Mbps), Pod Memory Usage(MB), Istio-Proxy memory usage (MB), Pod CPU Usage(cores), Istio-proxy CPU usage(cores), Memory Usage(%), OPA cpu usage, OPA memory usage.

D. Data Analysis

- **Descriptive Analysis:** A preliminary descriptive analysis was performed on the collected data to summarize and visualize the information represented by the data. Grouped bar graphs were generated for each performance and resource utilization metrics to visualize and analyze trends and changes in metrics across the three configurations: baseline, zero trust with Istio+JWT, and zero trust with Istio+JWT+OPA. This gave a clear picture of how the addition of each new security layer affected the system performance and resource utilization.
- **Kruskal-Wallis Test:** The Kruskal-Wallis Test is a non parametric test which can be used instead of ANOVA if the assumptions for ANOVA fails. It can be used to determine if the differences between the medians of three or more groups are statistically significant.
  Null hypothesis: There is no statistically significant difference in the distribution of a performance/resource utilization metric value among different configurations (baseline, istio+jwt, istio+jwt+opa) under the same load.
  Alternative hypothesis: There is statistically significant difference in the distribution of a performance/resource utilization metric value among different configurations (baseline, istio+jwt, istio+jwt+opa) under the same load.
- **Dunn's post hoc test:** The Dunn's test is a non-parametric post hoc test usually performed after the kruskal-wallis test results in significant difference among groups. Dunn's test performs pairwise comparisons among the various combinations of groups. For each pair, it compares the average ranks of values of each group. To avoid the risk of error due to false positives, it applies correction measures like bonferroni correction to adjust the p-values. It finally calculates the test statistic and a p-value for each pair. If the p-value for any pair is less than 0.05 then it means that the pair significantly differs from each other.

IV. Findings

A. System Perspective

The dataset consisted of performance metrics data for each configuration type (baseline, ISTIO+JWT and ISTIO+OPA+JWT) under three different loads (20 RPS, 60RPS and 100RPS). A total of 405 rows of data were collected for all the combinations of three configuration types and three load types. A preliminary descriptive analysis was performed on the dataset to gain insights on existing trends and relationships within the dataset.

P95 Latency: It can be observed from figure 2 that under low load (20 RPS), latency increased moderately across configurations: from 47.89ms (baseline) to 78.38ms (Istio+JWT) and 95.74ms (Istio+JWT+OPA). However, as load increased (60 to 100 RPS), latency spiked significantly, especially with OPA, indicating that the system struggles to maintain performance under high load when complex security layers are added. Table II shows the results of Kruskal-Wallis test for p95 latency for all three configurations under all three load types. The Kruskal-Wallis test showed statistically significant differences in P95 latency across Baseline, Istio+JWT, and OPA configurations under low, medium, and high load, with all p-values less than 0.05, indicating the differences are

not due to chance. Dunn's post hoc tests further confirmed that each pairwise comparison (Baseline vs Istio, Istio vs OPA, Baseline vs OPA) under all load levels is statistically significant, meaning each configuration has a measurable and distinct impact on latency performance.

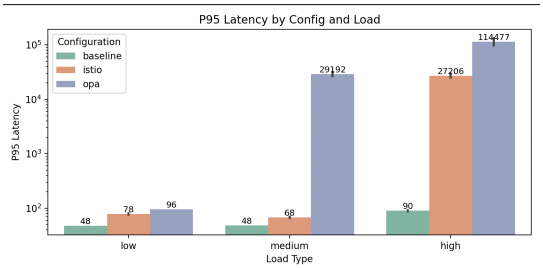


Fig. 2. Grouped bar graph demonstrating P95 Latency Comparison Across Baseline, Istio, and OPA Configurations Under Low, Medium, and High Load Conditions

Metric	Load	H	p-value
P95_Latency	High	117.95	2.44×10^{-26}
P95_Latency	Low	119.43	1.17×10^{-26}
P95_Latency	Medium	116.92	4.09×10^{-26}

TABLE II
KRUSKAL-WALLIS TEST RESULT FOR P95 LATENCY DISTRIBUTION ACROSS BASELINE, ISTIO, AND OPA CONFIGURATIONS UNDER LOW, MEDIUM, AND HIGH LOAD CONDITIONS

Cluster Network Throughput: Cluster network throughput, measured in Mbps and scraped from Prometheus, reflects total system network traffic and helps identify performance bottlenecks. As shown in Figure 3, throughput slightly increased from Baseline to Istio+JWT, but dropped with the addition of OPA, a trend consistent across low (20 RPS), medium (60 RPS), and high (100 RPS) load conditions.

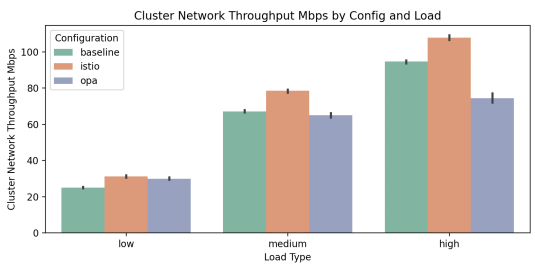


Fig. 3. Grouped bar graph demonstrating Cluster Network throughput Comparison Across Baseline, Istio, and OPA Configurations Under Low, Medium, and High Load Conditions

Table III shows that the p-values for cluster network throughput under low, medium, and high loads are all below 0.05, indicating statistically significant differences across configurations. Dunn's post hoc test results confirmed that all configuration pairs (Baseline vs Istio, Baseline vs OPA, OPA vs Istio) differ significantly under each load level, meaning the variations in throughput are not due to random chance.

Metric	Load	H	p-value
Network_Throughput	High	117.5936645	2.92×10^{-26}
Network_Throughput	Medium	91.85994873	1.13×10^{-20}
Network_Throughput	Low	102.5118764	5.49×10^{-23}

TABLE III
KRUSKAL-WALLIS TEST RESULT FOR CLUSTER NETWORK THROUGHPUT DISTRIBUTION ACROSS BASELINE, ISTIO, AND OPA CONFIGURATIONS UNDER LOW, MEDIUM, AND HIGH LOAD CONDITIONS

Cluster Memory Usage: Figure 4 shows that under low and medium loads, Istio+JWT and OPA configurations consumed slightly less cluster memory compared to the baseline. However, under high load (100 RPS), memory usage increased slightly for Istio+JWT and OPA, surpassing the baseline configuration.

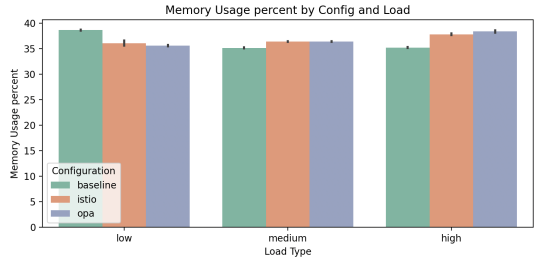


Fig. 4. Bar Graph of cluster memory usage percentage comparison across Baseline, Istio, and OPA Configurations Under Low, Medium, and High Load Conditions

Table IV shows that cluster memory usage varies significantly across configurations under all load levels, as all p-values are below 0.05, leading to rejection of the null hypothesis for Kruskal-Wallis test. Dunn's Post hoc analysis revealed that baseline differs significantly from both Istio and OPA configurations, while Istio and OPA show no significant difference, indicating that adding OPA on top of Istio does not notably impact cluster memory usage.

Metric	Load	H	p-value
Memory_Usage_percent	High	93.61	4.72×10^{-21}
Memory_Usage_percent	Medium	85.24	3.09×10^{-19}
Memory_Usage_percent	Low	59.76	1.06×10^{-13}

TABLE IV
KRUSKAL-WALLIS TEST RESULT FOR CLUSTER MEMORY USAGE DISTRIBUTION ACROSS BASELINE, ISTIO, AND OPA CONFIGURATIONS UNDER LOW, MEDIUM, AND HIGH LOAD CONDITIONS

Cluster CPU Usage Percentage: Figure 5 shows that under low load (20 RPS), CPU usage remained relatively similar across configurations, with only a slight increase from baseline to Istio+JWT+OPA. However, at medium (60 RPS) and high (100 RPS) loads, a notable rise in CPU usage was observed, especially when OPA was added, indicating increased processing overhead under heavier traffic.

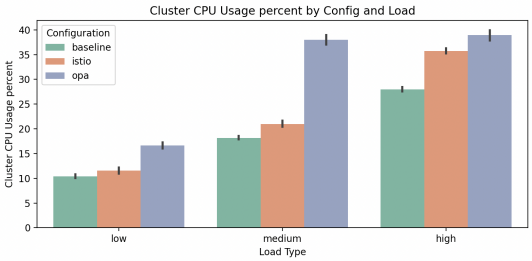


Fig. 5. Grouped bar graph demonstrating CPU Usage Percentage Comparison Across Baseline, Istio, and OPA Configurations Under Low, Medium, and High Load Conditions

Table V shows that CPU usage percentage differs significantly across configurations under all load levels, as all p-values are below 0.05. Dunn's post hoc tests revealed that under high and medium loads, all configuration pairs differ significantly, while under low load, only the comparisons involving OPA are significant indicating that OPA has a distinct impact, whereas Baseline and Istio show similar CPU usage.

Metric	Load	H	p-value
Cluster_CPU_Usage_percent	High	98.99	3.18×10^{-22}
Cluster_CPU_Usage_percent	Medium	107.64	4.23×10^{-24}
Cluster_CPU_Usage_percent	Low	78.25	1.02×10^{-17}

TABLE V
KRUSKAL-WALLIS TEST RESULT FOR CLUSTER CPU USAGE PERCENTAGE DISTRIBUTION ACROSS BASELINE, ISTIO, AND OPA CONFIGURATIONS UNDER LOW, MEDIUM, AND HIGH LOAD CONDITIONS

Pod CPU Usage: Figure 6 shows that total pod CPU usage increases with load across all configurations, with OPA introducing the highest overhead especially under medium load, where it used over 3 CPU cores compared to 1.5 for Istio+JWT and just over 1 for Baseline. While CPU usage rose linearly with load for Baseline and Istio+JWT, OPA's usage peaked at medium load and then slightly dropped at high load, likely due to factors like resource saturation or request drops from latency.

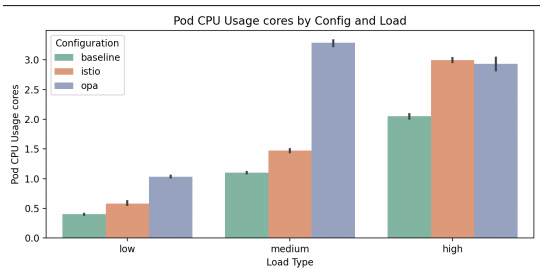


Fig. 6. Grouped bar graph demonstrating Pod CPU usage Comparison Across Baseline, Istio, and OPA Configurations Under Low, Medium, and High Load Conditions

Table VI shows that pod CPU usage varies significantly across configurations under all load levels, as all p-values are below 0.05. Dunn's post hoc tests revealed that under high load, pod CPU usage differs significantly between baseline and both Istio and OPA configurations but not between Istio and OPA, while under low and medium loads, all three configurations show significant differences in pod CPU usage.

Metric	Load	H	p-value
Pod_CPU_Usage_cores	Low	114.55	1.33×10^{-25}
Pod_CPU_Usage_cores	Medium	119.95	8.98×10^{-27}
Pod_CPU_Usage_cores	High	82.48	1.23×10^{-18}

TABLE VI
KRUSKAL-WALLIS TEST RESULT FOR POD CPU USAGE DISTRIBUTION ACROSS BASELINE, ISTIO, AND OPA CONFIGURATIONS UNDER LOW, MEDIUM, AND HIGH LOAD CONDITIONS

Pod Memory Usage: It can be observed from figure 7 that Pod memory usage increased from baseline to Istio+Jwt and further to Istio+Jwt+OPA under low load, but under medium and high loads, Istio+Jwt uses slightly less memory than baseline while OPA adds some overhead compared to the istio only configuration. The additional OPA sidecar increases memory usage noticeably, especially at higher loads, possibly due to the extra processing required for fine-grained access control.

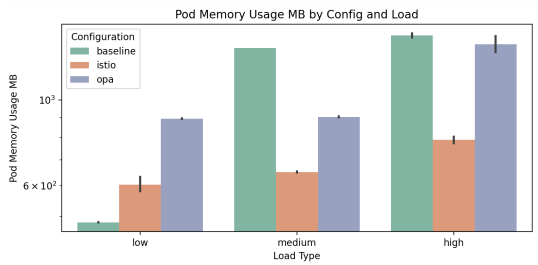


Fig. 7. Grouped bar graph demonstrating Pod Memory usage Comparison Across Baseline, Istio, and OPA Configurations Under Low, Medium, and High Load Conditions

Metric	Load	H	p-value
Pod_Memory_Usage	Low	119.27	1.26×10^{-26}
Pod_Memory_Usage	Medium	119.35	1.21×10^{-26}
Pod_Memory_Usage	High	90.25	2.52×10^{-20}

TABLE VII
KRUSKAL-WALLIS TEST RESULT FOR POD MEMORY USAGE DISTRIBUTION ACROSS BASELINE, ISTIO, AND OPA CONFIGURATIONS UNDER LOW, MEDIUM, AND HIGH LOAD CONDITIONS

From Table VII it can be observed that pod memory usage varies significantly across configurations under all load types, with p-values less than 0.05 leading to rejection of the null hypothesis. Dunn's Post Hoc test revealed that under high load, Istio differed significantly from both baseline and OPA, while

baseline and OPA behaved similarly; under low and medium loads, all three configurations showed statistically distinct pod memory usage.

B. Client/User Perspective

The results of the K6 load test represent the performance metrics from the client's perspective. The results showed how the microservice application on Kubernetes cluster performed in terms of Http request latency, throughput and error rates with increasing load. At a low load of 20 Requests per second, all three configurations (baseline, Istio+Jwt, OPA+Istio+Jwt) handled the traffic relatively well. The baseline configuration recorded p95 latency of 53.21ms with a throughput of 43KBps. Only 2 Virtual users were needed to maintain stable throughput and request rate of 20RPS for the baseline configuration. However, when Istio sidecars and JWT authentication were added, the latency increased slightly to 61.388ms and the throughput decreased to 139KB/s. The Istio+Jwt configuration also needed only 2 Virtual users. For the Opa+Istio+Jwt configuration, latency increased significantly to 92.248ms and throughput decreased to 106KB/s. Only 2 virtual users were needed in this case as well. Even though the performance declined in Opa+istio+jwt configuration as compared to baseline and istio+jwt configuration, the system remained stable under low load for all three configurations and there were no http request errors observed.

At a medium load of 60 Requests per second, the baseline configuration performed relatively well with p95 latency of 47.9ms and throughput of 416Kbps and no http request errors. Maximum of 5 virtual users were required to maintain a stable load of 60 requests per second for the baseline configuration. The Istio+Jwt configuration also performed well with p95 latency of 61.49ms and throughput of 416KB/s with no http request errors. This configuration required 7 virtual users. However, the OPA+Istio+Jwt configuration recorded a very high latency of 24.28 seconds and a significantly low throughput of 302.6KB/s and resulted in 0.136% of the http request to fail. The total number of virtual users required to maintain 60 requests per second with stable throughput for this configuration was 1028. The large number of Virtual users indicate that the requests are taking longer to complete, so more virtual users are spawned by k6 to maintain the same request rate. This shows that some performance bottleneck is observed under moderate load when an additional OPA sidecar is used whereas the baseline and istio+jwt configurations performed relatively well under medium load.

At a high load of 100 Requests per second,only the baseline configuration shows good performance with p95 latency of 79.9 ms, a throughput of 695 KB/s, and zero http errors. It required a maximum of 22 Virtual users to maintain the request rate. The Istio+JWT setup performed poorly with p95 latency of 21.6 seconds, a throughput of 636KB/s and 0.022% of http requests failure. It needed 1426 Virtual users to maintain the load. This indicates that the Istio+Jwt configuration struggles under high load. The OPA+Istio+Jwt system performs even worse with a high p95 latency of 59.99 seconds and very low throughput of 278.4 KB/s. When the system was under high load 66.316% of the requests failed. It needed 2744 Virtual users to maintain the load which highlights a complete system breakdown while handling very high load.

V. Discussion

The research implemented zero trust security using Istio service mesh in an edge-based microservices architecture and conducted a comparative analysis of performance and resource utilization under different loads and configurations. Experiments were carried out on a heterogeneous cluster of VMs and Raspberry Pi4 devices connected via Tailscale, deploying a resource-constrained microservices application without replicas, comparing a baseline setup with no security, an Istio+JWT setup with peer and request authentication, and an Istio+JWT+OPA setup adding attribute-based access control via OPA sidecar injection. The impact of overhead caused by zero trust security on the performance metric like p95 latency was noticeable as the load increased. It was observed that the latency increased with the addition of each security layer. Adding zero trust security with peer authentication and request authentication through Istio sidecar injection to the application pods contributed to higher latency especially under high loads suggesting that it added overhead as compared to baseline deployment with no security. When one more layer of security was added in the form of attribute based access control with OPA, the latency increased drastically. The latency was worst at medium and high loads. This could be because OPA added overhead because it was added as an additional sidecar and also involved policy evaluation and making authorization decisions which might have increased the response time. The cluster network throughput increased for all three configurations with increasing load. Clearly, the OPA+Istio+Jwt configuration performed poorly in terms of cluster network throughput when load increased. This indicated that the system could not handle high requests per second. The low throughput for the OPA+Istio+Jwt could be because of requests being dropped due to i/o timeouts which in turn might be caused by processing time added by external authorizations and attribute based access control policies enforcements. The cluster CPU usage percentage was observed to increase for all three configurations under all three load types: low, medium and high. It was clear from the results that adding peer authentication and request authentication with Istio sidecars does add cpu usage overhead compared to baseline configuration but the addition of OPA was observed to add drastically high overhead with increasing load. The addition of the OPA layer in the existing system, recorded saturation as load increased more. This might be due to many possible reasons: requests dropping due to high latency or the system being unable to handle a high number of requests causing bottlenecks. The cluster memory usage percentage was observed to not change much with increasing load. The cluster memory usage was observed to have similar values, with slight differences across configurations with increasing loads. Pod memory usage increased noticeably from baseline

to Istio+JWT and further to Istio+JWT+OPA under low load, but at higher loads, the baseline consumed significantly more memory, likely due to Istio offloading networking tasks. Pod CPU usage rose linearly for baseline and Istio+JWT across all loads, while Istio+JWT+OPA showed a spike at medium load and a slight decrease at high load, with baseline consistently using the least CPU cores; the addition of each security layer increased CPU usage, especially at low and medium loads, and OPA notably impacted resource usage under high load.

The results of K6 load test were analyzed to evaluate the performance overhead as observed from a user perspective. It was observed that under very high load, the overhead introduced by Istio and OPA were more significant. The baseline system was observed to be efficient under increasing load with latency under threshold values. Adding security with mTLS and request authentication with JWT introduced slight overhead as compared to the baseline system under low and medium load but the overhead was more noticeable and significantly high at higher load. It was also observed that a few requests failed for Istio+JWT configuration under high load. The OPA+Istio+JWT configuration severly impacted the performance as it introduced very high latency and low throughput under high and medium loads. Under high load, the HTTP error rate was very high with more than half of the requests failing with I/O timeout. The Istio+JWT configuration struggled under high load but the Istio+JWT+OPA configuration struggled even on medium and high load. Additional layers of security introduced in the form of service to service authentication and external authorization had significant impact on the performance from the user's perspective especially in terms of request latency and error rate. While the findings highlight the significant performance overhead introduced by Zero Trust enforcement mechanisms, these results also underscore the importance of scalability and optimization strategies. Mechanisms such as horizontal pod autoscaling or caching could potentially reduce latency and improve resilience under heavy workloads.

Limitations The study was conducted in a controlled and resource constrained environment and therefore may not fully represent the edge computing clusters with scalable resources. The real world edge clusters might include horizontal and vertical scaling functionalities. The research aims to give a baseline performance metrics without any scaling involved. Future work could integrate auto-scaling strategies to assess how dynamic resource allocation impacts Zero Trust performance in edge environments. The testbed cluster was formed over Tailscale and therefore the results might have been influenced by latency induced by Tailscale connectivity but its potential contribution to latency was minimized by ensuring stable connections and identical network paths across all configurations. Therefore, while Tailscale may have introduced some baseline latency, it was consistent across all experimental setups, allowing for a fair comparative evaluation of performance overheads.

VI. Conclusion

The study investigated the security vs performance tradeoff in a distributed edge cluster testbed running a resource constrained microservices based application. In an edge computing environment with limited resources, performance overhead can become a major concern. The study analyzed the impact on system performance and resource utilization due to different security configuration in a resource constrained edge cluster. It was observed that the baseline configuration with no security measures performed comparatively better with low latency. The baseline configuration scaled well with increasing load with no significant performance degradation under high load. When a layer of security was added by enabling peer authentication and request authentication with Istio, the system performed moderately well in low load and medium load scenarios but struggled slightly under high load. Adding authorization policies with OPA added one more layer of security with fine grained Attribute Based Access Control but also increased performance overhead. The system struggled to handle increasing loads and the performance degradation was visible even at low load. The system with additional layer of security with OPA struggled at medium load and high load with very high latency and high error rate at high load. The zero trust security configuration including peer authentication and request authentication with Istio sidecar proxy and fine grained authorization policies with OPA provides security in the microservices architecture by ensuring per request authentication and authorization and least privilege access but introduces high performance overhead. In resource constrained edge clusters, adding zero trust security in the microservices architecture can result in poor performance.

VII. Future Directions

The performance impact of zero trust security and the security vs performance tradeoff in a microservices based scalable distributed edge computing cluster can be studied in the future. Moreover, exploring optimization techniques such as caching of policy decisions could further reduce the performance overhead observed in this study. The impact of authorization with more complex and dynamic access control policies on the performance in both resource constrained and scalable edge clusters can be studied. The zero trust security in AI based microservices applications on edge and fog computing can be explored.

References

[1] Campbell, M. (2020). Beyond Zero Trust: Trust Is a Vulnerability. Computer, 53(10), 110-113. https://doi.org/10.1109/MC.2020.3011081

[2] Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., Doss, R. (2022). Zero Trust Architecture (ZTA): A Comprehensive Survey. IEEE Access, 10, 57143-57179. https://doi.org/10.1109/ACCESS.2022.3174679

[3] Yarygina, T., Bagge, A. H. (2018). Overcoming Security Challenges in Microservice Architectures. IEEE Symposium on Service-Oriented System Engineering. https://doi.org/10.1109/SOSE.2018.00011

[4] Chandramouli, R., Butcher, Z., Chetal, A. (2021). Attribute-based Access Control for Microservices-based Applications Using a Service Mesh. NIST Special Publication 800-204B. https://doi.org/10.6028/NIST.SP.800-204B

[5] Rodigari, S., O'Shea, D., McCarthy, P., McCarry, M., McSweeney, S. (2021). Performance Analysis of Zero-Trust multi-cloud. IEEE 14th International Conference on Cloud Computing (CLOUD). https://doi.org/10.1109/CLOUD53861.2021.00097.

[6] Kurpad, S., BT, S., Vijaykumar, S., Jain, S., Kalambur, S. (2023). Microarchitectural Analysis and Characterization of Performance Overheads in Service Meshes with Kubernetes. 3rd Asian Conference on Innovation in Technology (ASIANCON). https://doi.org/10.1109/ASIANCON58793.2023.10270428

[7] Rodigari , S. (2023). Performance Analysis of Zero Trust in Cloud Native Systems [Master's thesis, Munster Technological University]. https://sword.cit.ie/cgi/viewcontent.cgi?article=1542context=allthe

[8] Viswanathan, J., Kumar. N, D. K., Kumar, S. U. (2024). Zero Trust Security for Web Applications in Microservice-Based Environments. 2024 First International Conference on Data, Computation and Communication (ICDCC), 488-494. https://doi.org/10.1109/ICDCC62744.2024.10960955

[9] Qu, Q., Xu, R., Nikouei, S. Y., Chen, Y. (2020). An Experimental Study on Microservices based Edge Computing Platforms. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 836-841. https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9163068

[10] Hossain, M. D., Sultana, T., Akhter, S., Hossain, M. I., Thu, N. T., Huynh, L. N., Lee, G. W., Huh, E. (2023). The role of microservice approach in edge computing: Opportunities, challenges, and research directions. ICT Express, 9(6), 1162-1182. https://doi.org/10.1016/j.icte.2023.06.006

[11] Ganguli, M., Ranganath, S., Ravisundar, S., Layek, A., Ilangovan, D., Verplanke, E. (2021). Challenges and Opportunities in Performance Benchmarking of Service Mesh for the Edge. IEEE International Conference on Edge Computing (EDGE), 78-85. https://doi.org/10.1109/EDGE53862.2021.00020

[12] DE JESUS SILVA, J. M. (2024). Zero Trust Security for microservices in scalable systems [Master's Thesis, Instituto Superior de Engenharia do Porto (ISEP)]. https://recipp.ipp.pt/entities/publication/2cb18176-f6ab-4ac3-bee8-c9852a59b999

A Study on Cybersecurity Risks and Protections for Digital Twin Applications

Minhaz Mahmud
Erasmus+ MUNDUS: CyberMACS
SRH Campus Berlin and Kadir Has University
https://orcid.org/0009-0005-4699-1940

Reiner Creutzburg
SRH Campus Berlin
Berlin, Germany
reiner.creutzburg@srh.de

Adele Nasti
SRH Campus Berlin
Berlin, Germany
adele.nasti@srh.de

Md Saiful Islam
SRH Campus Berlin
Berlin, Germany
saiful.islam@srh.de

***Abstract*—Digital twins are increasingly used in smart grids for real-time monitoring and analysis, but they introduce new cybersecurity concerns, especially regarding data integrity and false data injection attacks. In this paper, we design defenses against two feasible data-integrity attacks – false data injection and data tampering – in a smart grid digital twin to promote the use of a hybrid defense combining machine learning-based anomaly identification and blockchain. We build a high-fidelity smart grid twin by extending its accuracy through residual learning models that validate simulation output with real lab data. An unsupervised One-Class Support Vector Machine (OCSVM) is trained to detect anomalous injection or tampering attempts in real-time, while all identified anomalies are logged by a Hyperledger Fabric permissioned blockchain to maintain tamper-proof data integrity. The enhanced twin achieves virtual-equivalence to physical readings and substantially higher accuracy over basic simulations (up to 56% reduced prediction error). Under attack, OCSVM detects $> 95\%$ of malicious data points with negligible false alarms. Simultaneously, blockchain-based logging injects an immutable, consensus-verified audit trail for every identified anomaly that defeats attackers from covering their tracks. The results demonstrate that through the combination of an accurate, data-upgraded twin with simultaneous smart anomaly identification and blockchain-based auditing, the resiliency and trustworthiness of digital twin applications to power grids can be substantially enhanced.**

***Index Terms*—Smart grid, Digital twin, Blockchain, Machine learning**

I. Introduction

Digital twin technology enables physical power grid infrastructure to be simulated and digitally mirrored for real-time monitoring, with operations and analysis benefits. Digital twins are being rapidly adopted by utilities – by a 2023 survey's estimate, around 65% of electric utilities (87% in the United States) adopt them to varying extents. However, those benefits come with those of cybersecurity risks; according to a comprehensive survey, many unresolved security flaws target digital twins directly. In particular, False Data Injection Attacks (FDIAs) and Data Tampering Attacks threaten the integrity of the sensor information of smart grids. An FDIA comprises an attacker injecting ambiguous measurements or tampering with sensor inputs to mislead the grid's state estimate and operating conditions. By falsifying telemetry (e.g., overestimating a feeder load or spoofing a voltage reading), an attacker may befuddle control actions or conceal faults. More generally, data tampering is unauthorized data alteration of data in transit or at rest that causes systems to operate upon tainted information. Such exploits may be subtle – advanced attackers preserve false data within plausible values in hopes of slipping beneath simple threshold alarms. Both FDIA and tampering directly attack data integrity, which is critical in power grids depending upon accurate sensor inputs. Indeed, tainted data may cause improper control actions, equipment damage, or blackouts, as was seen in real-world cases (e.g., during the 2015 Ukraine grid cyberattack, malicious SCADA data modifications took place). Traditional defenses cannot sufficiently protect against advanced attacks. Recent research trends are oriented towards advanced anomaly detection – numerous works implement machine learning and deep learning to detect FDIAs. One-class unsupervised algorithms are of particular interest since they learn only normal behavior and can signal an alarm for any divergence, including unknown attack novelties. At the same time, blockchain technology has emerged as a way to secure data logs by means of immutability and distributed consensus. Permissioned blockchains like Hyperledger Fabric allow only qualified nodes to write and read data, yet show high throughput and acceptably low latency for industrial control environments. Previous works indicated blockchain-based solutions for the prevention of data tampering in smart grids, referring to the attractiveness of a tamper-proof ledger for critical infrastructure. Nevertheless, integrating real-time anomaly detection and blockchain in the scope of a digital twin is an open task. In this paper, the hybrid cybersecurity architecture of the digital twin of a smart grid is proposed that combines: (1) an augmented machine learning-based digital twin model to represent accurately the physical system, (2) an OCSVM-based anomaly detector to reveal potentially malicious data or corrupted data in the twin's streams of inputs, and (3) blockchain-based Hyperledger Fabric layer to commit detected anomalies to an immutable ledger. Using machine learning to specify normal system activity and blockchain to protect the outcomes of detection, the solution addresses real-time prevention of attack as well as after-the-fact forensics following an incident. The solution is validated in simulations of a testbed smartgrid (PV panel, battery storage system, load, and OPAL-RT simulator) and its corresponding digital twin. The research findings are as follows: Digital Twin

Enhancement: A residual learning-based method to advance power system digital twin realism through learning from simulation (Pandapower) to physical-measurement variations. It significantly reduced simulation error, ensuring the twin mirrors the real system behavior. Anomaly Detection Using One-Class SVM: Application of a one-class SVM trained for real-time detection of the injected fake data or tampering. Following extensive testing against various methods (Isolation Forest, PCA, kNN, etc.), OCSVM was selected based on its favorable performance (F1-scores 0.91–0.99 in our tests) and demonstrated success in SCADA environments. Blockchain-based Anomaly Protection: Utilization of a Hyperledger Fabric network to commit the ML detector's identified anomalies. Each of the suspicious data activities is committed as a transaction directly to the ledger, which cannot be altered or deleted by design. It establishes a decentralized, trustworthy record of attack audit trails that minimizes insider threats and supports forensic analysis. By combining these aspects, we achieve a cyber-resilient digital twin: the machine learning detector prevents hidden data assaults in real-time, and the blockchain ensures that all recognized events are immortalized beyond the attacker's control. The rest of the paper is structured into sections that describe related work (Section II), our method comprising the digital twin modeling, anomaly detection, and blockchain deployment (Section III), experimental results (Section IV), discussion of implications (Section V), and conclusions (Section VI).

II. Literature Review

Cybersecurity Risks to Digital Twins: Cybersecurity risks for critical infrastructure's digital twin technology are of concern. [1] provides an in-depth survey of the risks to digital twins, reporting that industry system digital surrogates grow the attack surface and necessitate new defenses. In power grids, researchers have noted data integrity attacks like FDIA as particularly dangerous, as they can covertly modify grid sensor readings. [2] first revealed FDIA vulnerabilities to state estimation in power grids, showing that such attacks can evade standard bad-data detection if carefully constructed. Subsequent surveys [3] describe many machine learning algorithms for FDIAs in smart grids, and agreement is apparent that anomaly-based bad-data detection is required if only normal measurements can be spoofed by attackers. Data tampering attacks – more broadly, unauthorized data tampering – were also discussed. For example, [4] reports a blockchain-based twin-based architecture to prevent data tampering in smart grids, unveiling decentralized ledger potential for guaranteeing data integrity. An extension of these findings by our research is that FDIA and tampering should be treated in the same twin security model. Stealthy false data is of particular value to us, for it addresses scenarios in which attackers hold changes in normal bands to stay invisible, as is reported to be of concern in today's literature.

Anomaly Detection for Smart Grids: Various anomaly detection techniques have been experimented with for power grid cybersecurity [5]. Traditional methods are residual-based detectors in state estimators and statistical thresholds, but they are often outsmarted by carefully engineered attacks. Thus, research has shifted to data-driven methods. Supervised classifiers (e.g., tree ensembles, deep learning) obtain good detection rates for observed attack patterns, but require labeled attack data and may not generalize to novel risks. Supervised detectors, especially one-class detectors, are interesting for learning normal patterns from normal data without experiencing attack data. One-Class SVM (OCSVM) and isolation forest were compared for application to anomaly detection for critical infrastructure; [6] showed that OCSVM produced more reliable detection under changing attack scenarios at minimal computational cost. One-Class SVM has been very successful in industrial control systems (ICS) intrusion detection – for instance, [7] outlined how OCSVM outperformed several techniques of outlier detection to detect malicious traffic through the substation SCADA network. In consideration of such results, the emphasis of this work is thus on OCSVM. Nevertheless, for completeness, the investigation of alternative unsupervised algorithms (e.g., kNN, Local Outlier Factor, PCA-based detection, etc.) is included, as different algorithms may be strongest for various signals. Indeed, tests by us demonstrate that kNN and even PCA can be similar to OCSVM in certain cases (e.g., for battery anomalies) and that for one of the subsystems (the OPAL simulator), LOF was somewhat higher. Typically, results thus validate literature findings that OCSVM provides a widely based, high-accuracy solution to anomaly detection for smart grids, especially if it is made to be adequately sensitive without over-firing to normal variations.

Blockchain for Data Integrity: Blockchain has been explored in the case of smart grid security to permit tamper-proof, decentralized data logs. At the same time, anyone can participate in public blockchains (e.g., Ethereum), permissioned blockchains like Hyperledger Fabric permit only known entities to submit proposals and forgo proof-of-work, yet reaching much higher throughput and reduced latency – qualities relevant to real-time grid activity. [8] discuss Fabric's architecture, including modularity and advanced security capabilities (identity management, endorsement policies, etc.), that qualify Fabric for enterprise and critical infrastructure deployment. In a smart grid case, developing researchers used blockchain to secure control commands and measurement data. [9] demonstrated a Fabric-based approach to secure IoT sensor data at the grid edge and noted that Fabric's consensus and access control can ensure data integrity and confidentiality for multi-stakeholder use cases. Some works combine blockchain with anomaly detection or another AI: for example, [10] document a hybrid AI-blockchain design for smart grids, and [11] propose a decentralized blockchain-ML design for grid security. These papers note the complementarity of making use of blockchain as a trust overlayer atop bright-eyed detectors. Our work differs in that the tight coupling of the anomaly detector and blockchain logging into one real-time digital twin. Instead of handling blockchain as a generic afterthought, we specifically design programmable

smart contracts (chaincode) to log anomalies and invoke auto-response. It draws inspiration from antecedent work that points to the strength of programmability of Fabric – e.g., custom chaincode can automatically cross-validate data or invoke an auto-alert. We also meticulously measure the blockchain layer's impact on performance, in agreement with recent Fabric scalability studies (e.g., [12] showing Fabric can realize high transaction throughput at sub-second latency in real-world deployments). Our prototype confirms that anomaly logging can be maintained at near real-time (0.5–2 second latencies per-event) while security guarantees are not sacrificed with suitable batching and network settings.

METHODOLOGY

A. Digital Twin Overview

Our test case is a lab-scale smart grid testbed with a PV inverter (up to 3.2 kW), a battery storage unit (±9 kVA bidirectional inverter), a programmable load (0–1 kW), and an OPAL-RT real-time simulator that represents an external microgrid supply. All devices run at 50 Hz, 400 V (line-line). The digital twin was built using the Pandapower library to model the electrical network and calibrated with real-world data. The twin model includes an infinite bus for the grid and buses for each component (PV, battery, load, OPAL), connected by short lines that mimic lab cabling. Only steady-state power flow simulations were performed since the focus is on steady-state anomalies rather than transients. The twin was carefully tuned using historical measurements so it could replicate the actual system's operating envelope, including empirical means and variances for voltages, currents, frequency, and power factors. Outputs were also post-processed with realistic noise and minor imbalances so that voltage readings oscillate around 230 V with the same subtle variations seen in the lab. This prevents the twin from being a "perfect textbook" model and makes it behave like the real system—imperfections included—which is crucial for training an anomaly detector with realistic data.

B. Digital Twin Residual Enhancement

To boost fidelity further, we applied residual learning with supervised machine learning to correct systematic errors in the simulation. A dataset of synchronized readings from both the physical lab and the twin (under identical conditions) was collected, covering key signals: three-phase voltages (V_L1, V_L2, V_L3), currents (I_L1, I_L2, I_L3), per-phase active powers (P_L1, P_L2, P_L3), and system frequency. Different operating scenarios (varying PV outputs, battery charge/discharge states, load levels) were included. Residuals were computed as: Residual(t) = Physical Measurement(t) – Twin Simulation(t). These residuals represent simulation error. Models were trained to predict these residuals using the twin's outputs and system inputs, effectively learning the gap between simulation and reality. Separate models were developed for each subsystem: Battery Twin Model: Trained on battery signals to refine battery behavior. Inverter Twin Model: PV inverter data was used to improve PV subsystem predictions. Load Twin Model: Corrected outputs based on measured load usage. Microgrid (Opal) Twin Model: Used all inputs to correct the entire system collectively. After training, each model produced corrected outputs as: Twin_corrected = Twin_raw simulation + Predicted Residual. This provided outputs that were much closer to real measurements. Evaluation on unseen scenarios used metrics like MAE and RMSE, along with percentage improvement over raw simulation. A 100% improvement indicates perfect alignment with reality, 0% means no change, and negative values mean performance worsened. While training/testing was offline with logged data, the approach could be adapted for real-time online calibration.

C. Attack Injection and Data Preparation

For anomaly detector training, we required both normal and attack data. Physical system in normal operation data were treated as normal data, ensuring the detector was trained on system-accurate inputs. Cyber-attacks were simulated by injecting anomalies into a copy of this dataset. Examples include sudden spikes/drops in sensor readings, biased voltage/current signals, stuck-at values, or other invalid deviations. Some anomalies were subtle (stealthy, within plausible ranges) while others were obvious extremes, creating a diverse attack set. Anomalies spanned all subsystems (e.g., false PV readings, incorrect battery states, frequency errors) and replaced 5–10% of the dataset, so training was mostly on clean data. Labels were kept for evaluation but not used in training since a one-class model relies only on normal data. Importantly, all attacks were introduced at the data stream level of the twin (inputs/outputs), not via hardware tampering, providing a controlled environment for detection and response tests.

D. Machine Learning Anomaly Detection

We implemented an unsupervised anomaly detection module that monitors the digital twin's data feed. Several algorithms were evaluated: Isolation Forest (IF), One-Class SVM (OCSVM), Local Outlier Factor (LOF), Elliptic Envelope (robust covariance), Principal Component Analysis (PCA) based anomaly detection, Histogram-Based Outlier Score (HBOS), and k-Nearest Neighbors (kNN) outlier detection. Each model was trained on the physical system in the normal operation data dataset for each component. That is, we trained separate detectors for the Battery subsystem, PV inverter, Load, and Opal (microgrid). This component-wise approach recognizes that standard behavior profiles differ across subsystems. Training a one-class model involves tuning hyperparameters to balance sensitivity and false alarms. For OCSVM, we used a radial basis function (RBF) kernel and tuned the parameter (which roughly sets the expected fraction of outliers) to around 0.05 (5%) based on validation data. This meant the OCSVM would flag roughly the top 5% deviating points as anomalies, aligning with the known attack injection rate. Similar contamination or threshold parameters were set for the other models (e.g., LOF, Elliptic Envelope) to ensure a fair comparison in terms of how many points they flag. During testing, each model produced an anomaly score or binary classification for each incoming data sample. We computed

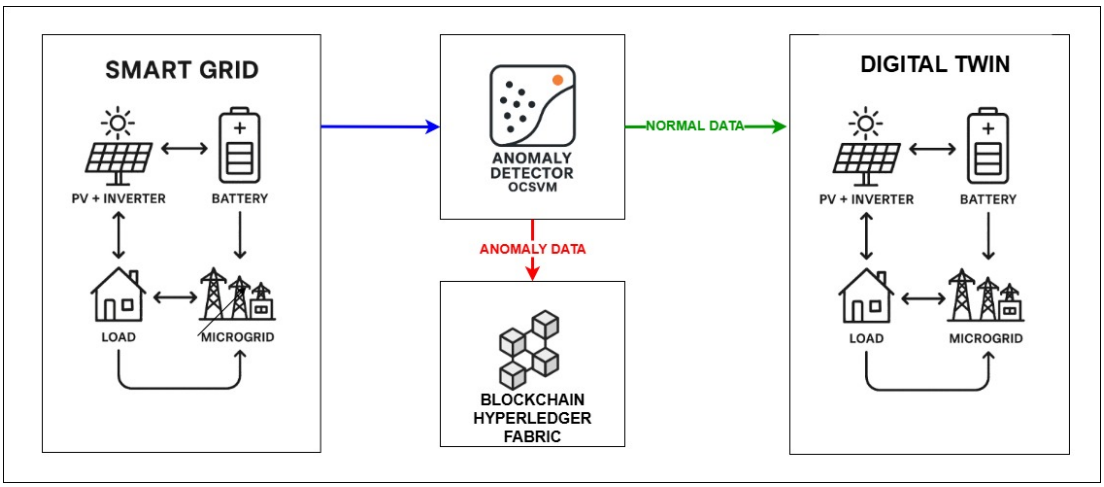


Fig. 1. Complete system overview

standard classification metrics against the ground truth labels of injected vs. normal points: Accuracy, Precision, Recall, and F1-score. These metrics were calculated per subsystem to gauge detection performance in different operating contexts. For brevity, our discussion focuses on F1-score as a balanced indicator of detection efficacy. The models performance on the test data revealed that certain algorithms consistently outperformed others. In particular, the OCSVM emerged as the top or near-top performer across most components, aligning with our expectations from prior works. In our final system deployment, we chose the OCSVM as the primary detector for its overall best performance and conceptual advantages discussed earlier. (In cases where other models tied or excelled marginally – e.g., kNN and PCA on the battery – the difference was small, and OCSVM offers a well-defined decision boundary useful for interpretability.)

E. Blockchain Logging and Security Layer

The final layer of our architecture is the permissioned blockchain for secure logging. We set up a small Hyperledger Fabric network with two organizations (simulating, for example, a utility company and a regulator, both of whom have an interest in the anomaly records). The Fabric network comprises multiple peer nodes and an ordering service using the Raft consensus protocol. We developed chaincode (smart contract) in Go that defines an Anomaly Log asset and transactions to record anomalies. When the anomaly detector flags a data point as suspicious, the system immediately submits a LogAnomaly transaction to the blockchain network. The chaincode was designed with key features: it records details of the anomaly (timestamp, component, measured values, detection confidence, etc.) and attaches a cryptographic hash of the data for integrity. Each anomaly record is indexed by component and time to allow quick queries (e.g., retrieve all anomalies for the battery subsystem). Security Configurations: We leveraged several of Hyperledger Fabric's security mechanisms to ensure the logging system is robust against tampering: an endorsement policy requiring endorsement from both organizations' peers was set for the LogAnomaly transaction. This means an anomaly entry is only added if peers from independent orgs vouch for it, preventing a compromised insider from unilaterally inserting or deleting records. All transactions are identity-authenticated using X.509 certificates for clients, and each anomaly entry is digitally signed and included in a block with a hash chain linking to previous blocks (ensuring immutability). The ledger is replicated across peers in both organizations, providing fault tolerance – even if some nodes are down or compromised, the anomaly log persists on others. Access control is enforced so that only permissioned members (e.g., grid operator, regulator) can query anomaly data. Fabric's inherent immutability guarantees that once an anomaly is recorded on-chain, it cannot be altered or erased without detection. This immutable audit trail is crucial for post-incident investigations – for example, if a false data injection attack is attempted at a certain time, the forged readings and their timestamps will be permanently recorded on the ledger for forensic analysis.

F. Integration and Workflow

Figure 1 illustrates the integrated system workflow. The digital twin continuously receives live data from the physical sensors and produces a synchronized simulation output. The residual correction model (if deployed online) adjusts these outputs to improve accuracy. The anomaly detector (OCSVM) monitors the twin's output (or input) streams in real-time. Under normal conditions, data flows through to operators or control systems as usual. If an anomaly is detected – e.g., a sudden unexplained jump in load measurements indicative of

an FDIA – the detector module flags it and immediately invokes the blockchain client SDK to propose a new anomaly log transaction. The anomaly (with metadata) is then committed to the Fabric ledger after endorsement and ordering. The presence of the blockchain layer does not prevent real-time control actions; it operates asynchronously. In practice, the detection alert can also be used to trigger mitigation responses (such as ignoring a suspected false reading or switching to a safe mode), but in this work, we focus on detection and logging. The blockchain serves as a forensic memory, ensuring that even if an attacker were to clear or falsify logs in the central SCADA system, the distributed ledger retains evidence of the anomaly. We also carefully evaluated the performance: logging one anomaly per transaction achieved latencies of around half a second, and in stress tests, a batch of 100 anomalies took only a few seconds to commit, confirming the approach is viable for near-real-time requirements.

Results and Discussion

G. Digital Twin Fidelity

We first assess the baseline accuracy of the digital twin in replicating the physical system's behavior, and then the improvements achieved by the residual learning enhancement. The twin achieves an exact match in steady-state bus voltages (all phase voltage ratios = 1.000) and closely matches frequency (50.001 Hz vs physical 49.122 Hz). However, notable discrepancies appear in currents and power: the twins' battery currents differ by up to 13% from the physical (e.g., I_L3: 0.482 A vs 0.554 A), and it underestimates the battery's three-phase active power by 15–18%. This indicates the initial Pandapower-based twin, while good for voltages and frequency, did not perfectly capture the battery's current draw and output power – likely due to simplifications in the battery model (e.g., not modeling all internal losses or dynamics). The PV Inverter subsystem shows much tighter agreement. The twin and physical PV readings are nearly identical across all metrics: phase voltage ratios 0.998 (difference ¡0.2%), currents 2.12 A each with ¡0.3% difference, and active powers matching within 0.1%. The twin correctly reflected the PV inverter exporting 465 W per phase (negative sign indicating power injection), almost exactly as observed, and frequency was spot on at 50.001 Hz. This high fidelity for the PV is attributed to effective calibration of the inverter model and static generator in Pandapower, yielding an excellent digital replica. The Load subsystem results likewise show good agreement for voltages and frequency (ratios 1.0), but because the absolute load was very small (tens of watts or fractions of an ampere), even tiny absolute differences manifest as large percentage errors. For instance, physical load current 0.137 A vs twin 0.030 A on one phase yields a ratio 0.216 (an apparent 78% underestimation). In absolute terms, the difference is only 0.107 A, but it appears large relative to the near-zero load. Similarly, one phase current was overestimated (0.232 A vs 0.137 A, ratio 1.696). These variations are carried into the computed load power, with one phase's power being 7.5% high and another 3.4% low. Such divergences at very low loads highlight the twin's sensitivity; minor model uncertainties can cause relatively large percentage swings when the true values are near zero. Finally, the Opal/Microgrid subsystem exhibited close correspondence overall: the twin slightly overestimates currents (1.758 A vs 1.677 A, 4–5% high) and consequently power (twin shows about 1.045–1.050 times the physical power export). Despite this bias, the variations (standard deviations) of all measurements are almost identical between the twin and reality, indicating the twin captured not only the mean operating point but also the fluctuation range of the microgrid. In summary, the baseline twin was highly accurate for grid voltage and frequency, and reasonably accurate for currents and power with some component-specific errors. These results instill confidence that the twin can serve as a proxy for the real system's normal behavior, while also pointing to where improvements are needed (e.g., battery current modeling).

H. Enhancement via Residual Learning

Incorporating real data through the residual ML models substantially improved twin fidelity for most components. The Battery twin model, interestingly, did not yield improvement – in fact, it degraded accuracy. With only battery-related features, the model appeared to overfit noise or mis-adjust the simulation. The result was an average MAE increase of about 186% (i.e., error nearly tripled) for the battery signals. We include this negative result for completeness: it suggests that the battery's behavior in isolation was too subtle or complex to learn with the given data. Minor mismatches (e.g., due to battery internal resistance or state-of-charge effects) might have been drowned out by noise, causing the model to introduce erroneous corrections. This underscores that adding a single component's data in isolation might not always improve the twin; a holistic approach may be needed. In contrast, the Inverter twin model achieved a moderate but clear improvement. It reduced the PV inverter subsystem's MAE by roughly 20% on average (with a similar 22% reduction in RMSE). Currents benefited the most – the inverter twin cut current measurement error by about 23–26%, reflecting that it learned to account for real-world inverter dynamics or losses not in the base simulation. Voltage predictions were already very close, but two of the phase voltages saw 5% MAE improvement. Overall, while the base twin was already excellent for the inverter, the residual model fine-tuned it further, especially improving current accuracy. The Load twin model yielded a large jump in accuracy. By training on the actual load measurements, the twin's average error dropped 42% MAE (40% RMSE) relative to the raw simulation. Certain signals improved dramatically – for example, one phase current error decreased by over 90%, essentially aligning the twin's current trace almost exactly with the real lab data. Phase voltages, which had small errors to begin with, saw about 30% error reductions with the load model. These substantial gains suggest that the Pandapower simulation's handling of loads had notable uncertainty; the twin initially didn't perfectly capture how small load changes affect the system. By feeding actual load data into the model, the twin learned to correct those discrepancies, closely tracking

TABLE I
PERFORMANCE COMPARISON OF MODELS ACROSS DIFFERENT COMPONENTS

Component	Best Accuracy (Model)	Best Precision (Model)	Best Recall (Model)	Best F1 (Model)
Battery	kNN (98.9%)	kNN (96.6%)	kNN (97.7%)	kNN (97.1%)
Inverter+PV	OCSVM (96.3%)	OCSVM (90.7%)	OCSVM (90.9%)	OCSVM (90.8%)
Load	OCSVM (99.4%)	OCSVM (98.4%)	OCSVM (98.7%)	OCSVM (98.5%)
Opal	LOF (96.3%)	LOF (90.7%)	LOF (90.9%)	LOF (90.8%)

the real system's response to load variations. This result demonstrates the value of integrating live load measurements into a power system's digital twin – load is a major driver of system state, and even modest misestimations can be fixed with such data-driven calibration. Finally, the Microgrid (Opal) twin model provided the most impressive improvement. The overall MAE across signals dropped by 56.5% (from an average of 54.2 in base sim to 23.6 in twin), and RMSE by 49%. Virtually all signals – three-phase voltages and currents – saw an order of 50–60% error reduction. Even though the base twin was already quite good for voltages/frequency (within 2% as noted), the full-system model cleaned up much of the remaining error and also significantly corrected the power flow bias in the Opal subsystem. In fact, a major discrepancy in the base model was the Opal power flow: the simulation often assumed a higher export power than reality (essentially overestimating how much the OPAL-RT source was supplying). The residual model recognized this and adjusted the twin's power predictions closer to actual import/export values. After enhancement, the twin's representation of the microgrid was much more faithful, both in steady magnitudes and in dynamic response. In summary, except for the isolated battery case, the residual machine learning calibration substantially improved twin accuracy, especially when using comprehensive system data. The enhanced twin provides a more precise normal behavior profile, which is beneficial for anomaly detection (fewer false positives due to model error) and for any operational planning that uses the twin.

I. Anomaly Detection Performance

We evaluated the anomaly detection across the four subsystems using the test dataset with injected attacks. The key metrics for the best-performing models are compiled in Table 1. Overall, the OCSVM, kNN, and PCA detectors performed exceptionally well, each achieving high true positive rates and low false positives on most components. For the Battery data, OCSVM, kNN, and PCA all attained nearly identical performance, with F1 of 0.97. For instance, OCSVM caught 87% of injected battery anomalies with only 3% false alarm rate. This corresponds to confusion matrix counts of about 76 true positives out of 87 attacks, and 12 false positives out of 350 normal points. Such performance – over 96% precision and 96% recall – indicates the top models effectively learned the normal battery behavior and detected even subtle deviations. Simpler algorithms didn't fare as well: for example, Histogram-Based Outlier Score (HBOS) only detected 37% of battery anomalies and produced many false alarms (precision 36%), showing the difficulty of a univariate approach on complex multi-sensor data. Isolation Forest and LOF were intermediate, catching roughly 50–66% of anomalies with some false positives. Similar trends were observed in the other subsystems. For the PV Inverter data, OCSVM emerged as the top model with F1 of 0.91. It detected over 90% of the false data injections on inverter measurements while keeping false alerts low. The PV twin's high fidelity made the anomalies (such as fabricated PV output readings or voltage deviations) relatively easier to spot as outliers. The Load subsystem, which normally operates at near-zero load, was extremely sensitive to anomalies – any significant non-zero reading outside the learned tiny range was flagged. OCSVM achieved F1 0.985 on load data, essentially near-perfect detection. This is expected, as any false load injection (e.g., a fake surge of several kW) is a large deviation from the baseline of only tens of watts; all models found load attacks easy to detect due to the clear separation between normal and abnormal in that case. For the Opal (microgrid) subsystem, which had the lowest twin fidelity initially, the detection was slightly more challenging. Interestingly, the LOF algorithm performed best on Opal data with F1 of 0.908, marginally above OCSVM and kNN (each of which was "equally close" to LOF's performance). LOF's strength on Opal could be due to its local density focus, which might handle the twin's bias better. Even so, an F1 of 0.90 means around 90% of anomalies were caught. Specifically, many anomalies in the Opal signals involved abnormal power flow readings or frequency disturbances; these were detected reliably despite the twin's baseline error, because our detectors were trained on actual twin outputs (biased as they were). This emphasizes a benefit of our approach: even if the twin isn't perfect, as long as its normal deviations and biases are consistent, the one-class model can still identify points that deviate from that norm (including those biases). In practice, improving the twin helped detection – the residual-corrected twin for Opal meant fewer false positives for the detector. Across all components, the One-Class SVM provided a balanced high performance, making it an ideal single choice for deployment. OCSVM's ability to form a tight boundary around normal data made it conservative against false alarms while still catching the vast majority of attacks. Notably, OCSVM (as well as kNN and PCA) produced near-diagonal confusion matrices for each subsystem, indicating robust detection across multiple attack types. For example, in the battery case, the

OCSVM had 76 true positives vs 11 false negatives, and 338 true negatives vs 12 false positives – an excellent balance. Meanwhile, models like HBOS or Isolation Forest showed scattered matrices with many misclassifications. These detailed results confirm that our anomaly detection approach (training on twin data and injecting test anomalies) was effective, and validate the selection of OCSVM as the primary detector going forward.

J. Blockchain Logging Outcomes

We next examine the performance and security outcomes of the Hyperledger Fabric anomaly logging component. The Fabric network consistently achieved low-latency commits for anomaly records. In single-anomaly transactions (the common case when anomalies are infrequent), the end-to-end commit time – from proposal to ledger write – was approximately 0.5 seconds on average. This is fast enough for near-real-time requirements, meaning the anomaly is on-chain almost immediately after detection. In scenarios with bursts of anomalies, we used the chaincode's batch logging capability (grouping multiple anomaly events into one transaction). Even in a worst-case test of 100 anomalies batched, the commit latency was only around 2–3 seconds. This demonstrates that the system can handle events at the speed of grid dynamics (which typically operate on the order of seconds or longer for control decisions). Throughput was not a bottleneck in our relatively small network; prior research indicates Fabric can handle hundreds of transactions per second easily in similar settings, and our design doesn't generate an excessive volume of transactions (only when anomalies occur). From a security standpoint, the implementation of blockchain added a strong layer of resilience. In our deployment, these translated to concrete protections: the endorsement policy requiring two organizations ensured that a rogue operator or compromised node in one organization could not manipulate the anomaly log. For instance, if an insider at the utility tried to delete or alter an anomaly record to hide an incident, the modification would be rejected because it wouldn't satisfy the endorsement policy (the regulator's peer would not endorse a tampered transaction). The cryptographic integrity of the ledger (hash-linked blocks and signature validation) makes any on-chain tampering practically impossible – any attempt to change a logged anomaly would break the chain and be immediately detected by all peers. We confirmed this immutability: anomaly entries once written could not be altered even by an admin without invalidating ledger hashes. The ledger thus serves as an immutable audit trail. To illustrate, consider a false data injection attack on the load readings where an attacker briefly fakes a high load. Our system would detect it (OCSVM flags the deviation) and record it on-chain. Even if the attacker then stops the attack and the system returns to normal, the event is permanently stored with a timestamp and details. The attacker (or even a malicious insider) cannot erase this evidence. This immutability addresses a key aspect of cyber resilience – forensics and accountability. Any future investigation or compliance audit can trust the blockchain records to determine what happened. Moreover, access control in Fabric ensured that sensitive anomaly information was only visible to authorized parties (in our case, we configured that only the utility and regulator org members could query the full anomaly history, protecting data from exposure to outsiders). The blockchain layer did not hinder the digital twin's performance; it operated in parallel, logging events without interfering with detection or the twin's simulation. Our analysis of CPU and network usage indicated that the overhead of signing and committing anomaly transactions was minimal on the scale of our system. This is consistent with other studies that have reported Fabric can be integrated into IoT and control systems with manageable overhead. In summary, the blockchain provided high-integrity, distributed logging of anomalies with negligible impact on speed, fulfilling its role as the tamper-resistant memory of the system.

CONCLUSION

In this paper, we presented a comprehensive cybersecurity solution for digital twin applications in smart grids, addressing the prominent risks of false data injection and data tampering. We demonstrated that enhancing the digital twin's fidelity using residual machine learning models greatly improves the twin's credibility and the effectiveness of subsequent security measures. On this foundation, a One-Class SVM anomaly detector was able to identify integrity attacks on the twin's data streams with high precision – achieving 90–99% F1-scores in detecting a variety of injected false data points across different subsystems. Finally, by integrating a Hyperledger Fabric blockchain, every detected anomaly is immediately recorded on an immutable, distributed ledger, ensuring that the system's security events are tamper-proof and auditable. The hybrid approach proved effective: the machine learning component caught subtle attacks in real-time, while the blockchain component guaranteed end-to-end data integrity (both in operation and in logs) without imposing undue latency. The outcome is a resilient digital twin architecture that not only faithfully represents the physical system under normal conditions but also robustly defends against and records malicious data manipulations. We have shown that the confluence of digital twin technology, anomaly detection, and blockchain can significantly raise the bar for attackers – anomalies are detected swiftly, and any malicious activity cannot be erased from history. This contributes to the overall safety and trustworthiness of smart grid operations, as the twin can be confidently used for monitoring and control knowing that attacks will be pinpointed and preserved for response.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to all who supported this work. I am deeply thankful to Saiful Islam, Lecturer at SRH University of Applied Sciences, for setting up the laboratory and providing valuable guidance in understanding the laboratory equipment and electrical components. I also extend my heartfelt appreciation to Prof. Reiner for his continuous support and expertise in the field of Cyber

Security. Furthermore, I am grateful to Adele Nasti for her insightful contributions in the area of Digital Twin research. I also acknowledge the Erasmus Mundus Master's Programme in Applied Cybersecurity (CyberMACS) for providing me with the academic environment and resources that greatly enriched this research. Their collective support and guidance were essential in completing this work.

REFERENCES

[1] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1475–1503, 2022.

[2] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, pp. 1–33, 2011.

[3] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2218–2234, 2019.

[4] B. Boi, C. Esposito, and J. T. Seo, "Preventing data tampering in smart grids: A blockchain-based digital twin framework," in *International Conference on Computational Science and Its Applications*. Springer, 2024, pp. 144–156.

[5] P. Kumar, R. Kumar, A. Aljuhani, D. Javeed, A. Jolfaei, and A. N. Islam, "Digital twin-driven sdn for smart grid: A deep learning integrated blockchain for cybersecurity," *Solar Energy*, vol. 263, p. 111921, 2023.

[6] K. Lukito, A. F. Ihsan *et al.*, "Comparison of isolation forest and one class svm in anomaly detection of gas pipeline operation," in *2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*. IEEE, 2023, pp. 118–123.

[7] M. Egger, G. Eibl, and D. Engel, "Comparison of approaches for intrusion detection in substations using the iec 60870-5-104 protocol," *Energy Informatics*, vol. 3, no. Suppl 1, p. 15, 2020.

[8] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[9] H. Honar Pajooh, M. Rashid, F. Alam, and S. Demidenko, "Hyperledger fabric blockchain for securing the edge internet of things," *Sensors*, vol. 21, no. 2, p. 359, 2021.

[10] Y. Y. Ghadi, T. Mazhar, T. Shahzad, I. H. Jaghdam, S. Khan, M. A. Khan, and H. Hamam, "A hybrid ai-blockchain security framework for smart grids," *Scientific Reports*, vol. 15, no. 1, p. 20882, 2025.

[11] Y. ISHAQ, A. PRINCE, G. CLAUDE, and T. BEBE, "Decentralized framework for securing smart grids using blockchain and machine learning," *INTERNATIONAL JOURNAL*, vol. 13, no. 1, pp. 679–685, 2025.

[12] Y. Ucbas, A. Eleyan, M. Hammoudeh, and M. Alohaly, "Performance and scalability analysis of ethereum and hyperledger fabric," *IEEE Access*, vol. 11, pp. 67 156–67 167, 2023.

Adapting Cybersecurity Governance Frameworks to Manage Risks in Generative AI Systems

Remilekun Adeopatoye
SRH University of Applied Sciences
Berlin, Germany
adeopatoye@gmail.com

Prof. Knut Haufe
SRH Heidelberg University of Applied Sciences
Berlin, Germany
Knut.Haufe@de.ey.com

Prof. Reiner Creutzburg
SRH Heidelberg University of Applied Sciences
Berlin, Germany
Reiner.Creutzburg@srh-hochschulen.de

Izuchukwu Patrick Udechukwu
SRH Heidelberg University of Applied Sciences
Berlin, Germany
pi.udechukwu.max@gmail.com

ABSTRACT

The accelerated adoption of Generative AI (GenAI) systems, particularly large language models (LLMs) has introduced cybersecurity risks that exceed the scope of traditional governance frameworks like NIST CSF and ISO/IEC 27001. This paper presents a comprehensive evaluation of these governance gaps and proposes the Minimal AI Information Security Control Set (M-AI-ISCS)—a hybrid control set that integrates AI-specific safeguards with established cybersecurity standards. Developed through Design Science Research (DSR), the control set addresses emerging threats including prompt injection, data leakage, model inversion, and adversarial manipulation. Scenario-based testing in healthcare and fintech domains assesses the control set's effectiveness, feasibility, and regulatory alignment. Results indicate that while certain controls—such as continuous monitoring and incident response—are universally critical, others require domain-specific adaptation, including ethical guardrails and privacy protection in healthcare, and adversarial detection and API security in fintech. The findings demonstrate that M-AI-ISCS improves organizational preparedness, enhances regulatory compliance, and strengthens operational resilience in GenAI deployments.

Keywords— Generative AI (GenAI), Cybersecurity Governance, Compliance, Risk Management Frameworks, ISO/IEC 27001, NIST CSF, NIST AI RMF, ISO/IEC 42001, Compliance.

I. INTRODUCTION

The accelerated deployment of generative AI (GenAI) systems, particularly large language models (LLMs) like ChatGPT, LLaMA, and Claude, has fundamentally shifted the risk landscape in cybersecurity. These models, now integral to operations in sectors from finance and healthcare to legal services and content creation, generate outputs that are not only innovative but also unpredictable, amplifying vulnerabilities in ways that traditional governance frameworks may not anticipate.

[1] and [2] state that the adoption of GenAI in critical workflows has gone ahead of the maturation of security standards, leaving users vulnerable to threats combining technological savvy and human exploitation. The disconnect exists most explicitly in the inability of current frameworks to cover the specific risks in GenAI like as prompt injection, model inversion, data leakage from training sets, and adversarial manipulation [3], [4]. Standards such as ISO/IEC 27001 provide a flexible, risk-based approach, enabling organizations to categorize AI systems as information assets but are not capable of having prescriptive control measures specific to the probabilistic nature of GenAI where the output can change due to incremental changes in the input or hidden biases. [5] cites this weakness, in that without special guidance, the type of risks such as adversarial manipulation in which attackers design the input to deceive the models are not yet adequately controlled. Likewise, the NIST Cybersecurity Framework (CSF) calls out functions such as Identify and Protect but in the nature of its framework assumes deterministic systems, for which it has difficulty coping with the adaptive behaviors in GenAI described by [6] as an inherently dynamic and difficult to bound system. Such weaknesses in high-risk contexts, such as financial decisions or diagnosis in healthcare, could result in catastrophic outcomes ranging from the theft of proprietary algorithms to wrongful output leading to reputational or legal damage.

This paper presents a scenario-based evaluation of the Minimal AI Information Security Control Set (M-AI-ISCS), a governance toolkit developed to mitigate risks associated with generative AI deployment. While the control set was derived from a broader Design Science Research (DSR) project, this study focuses specifically on testing its applicability and effectiveness through two representative scenarios—one in healthcare and one in fintech. These scenarios simulate realistic adversarial attacks and operational conditions, highlighting the practical relevance and domain-specific performance of the proposed controls.

II. LITERATURE REVIEW

A. Cybersecurity Governance Frameworks

Cybersecurity governance frameworks such as the NIST Cybersecurity Framework (CSF) and ISO/IEC 27001 have long served as foundational instruments for managing information security risks across sectors. Their structured methodologies which consist of comprehensive risk assessments, asset prioritization, threat evaluation, and incident response are well established for traditional IT environments. However, their ability to address the distinctive challenges posed by artificial intelligence (AI), especially GenAI systems, remains limited [3], [4].

While NIST CSF offers modularity and flexibility beneficial to risk monitoring and incident response, it overlooks specific provisions for AI-centric threats such as

prompt injection, data poisoning, and model evasion. On the other hand, ISO/IEC 27001 supports structured compliance processes through its Plan-Do-Check-Act (PDCA) cycle and is widely adopted for certifiable governance. Yet, its firmness and lack of adaptability make it not suitable for the fast-paced evolution of GenAI systems [7]. The introduction of ISO/IEC 42001, a dedicated AI management system standard, marks significant progress, yet its coverage of generative AI risks remains incomplete [4].

Framework	Strengths in AI Contexts	Limitations in AI Contexts
ISO/IEC 27001	- Structured ISMS with risk assessment methodology [3] - Globally recognized certification supports trust and compliance	- Lacks explicit AI-specific risk controls - Static controls do not adapt to AI learning or evolution [4]
ISO/IEC 27002	- Detailed control guidance for ISO/IEC 27001 - Flexibility in tailoring to organizational context	- Not designed with AI threat models in mind - No support for explainability, human oversight, or model monitoring
NIST CSF	- Modular and technology-neutral - Five-function model aligns well with risk management processes [6]	- Voluntary, lacking prescriptive depth - Inadequate provisions for generative AI risks like hallucinations or prompt injection [4]

Table 1 Summarizes the strengths and limitations in AI context

These limitations highlight the urgency for AI-aware adaptations of traditional cybersecurity frameworks to accommodate the emergent threat landscape associated with advanced AI technologies. NIST CSF's flexible design better supports iterative AI deployment and contextual risk analysis, whereas ISO/IEC 27001's emphasis on compliance may hinder responsiveness to dynamic threat landscapes. Importantly, neither framework fully considers the probabilistic nature or evolving behavioral dynamics of GenAI models, reinforcing the need for integrated AI-specific governance mechanisms.

B. Generative AI Risks

The integration of generative AI (GenAI) systems, particularly large language models (LLMs) such as ChatGPT, LLaMA, Claude, and Gemini has introduced new dimensions of cybersecurity risk within digital systems. Unlike traditional software governed by deterministic logic, GenAI systems are probabilistic, learn from continuously evolving datasets, and are embedded in complex sociotechnical contexts. These characteristics generate risks that are underrepresented in conventional governance tools such as NIST CSF and ISO/IEC 27001.

Key risk categories include prompt injection, adversarial prompting, model evasion, and training data extraction [2]. These methods exploit vulnerabilities in AI models to manipulate outputs, degrade accuracy, or exfiltrate sensitive training data [5]. These threats introduce new vectors of attack and ethical concerns, often leveraging the opacity of model internals and user over trust in AI outputs. Adversaries now employ role-based manipulation and "jailbreaking" techniques to elicit harmful content, bypass filters, or subvert operational rules [8], [9]. Further technical threats such as model inversion and membership inference pose data confidentiality challenges, enabling attackers to reconstruct or infer sensitive training data, often in violation of privacy mandates [6]. These attacks have critical implications for GDPR compliance and AI ethics, particularly in high-stakes environments like healthcare, finance, and public safety.

Traditional governance frameworks are ill-equipped to model or mitigate these AI-specific threats. Both NIST CSF and ISO/IEC 27001 lack built-in mechanisms for AI model lifecycle monitoring, bias detection, or post-deployment adaptation. Newer frameworks such as NIST AI RMF and ISO/IEC 42001 attempt to address these limitations, but their integration into enterprise risk workflows remains uneven and lacks harmonized operational tooling. This gap presents a challenge to organizations seeking comprehensive and responsive governance over GenAI deployments.

C. AI-Specific Standards

To address the gaps in traditional frameworks, dedicated standards have emerged to address the unique risks and lifecycle considerations of AI. Notably, the NIST AI Risk Management Framework (AI RMF) and ISO/IEC 42001 are positioned to guide organisations through AI-specific risk governance, offering structured models aligned with principles of transparency, fairness, accountability, and robustness. The NIST AI RMF organises risk management activities into four functional pillars and introduces a sociotechnical lens that prioritises context-aware risk assessment and stakeholder engagement [1]. It aligns closely with the EU AI Act compliance, particularly Articles 8 and 9 [10]. However, its guidance remains largely abstract, and it lacks prescriptive controls for GenAI risks such as adversarial prompting, training data leakage, or model drift. In practice, organizations require clearer interoperability between AI-specific frameworks and cybersecurity standards to ensure end-to-end coverage of AI risks.

While the AI RMF and ISO 42001 are promising, they must be operationalised through maturity models, interoperability guidance, and stronger enforcement mechanisms. In practice, organizations will benefit from hybrid models that integrate both traditional and AI-specific standards that translate these high-level standards into actionable governance practice [1]

D. Regulatory Landscape

The evolving regulatory landscape reflects increasing recognition of the systemic risks posed by generative AI. In particular, the European Union Artificial Intelligence Act (EU AI Act), a landmark legal instrument, introduces a risk-tiered framework for AI regulation and imposes stringent obligations on high-risk systems. Under Articles 8–15, the Act mandates the implementation of risk management systems, data governance, technical robustness, transparency, and human oversight. It also recommends alignment with established security frameworks such. As such, regulatory compliance now requires a layered approach that merges information security controls with AI governance protocols [10].

Simultaneously, the General Data Protection Regulation (GDPR) continues to play a central role in shaping data protection obligations in the context of AI. Core provisions on lawfulness (Art. 6), data minimization (Art. 5), and automated decision-making (Art. 22) apply directly to generative models that collect, infer, or output personally identifiable information. Threats such as training data

leakage and model inversion can violate the confidentiality and integrity principles outlined in Article 5(1)(f), placing organizations at risk of significant penalties [8]. Regulatory convergence is accelerating, but a gap remains between legal requirements and their technical operationalization. Frameworks like ISO/IEC 42001 and NIST AI RMF to offer regulatory mappings, impact assessments, and documentation protocols tailored to generative AI. Yet practical tools and enforcement pathways are still underdeveloped.

E. Governance Gaps for Novel Threat Surfaces

The accelerated deployment and adoption of generative AI systems have exposed critical governance gaps in existing cybersecurity governance frameworks, which were primarily designed to address traditional IT risks rather than the probabilistic, evolving, and opaque nature of AI-specific threats [1]. Standards such as ISO/IEC 27001 and the NIST Cybersecurity Framework (CSF) lack explicit provisions for novel AI attack vectors including prompt injection, model poisoning, and leakage of sensitive training data [11]. These outlier present unique challenges for risk assessment, traceability, and incident response [12], [13]. These emerging threats exploit the probabilistic and adaptive nature of generative AI models, posing challenges to conventional security approaches that assume static asset boundaries and deterministic behaviour [14]. Lee, Kim, and Whang (2025) categorize generative AI risks into systemic misuse, content safety, societal impact, and legal/rights-related risks, emphasizing AI's ability to produce harmful outputs across modalities such as text, images, and video, that threaten service integrity and public trust. They demonstrate that traditional cybersecurity frameworks lack mechanisms to address these evolving vulnerabilities, such as prompt injections, data leakage, and algorithmic bias, underscoring an urgent governance deficit [15].

Shared to these studies is consensus that existing governance frameworks cybersecurity and regulatory are ill-prepared for these dynamic, multi-sided risks presented by generative AI. Propounded solutions converge upon:

- Developing continuous, real-time risk monitoring and adversarial testing pipelines integrated with traditional ISMS and risk management systems [16], [17]
- Embedding human oversight and explainability mechanisms to enhance transparency, trust, and accountability [18], [19]
- Implementing unified AI TRiSM frameworks that align trust, risk, and security governance across the AI lifecycle [20]
- Advocating for adaptive, experimental regulation that balances innovation and safety while fostering international cooperation [13], [14]
- Enhancing cross-disciplinary collaboration to address technical, ethical, and legal challenges holistically [19], [20]

F. Role of MITRE ATLAS in AI Adversarial Taxonomy

MITRE ATLAS (Adversarial Threat Landscape for AI Systems) provides a common repository for finding and mitigating threats in artificial intelligence systems that are adversarial, particularly those related to those involving generative AI models. Constructed from within the very foundation of what is in the MITRE ATT&CK framework, ATLAS zeroes in specifically on AI system vulnerability and provides a formal methodology for documenting mappings of TTPs in an adversarial nature at different stages in an AI lifecycle such as training, inference, and post-deployment stages [21]. As a newer repository which continues in its expansion, ATLAS classifies TTPs for exploitation in AI system vulnerability and differs significantly from legacy cybersecurity conceptualizations such as MITRE ATT&CK®, which are for use with legacy IT infrastructures. Through its classification of security professionals on specific risks such as those involving data poisoning, model inversion, and input crafting as an adversarial input, ATLAS provides information on an actionable nature involving means defences can strengthen [22]. Through its enumeration of the typical adversarial behaviours employed in exploiting an AI model, in particular those involving those of a generative nature, ATLAS allows cybersecurity professionals in mitigating risks in an AI lifecycle better

III. METHODOLOGY

This study adopts Design Science Research (DSR) as its all-encompassing methodological approach, which provides a robust framework for addressing complex, emergent problems by systematically designing, constructing, and evaluating innovative artefacts. DSR is particularly well suited for contexts where the goal is to develop, test, and refine innovative artefacts that address real-world problems while contributing to scientific knowledge which in this case, a unified cybersecurity governance toolkit adapted to manage risks in Generative AI (GenAI) systems. This approach transcends purely explanatory research by blending theoretical rigour with practical relevance, aligning with Peffers, K DSR methodology model [23]. DSR is distinctively positioned at the intersection of theory and practice, it integrates abductive, deductive, and inductive reasoning to guide iterative design, systematic evaluation, and theory building [24].

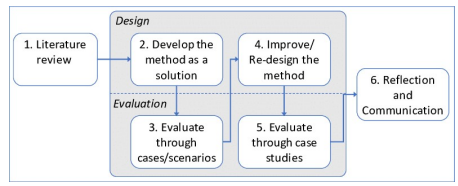


Figure 1: Design Science Research

For the purpose of this conference paper, the focus is on evaluating the effectiveness of the developed control set using scenario-based testing. The control set integrates AI-specific governance mechanisms from NIST AI RMF and ISO/IEC 42001 into traditional cybersecurity frameworks (NIST CSF and ISO/IEC 27001), targeting critical GenAI risks such as prompt injection, model poisoning, adversarial prompting, and training data leakage.

A. Artefact: Minimal AI Information Security Control Set (M-AI-ISCS)

The primary artefact developed in this study is the Minimal AI Information Security Control Set (M-AI-ISCS). The control set integrates governance, technical, operational,

and ethical dimensions to provide a structured framework for managing risks associated with Generative AI deployment. The Minimal AI Information Security Control Set (M-AI-ISCS) is structured around three core principles. First, it emphasizes risk mitigation by addressing AI-specific threats such as adversarial attacks, model backdoors, and data leakage. Second, it focuses on regulatory compliance to ensure alignment with applicable laws and standards governing AI use. Third, it prioritizes operationalization, aiming to design practical and adaptable controls that can be effectively implemented in real-world AI deployment scenarios. This control set is aligned with the AI-specific overlays and existing cybersecurity frameworks, and is categorized into three primary groups:

Governance & Management Controls

These controls focus on policy, oversight, and compliance for AI systems to be constructed, maintained, and monitored securely. Effective governance for models using AI requires efficient accountability frameworks, risk management, and regular audits.

Control	Description	AI-Specific Focus
AI Risk Governance Framework	Establish an AI-specific risk governance framework to manage AI-specific threats.	Formalizes responsibility for AI risk management across departments.
Model Risk Assessment & Auditing	Perform regular risk assessments and audits to evaluate security, fairness, and ethical implications of AI models.	Ensures models are free from biases, vulnerabilities, and alignment issues.
AI Transparency & Accountability Policy	Create policies for transparency in AI system design, data handling, and decision-making processes.	Focus on ensuring AI systems provide explainability and are auditable.
AI Compliance with Regulations	Implement processes to ensure compliance with EU AI Act, GDPR, and other AI regulations.	Controls for data privacy, fairness, and accountability in AI models, ensuring compliance with regulatory standards.

Table 2: Governance & Management Controls

Technical Controls

These controls target the technical means required for protecting AI systems against some of the emerging cybersecurity threats such as adversarial attacks, unauthorized access, and data exfiltration. They harden the technical security posture of AI models and make them more resistant to emerging threats.

Control	Description	AI-Specific Focus
Adversarial Attack Detection & Mitigation	Implement algorithms and processes to detect and mitigate adversarial inputs (e.g., prompt injection, data poisoning).	Adversarial robustness for AI systems, including model re-training and input sanitization.
Access Control & API Security	Strengthen access controls for AI models and APIs, including MFA, rate limiting, and RBAC.	Secure access to sensitive models and training data through API security measures.
Model Encryption & Data Protection	Encrypt AI model parameters and training data to protect sensitive data during training, deployment, and inference.	Prevents data leakage and model theft by securing critical components.
Integrity Monitoring & Validation	Use integrity checks to monitor models for unauthorized changes and ensure model output validity.	Detects unauthorized modifications to models (e.g., backdoor implantation).
Differential Privacy for Model Outputs	Apply differential privacy techniques to limit sensitive data exposure during inference.	Protects sensitive data from inference attacks and data leakage by ensuring that individual data points cannot be identified.

Table 3: Technical Controls

Operational Controls

These controls assist in making sure that the systems of AI are adequately controlled, tested, and monitored throughout their lifetime. They assist in realizing the continuous security, fairness, and ethical control of the deployed models of AI.

Control	Description	AI-Specific Focus
Continuous Monitoring of AI Models	Implement continuous real-time monitoring to detect abnormal behavior, performance issues, or adversarial activity.	Focuses on monitoring AI model behavior during deployment to prevent misuse and detect malicious activity.
Automated Vulnerability Scanning for AI Models	Perform automated scanning of AI models and infrastructure for vulnerabilities.	Identifies vulnerabilities like data poisoning, model theft, and backdoor creation, ensuring proactive risk management.
Ethical Guardrails & Model Fail-safes	Develop and implement ethical guardrails to prevent AI models from generating harmful, toxic, or misleading outputs.	Introduces safeguards to mitigate toxic content generation (e.g., hate speech, misinformation).
Regular AI Model Audits & Revisions	Conduct regular audits and revisions to ensure compliance with ethical guidelines, performance standards, and security policies.	Ensures that AI models comply with ethical, security, and performance benchmarks and remain fair and accurate over time.
Incident Response & Model Recovery Plans	Create incident response plans and model recovery protocols in case of security breaches or model malfunctions.	Focuses on recovering from AI-specific failures, ensuring the continuity and resilience of AI models post-incident.

Table 4:Operational Controls

B. Scenario-Based Evaluation

To evaluate the practical effectiveness of the M-AI-ISCS framework, two scenario-based tests were conducted in high-risk operational contexts in healthcare and fintech, where Generative AI systems are actively deployed. These scenarios were selected to represent distinct regulatory, ethical, and technical risk profiles.

- Healthcare Scenario: Assesses risks such as model poisoning, PHI leakage, and regulatory non-compliance.
- Fintech Scenario: Focuses on prompt injection, data exfiltration, and algorithmic bias.

Each scenario evaluates the framework's ability to identify relevant threats, apply appropriate controls, and maintain alignment with applicable standards and regulations.

C. Metrics for Evaluation

The effectiveness of the control set is assessed using three key metrics:

- Effectiveness: The extent to which the applied controls mitigated the targeted risks in each scenario.
- Compliance Coverage: Extent to which controls satisfy relevant standards and regulatory requirements.
- Feasibility: Practicality and ease of implementation of controls in real-world operational environments.

Results were documented in structured comparison tables (see Section IV) to support reproducibility and highlight domain-specific variations.

IV. SCENARIO DESIGN / IMPLEMENTATION

A. Healthcare Scenario

Scenario Overview

A national healthcare provider deployed a fine-tuned LLM to automate clinical documentation and patient communication, integrated with EHR systems on a HIPAA-compliant cloud.

AI System Purpose

- Clinical documentation generation (e.g., discharge summaries)
- Patient communication via chatbot
- Language simplification for medical terms

Risk Profile

- PHI leakage through model inference
- Inaccurate clinical documentation
- Unauthorized internal access
- Regulatory non-compliance (HIPAA, GDPR)

This healthcare scenario provides a high-stakes environment to test the effectiveness of AI-specific governance, security, and privacy controls. It establishes a robust foundation for evaluating the M-AI-ISCS control set in a real-world operational context.

Control Application

The M-AI-ISCS was applied to address these risks:

- Privacy Protection: Differential privacy and model encryption for PHI.
- Operational Resilience: Continuous monitoring, auditing, and model recovery.
- Governance & Accountability: Model risk auditing and transparency policies.
- Explainability & Ethical Safety: Ethical guardrails to prevent misinformation.

Evaluation Insights

The evaluation of M-AI-ISCS controls in the healthcare scenario demonstrated strong alignment with the sector's dual priorities of clinical safety and regulatory compliance. Controls emphasizing transparency, traceability, and output validation were particularly prioritized due to the life-critical nature of clinical decisions. Ethical guardrails and fail-safe mechanisms were deemed more important in healthcare than in fintech, reflecting the higher stakes associated with patient outcomes. Additionally, the auditability of model outputs was identified as a key area for future enhancement to ensure adherence to clinical documentation standards.

B. Fintech Organization Scenario

A fintech startup deployed a GenAI system for real-time investment advisory, processing user transaction histories, behavioral data, and market trends.

AI System Purpose

- Personalized investment recommendations
- Financial product matching
- Market risk assessments

Risk Profile

- Prompt injection and adversarial manipulation
- Data exfiltration of financial or personal information
- Biased recommendations leading to unfair financial outcomes
- Compliance nonconformity (GDPR, EU AI Act, financial conduct regulations)

Control Application

The M-AI-ISCS was applied as follows:

- Adversarial Threat Mitigation: Attack detection and API security.
- Privacy Protection: Encryption of sensitive financial data.
- Governance & Accountability: AI risk governance framework.
- Operational Resilience: Continuous monitoring and incident response plans.

Evaluation Insights

The application of M-AI-ISCS in the fintech scenario covered 80–90% of critical risk areas, with particular effectiveness in addressing adversarial and operational threats. Risks associated with public interfaces required the strongest mitigation measures, especially given the real-time nature of financial interactions. Certain controls, such as differential privacy, were found to have limited relevance in this context. Recommendations for future enhancement include improving algorithmic explainability and strengthening financial traceability protocols to ensure robust and accountable AI-driven financial advice.

V. RESULTS AND DISCUSSION

This section presents the comparative insights from the application of the Minimal AI Information Security Control Set (M-AI-ISCS) across the Fintech and Healthcare scenarios. By evaluating control performance in two high-risk, regulated domains, the analysis reveals how contextual factors such as data sensitivity, deployment models, and regulatory obligations—shape the prioritization and effectiveness of AI-specific security controls. The application of the M-AI-ISCS across two domains—fintech and healthcare—demonstrated the toolkit's adaptability,

effectiveness, and relevance in mitigating Generative AI-specific risks. While core control categories remained stable, their prioritization varied by domain:

- Fintech emphasized adversarial robustness, transparency, and API security.
- Healthcare prioritized data protection, explainability, and clinical integrity.

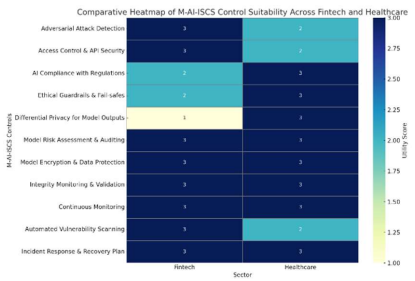


Figure 2: The relative utility of each M-AI-ISCS control across the fintech and healthcare scenarios

The heatmap above visually illustrates the relative utility of each M-AI-ISCS control across the fintech and healthcare scenarios, providing a clear comparison of control effectiveness. It supports the analyses presented in scenario section by highlighting both common and sector-specific priorities. Controls such as Continuous Monitoring and Incident Response & Recovery demonstrate high utility across both sectors, reflecting their critical importance in managing AI risks. Sector-specific variations are also evident: in fintech, greater emphasis is placed on Adversarial Attack Detection and API Security, while in healthcare, Differential Privacy, Ethical Guardrails, and Model Risk Auditing are prioritized due to the heightened sensitivity of patient data and the life-critical nature of clinical decisions.

Control Category	Universally Critical	Fintech-Weighted	Healthcare-Weighted
Governance & Accountability	AI Risk Governance	—	—
Adversarial Threat Mitigation	—	Adversarial Attack Detection	—
Privacy Protection	Model Encryption	—	Differential Privacy
Explainability & Ethical Safety	—	—	Ethical Guardrails
Operational Resilience	Continuous Monitoring	API Security	Model Validation

Table 5: Comparative control prioritization across Fintech and Healthcare domains.

The above table presents a cross-sector comparison of the relative priority of selected controls from the Minimal AI Information Security Control Set (M-AI-ISCS) as observed in the Fintech and Healthcare scenarios. It highlights controls that were universally critical, as well as those that showed increased relevance within a specific domain. This trend analysis suggests that while some controls form the foundation of AI risk management regardless of domain, others must be emphasized based on sector-specific threats, compliance obligations, and the societal impact of AI decisions. These results validate the M-AI-ISCS as a baseline AI security framework adaptable to sector-specific constraints and regulatory obligations.

VI. CONCLUSION

This paper presented a scenario-based evaluation of the Minimal AI Information Security Control Set (M-AI-ISCS), illustrating how existing cybersecurity governance frameworks can be adapted to effectively manage the unique risks posed by Generative AI (GenAI) systems. By applying the control set to two high-risk, high-regulation domains—fintech and healthcare—the study demonstrates the feasibility of tailoring governance mechanisms to address sector-specific threat profiles and regulatory requirements.

The findings indicate that core cybersecurity principles such as risk assessment, continuous monitoring, and incident response can be effectively extended to encompass AI-specific threats, including prompt injection, model inversion, and data leakage. In particular, the results underscore the importance of domain-specific customization. For instance, fintech environments require a heightened focus on adversarial threat mitigation and API-level security, while healthcare scenarios demand stronger controls around ethical guardrails, differential privacy, and model validation due to the sensitivity and impact of clinical decision-making.

Furthermore, the scenario-based approach offered practical insights into the prioritization and integration of AI-specific controls within existing operational workflows. It emphasized the value of aligning governance practices with real-world deployment conditions to support both compliance and security.

Overall, the evaluation demonstrates that the M-AI-ISCS offers a viable, adaptable, and pragmatic approach for enhancing cybersecurity governance in the age of generative AI. The framework not only bridges current governance gaps but also supports the safe and compliant deployment of GenAI systems. Future research may extend this approach to additional sectors and integrate automated assessment tools to enable scalable, real-time risk oversight across AI deployment lifecycles.

REFERENCES

[1] R. Dotan, B. Blili-Hamelin, R. Madhavan, J. Matthews, and J. Scarpino, "Evolving AI Risk Management: A Maturity Model based on the NIST AI Risk Management Framework," Jan. 2024, [Online]. Available: http://arxiv.org/abs/2401.15229

[2] D. Humphreys, A. Koay, D. Desmond, and E. Mealy, "AI hype as a cyber security risk: the moral responsibility of implementing generative AI in business," *AI and Ethics*, vol. 4, no. 3, pp. 791–804, Aug. 2024, doi: 10.1007/s43681-024-00443-4.

[3] A. Alghamdi, "Comparative Analysis of ISO27001 and NIST CSF," 2023.

[4] T. R. McIntosh, T. Susnjak, T. Liu, P. Watters, R. Nowrozy, and M. N. Halgamuge, "From COBIT to ISO 42001: Evaluating Cybersecurity Frameworks for Opportunities, Risks, and Regulatory Compliance in Commercializing Large Language Models," Feb. 2024, doi: 10.1016/j.cose.2024.103964.

[5] M. Malatji, “Comparative analysis of adversarial AI injection attacks: A preliminary study,” in *International Conference on Artificial Intelligence, Computer, Data Sciences, and Applications, ACDSA 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ACDSA59508.2024.10467951.

[6] S. Neupane, I. A. Fernandez, S. Mittal, and S. Rahimi, “Impacts and Risk of Generative AI Technology on Cyber Defense,” Jun. 2023, [Online]. Available: http://arxiv.org/abs/2306.13033

[7] A. CALDER and S. G. WATKINS, “THE ISO 27001 RISK ASSESSMENT,” in *Information Security Risk Management for ISO 27001/ISO 27002, third edition*, IT Governance Publishing, 2019, pp. 87–93. doi: 10.2307/j.ctvndv9kx.11.

[8] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, “From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy,” 2023, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2023.3300381.

[9] Y. Yigit, W. J. Buchanan, M. G. Tehrani, and L. Maglaras, “Review of Generative AI Methods in Cybersecurity,” Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.08701

[10] J. Pötsch, “Interplay of ISMS and AIMS in context of the EU AI Act.”

[11] Y. Bengio *et al.*, “Managing extreme AI risks amid rapid progress: Preparation requires technical research and development, as well as adaptive, proactive governance,” *Science (1979)*, vol. 384, no. 6698, pp. 842–845, May 2024, doi: 10.1126/science.adn0117.

[12] F. Hu, S. Liu, X. Cheng, P. Guo, and M. Yu, “Academic Journal of Management and Social Sciences Risks of Generative Artificial Intelligence and Multi-Tool Governance”.

[13] P. Hacker, A. Engel, and M. Mauer, “Regulating ChatGPT and other Large Generative AI Models,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jun. 2023, pp. 1112–1123. doi: 10.1145/3593013.3594067.

[14] X. Wang and Y. C. Wu, “Balancing Innovation and Regulation in the Age of Generative Artificial Intelligence,” *Journal of Information Policy*, vol. 14, Jul. 2024, doi: 10.5325/jinfopoli.14.2024.0012.

[15] K. Lee, H. Kim, and J. J. Whang, “SAIF: A Comprehensive Framework for Evaluating the Risks of Generative AI in the Public Sector,” Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.08814

[16] A. Srivastava and S. Panda, “A Formal Framework for Assessing and Mitigating Emergent Security Risks in Generative AI Models: Bridging Theory and Dynamic Risk Mitigation,” Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.13897

[17] R. H. Filho and D. Colares, “A Methodology for Risk Management of Generative AI based Systems.”

[18] Z. L. Teo, C. W. N. Quek, J. L. Y. Wong, and D. S. W. Ting, “Cybersecurity in the generative artificial intelligence era,” Jul. 01, 2024, *Elsevier B.V.* doi: 10.1016/j.apjo.2024.100091.

[19] R. Pasupuleti, R. Vadapalli, and C. Mader, “Cyber Security Issues and Challenges Related to Generative AI and ChatGPT,” in *Proceedings - 2023 10th International Conference on Social Networks Analysis, Management and Security, SNAMS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/SNAMS60348.2023.10375472.

[20] A. Habbal, M. K. Ali, and M. A. Abuzaraida, “Artificial Intelligence Trust, Risk and Security Management (AI TRiSM): Frameworks, applications, challenges and future research directions,” Apr. 15, 2024, *Elsevier Ltd*. doi: 10.1016/j.eswa.2023.122442.

[21] A. T. Olutimehin, A. J. Ajayi, O. C. Metibemu, A. Y. Balogun, T. O. Oladoyinbo, and O. O. Olaniyi, “Adversarial Threats to AI-Driven Systems: Exploring the Attack Surface of Machine Learning Models and Countermeasures,” *Journal of Engineering Research and Reports*, vol. 27, no. 2, pp. 341–362, Feb. 2025, doi: 10.9734/jerr/2025/v27i21413.

[22] A. Georgiadou, S. Mouzakitis, and D. Askounis, “Assessing mitre att&ck risk using a cyber-security culture framework,” *Sensors*, vol. 21, no. 9, May 2021, doi: 10.3390/s21093267.

[23] J. R. Venable, J. Pries-Heje, R. L. Baskerville, J. R. ; Venable, J. ; Pries-Heje, and R. Baskerville, “Choosing a Design Science Research Methodology,” 2017. [Online]. Available: https://aisel.aisnet.org/acis2017/112

[24] L. S. Goecks, M. De Souza, T. P. Librelato, and L. R. Trento, “Design Science Research in practice: Review of applications in Industrial Engineering,” *Gestao e Producao*, vol. 28, no. 4, 2021, doi: 10.1590/1806-9649-2021v28e5811.

Adaptive Access Control Using Threshold Cryptography and Dynamic Policy Management

Aldiyar Ismailov
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, North Macedonia
aldiyar.ismailov@stu.khas.edu.tr

Panche Ribarski
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, North Macedonia
pance.ribarski@finki.ukim.mk

Mehmet Aydin
Department of Management Information Systems
Boğaziçi University
Istanbul, Turkey
mehmet.aydin@khas.edu.tr

***Abstract*—Traditional access control systems, reliant on static credentials like PINs or RFID cards, are ill-suited for the complex social dynamics of modern smart buildings. These environments involve a diverse and transient set of individuals whose interactions are unpredictable and cannot be managed by rigid, predefined rules. This paper confronts this challenge by introducing a proof-of-concept (PoC) for an adaptive access control system that intelligently manages trust in these dynamic scenarios. Our framework integrates four key technologies, chosen specifically to address the limitations of static systems. To understand the crucial context of a visit, a Natural Language Processing (NLP) module interprets a visitor's spoken purpose, providing structured data on their intent. To create an evolving security posture, this data informs Dynamic Trust and Anomaly Detection modules, which maintain a persistent trust score for each visitor based on their behavior over time. The system's decision-making core translates this contextual data into a quantifiable security response. The Dynamic Policy Module uses the NLP context and real-time trust score to dynamically set the voting threshold (t) required for entry. To ensure authorization is both secure and democratic, a Cryptographic Voting Module uses Shamir's Secret Sharing (SSS). This choice facilitates decentralized decision-making among residents, eliminating single points of failure and reliance on a single trusted authority. Our results provide quantifiable evidence of its effectiveness: in response to high-risk scenarios, the system automatically hardened its security posture, increasing the required voting threshold to a 75% supermajority. This enhanced security is achieved with acceptable performance costs for real-time interaction; the core intelligence modules (NLP, Trust, and Policy) add a consistent overhead of only 10-50 ms per request, with the cryptographic core maintaining real-time viability for communities of up to 120 residents.**

***Index Terms*—Access Control, Threshold Cryptography, Natural Language Processing, Trust Management, Anomaly Detection**

I. Introduction

Traditional access control systems, which rely on static credentials like Personal Identification Numbers (PINs) or RFID cards, are increasingly insufficient for the dynamic security and social landscapes of modern smart buildings. These environments are characterized by a constantly changing mix of individuals whose legitimacy cannot be effectively assessed by simple, pre-defined rules. These individuals can include residents, guests, delivery personnel, and service technicians. This creates a significant gap for systems that can understand the context of an access request and make risk-based decisions, rather than forcing a trade-off between rigid security and unearned trust. Current systems particularly lack mechanisms for leveraging collective resident input for secure, decentralized decision-making.

In this paper, we introduce a novel framework for adaptive access control that directly addresses these challenges. Our primary contribution is the integration of four key technologies to create an intelligent, context-aware system: **1)** A Natural Language Processing (NLP) module to interpret the intent and purpose of a visitor's request, **2)** A anomaly detection module to analyze the request against historical logs to identify suspicious patterns, **3)** A dynamic trust module that continually adjusts a visitor's trust score based on their interaction history and detected anomalies, and **4)** A threshold cryptography core that enables decentralized access authorization through a voting mechanism among residents. This unique combination allows the system to make fine-grained security decisions based on the nuanced context of an interaction, not just on a static credential.

We designed and implemented a modular, web-based proof-of-concept (PoC) system to validate the feasibility and performance of our framework. Our evaluation demonstrates that the system can effectively adapt its security posture in response to perceived risk. For instance, when presented with a high-risk visitor, the dynamic policy engine automatically increased the required cryptographic voting threshold to a **75% supermajority**. This quantifiable enhancement to security is achieved with low performance overhead, as the NLP and trust modules add a consistent processing time of only **10-50 ms** per request, proving the approach is viable for real-time applications.

II. Related Work

Our research builds upon three core domains: threshold cryptography, Natural Language Processing (NLP) in security, and adaptive access control. This section reviews the state of the art in these areas to contextualize our contribution.

A. Threshold Cryptography for Access Control

Threshold cryptography, founded on primitives like Shamir's Secret Sharing (SSS) which uses polynomial interpolation for security [1], provides a robust framework for distributed and fault-tolerant decision-making. While widely applied in areas like e-voting [2], its use in access control is a more specific domain. The foundational concept of using threshold signatures for decentralized access control was validated by Saxena et al. [3]. Their work focused on environments like mobile ad hoc networks (MANETs) and peer-to-peer systems, which are characterized by node mobility and a flat trust structure [3]. Our smart building scenario presents different challenges, involving a more hierarchical trust model (residents versus visitors) and a physically constrained environment. Recent advancements in the field have focused on creating more flexible and scalable schemes. For example, Dynamic-FROST accommodates changes in the participant group and threshold by combining the FROST signature scheme with a dynamic proactive secret sharing protocol, avoiding the need for a trusted third party during committee updates [4]. Our work aims to apply these advanced cryptographic principles to our novel, real-time smart building context.

B. Natural Language Processing in Security

In the security domain, NLP is primarily used for the offline analysis of unstructured data. A significant body of research focuses on extracting access control policies from static, natural language requirement documents [5], [6]. While powerful, these approaches are applied to project artifacts before deployment, rather than to the live, unpredictable requests from users. Other research has explored using NLP to detect intrusions by analyzing machine-generated IoT device logs [7]. Our framework extends this paradigm by employing NLP not for passive, offline analysis, but as an **active, real-time component** of the access control decision. By interpreting a visitor's spoken intent at the moment of the request, the NLP module provides the critical, live context that is missing in traditional systems.

C. Adaptive and Trust-Based Access Control

The limitations of static, perimeter-based security have led to the development of adaptive access control systems, often guided by the principles of Zero Trust Architecture ("never trust, always verify") [8]. A key component of these systems is dynamic trust management, where an entity's trust score evolves based on its behavior and interaction history [9]. Frameworks like the risk-based model proposed by Atlam et al. are particularly relevant; their model estimates security risk using inputs such as user context, resource sensitivity, and action severity to make a dynamic access decision [10]. Our work builds directly on this concept by using the intent and entities derived from our NLP module as a primary source of rich, real-time user context.

While significant research exists in each of these areas, their integration remains a key challenge. Adaptive access control frameworks often rely on predefined attributes or simple sensor data but rarely incorporate the rich, semantic context from a live human interaction. Similarly, applied threshold cryptography has focused on protocol efficiency but less on how the cryptographic parameters themselves can be dynamically and securely adjusted by external, non-cryptographic modules. Our work is positioned directly at the intersection of these domains, addressing this integration gap by using the output of real-time NLP analysis to directly influence a dynamic trust score, which in turn sets the parameters for a cryptographically-secured, decentralized voting mechanism.

III. System Design and Methodology

This work follows the **Design Science Research (DSR)** methodology, where the primary artifact is a proof-of-concept (PoC) system designed to validate our proposed framework for adaptive access control. The system was developed using an iterative, modular approach, allowing for the individual testing and refinement of each core component before integration.

A. System Architecture and Data Flow

The system is architected as a modular, client-server web application that processes access requests through a logical pipeline. As illustrated in Fig. 1, this pipeline is designed to enrich a visitor's request with layers of context before a final, cryptographically-secured decision is made. A visitor's text-based request is first processed by the Natural Language Processing (NLP) module to extract structured data such as intent and key entities. Concurrently, the Anomaly Detection module analyzes the request against historical logs to identify suspicious patterns. This information feeds into the Trust and Policy Management module, which updates the visitor's evolving trust score and uses this score, along with the NLP context, to dynamically determine the required voting threshold (t) for access. Finally, the Cryptographic Voting module initiates the authorization process with the residents based on this dynamically set policy.

B. Core Module Design

The novelty of our framework lies in the interaction between its core modules. Each was designed with specific goals for performance, security, and configurability.

NLP Module: The NLP module was designed using a hybrid approach to transform unstructured text into actionable data. We chose the pre-trained `spaCy` library for its efficient and, crucially, deterministic Named Entity Recognition (NER), providing reliable extraction of entities like names and organizations. This choice was made over larger, non-deterministic Large Language Models to ensure predictable and fast responses suitable for a real-time security application. This is

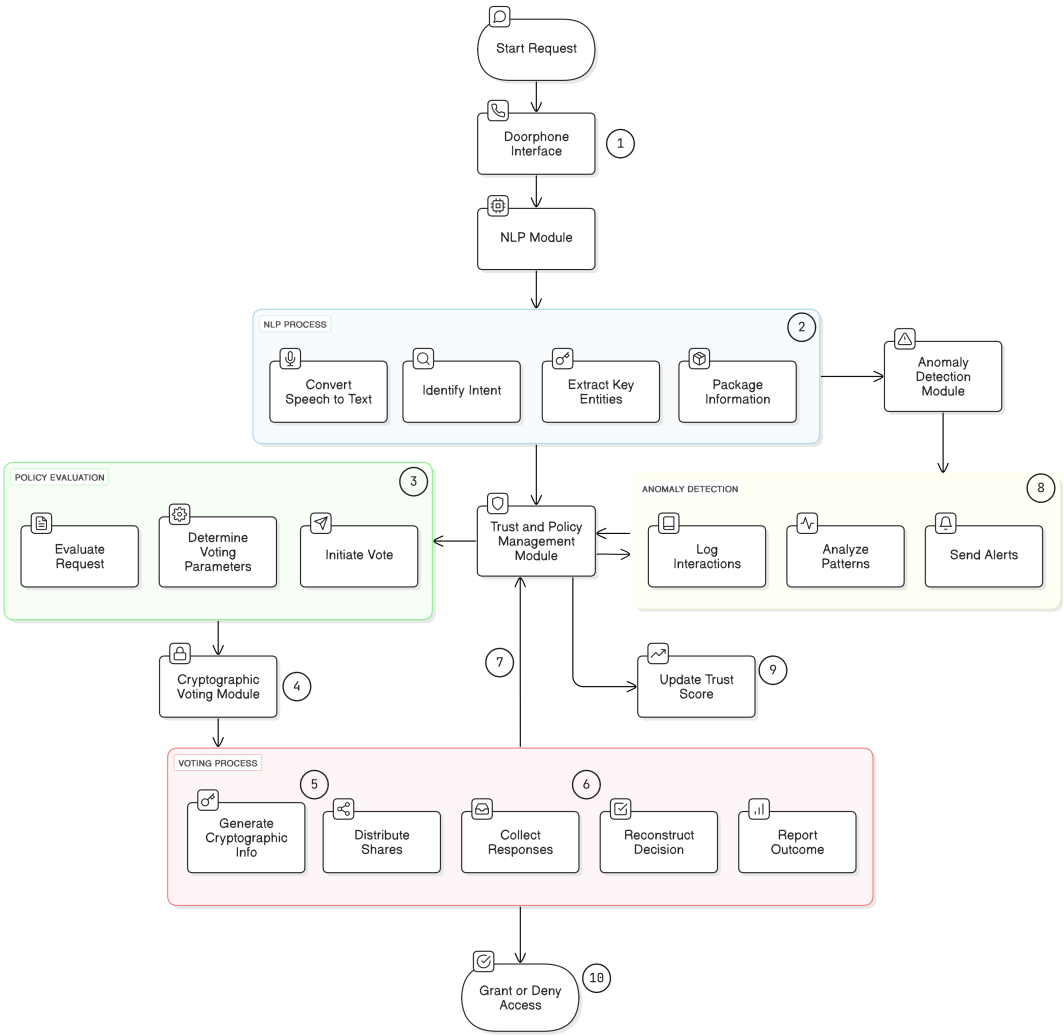


Fig. 1. The request processing pipeline of the adaptive access control system, from initial visitor input to the final cryptographic vote.

complemented by a highly configurable, rule-based system for intent classification. This layer uses lemmatization to match variations of keywords (e.g., 'delivering' becomes 'deliver') against a dynamically loaded vocabulary, allowing it to handle specific, pre-defined intents such as `delivery_request`, `guest_access_request`, and `service_request`.

Dynamic Trust and Policy Module: This module serves as the system's decision-making core and operates in two stages. First, it maintains a persistent trust score for each visitor in a database, which serves as a long-term reputation metric. This score is not static; it is dynamically adjusted by other system components, receiving a boost for pre-verification checks (e.g., when a visitor's request correctly identifies a resident) and penalties from the anomaly detection module. Second, the policy engine ingests this real-time trust score and the NLP-derived context to interpret a set of hierarchical, configurable rules. The logic is designed to prioritize risk; it first checks for high-trust or low-trust score overrides, which can immediately set a minimal or supermajority threshold, respectively. If no override is triggered, it falls back to intent-specific rules, ensuring the system's response is both adaptive and predictable.

Cryptographic Voting Module: To facilitate secure, decentralized authorization, the cryptographic module was designed with a swappable backend to allow for experimentation with different security models. The PoC implements two schemes representing a trade-off between simplicity and security. The first, a centralized trusted-dealer model using Shamir's Secret Sharing (SSS), serves as a functional baseline where the server generates and reconstructs the secret. The second, a simulation of the FROST Threshold Signature Scheme (TSS), demonstrates the principles of a more robust, decentralized model that provides enhanced security by ensuring the master private key is never reconstructed on any single machine. This dual implementation allows for a direct performance and security comparison between these two approaches.

C. Validation Approach

To evaluate the system's effectiveness and performance, we developed a dedicated benchmarking script that automates the execution of the entire server-side processing loop. This script systematically varies the number of participating residents and the active cryptographic scheme, measuring the execution time of each individual module to provide a quantitative analysis of performance trade-offs and scalability limits. The full source code for the PoC system, including all modules and testing scripts, is available for review.[1]

IV. Evaluation and Results

We evaluated our proof-of-concept (PoC) system to validate both its adaptive security logic and its performance characteristics under various scenarios. The evaluation was conducted using a suite of automated scripts to ensure reproducible results.

A. Adaptive Logic Validation

To verify that the system could dynamically adjust its security posture, we conducted a series of integration tests simulating different visitor requests. As summarized in TABLE I, the system correctly adjusted its cryptographic voting policy in response to varying contextual inputs derived from the NLP and Trust modules, confirming that the core logic of the policy engine is sound.

1) Qualitative Scenario Analysis: To illustrate the system's adaptive logic in practice, we analyze two contrasting scenarios from our tests. In the "Standard Courier" scenario (Scenario 1), the NLP module correctly identified the `delivery_request` intent and the courier's affiliation. The Policy Module, recognizing this as a routine, low-risk event, applied a specific rule that reduced the voting threshold (t) to a minimal value, prioritizing convenience and minimizing the burden on residents.

Conversely, consider the "Low-Trust Override" scenario (Scenario 6). A visitor with a pre-existing low trust score of 0.15 initiated a request. The Policy Module, upon receiving this score, immediately triggered the configured 'low_trust' override rule. This action bypassed the standard analysis of the visitor's NLP-derived intent and directly applied a supermajority policy, calculating a required threshold of $t = 15$ for the 20-resident committee. This demonstrates the system's ability to prioritize risk history over immediate contextual data to enforce a hardened security posture.

B. Performance and Scalability Analysis

To quantify the performance trade-offs of our integrated approach, we developed a benchmarking script to measure the server-side execution time of each module under an increasing number of residents (n). We conducted tests for two distinct scenarios: a low-risk case with a minimal, constant threshold, and a high-risk case where the threshold scales with the committee size.

The results, presented in Fig. 2 and Fig. 3, reveal two key findings. First, in both scenarios, the "intelligence" modules (NLP, Trust, and Policy) exhibit a minimal and constant processing overhead of **under 50 ms**, confirming their high scalability. Second, the primary performance bottleneck is the cryptographic core. As shown in Fig. 3, for high-risk scenarios where $t \approx 0.75n$, the total execution time scales steeply. Our benchmarks indicate that the system maintains real-time performance (under one second) for communities of up to approximately **120 residents with FROST** and **180 with SSS**.

A comparison of the two graphs powerfully illustrates that the required cryptographic threshold (t) is the dominant factor in system performance. In the standard-risk scenario (Fig. 2), where t is low and constant, the overall processing time remains flat and low, scaling minimally even as the number of residents increases. This visually reinforces that the scalability challenge is not merely the number of participants, but the security level demanded by the dynamic policy.

C. Discussion

The results of our evaluation have several key implications for the design of future access control systems, particularly concerning the trade-offs between intelligence, security, and usability, as well as the limitations of this study.

1) Implications of Findings: Our results have two key implications. First, the minimal, constant overhead of the intelligence modules suggests that adding more sophisticated contextual analysis (e.g., more complex NLP or anomaly detection rules) is feasible without creating a performance bottleneck. This confirms that the computational cost of "smart" features is not a barrier to real-time deployment. Second, the cryptographic core's clear role as the scalability limit highlights the critical importance of protocol choice for real-world deployments. While SSS was faster in our simulation, a full TSS like FROST provides superior security guarantees by avoiding secret reconstruction, presenting a crucial trade-off for system designers.

2) Security and Usability Trade-offs: Our framework fundamentally shifts the security model from a single point of failure (a lost key or a compromised server) to a distributed consensus model. This enhances resilience against many traditional attacks. However, this introduces a significant usability challenge: **voting fatigue**. In a high-traffic environment,

[1]The full source code is available at: https://github.com/IAvidI/voting

TABLE I
SUMMARY OF AUTOMATED TEST SCENARIOS AND ADAPTIVE POLICY RESULTS (FOR A COMMITTEE OF N=20)

#	Scenario Name	Key Input / Condition	Expected Threshold (t)	Result
1	Standard Courier	Visitor purpose: "UPS for apartment 5C"	2	Pass
3	Standard Guest	Visitor purpose: "visit my friend, Alice"	12	Pass
5	High-Trust Override	Visitor with Trust Score = 0.90	3	Pass
6	Low-Trust Override	Visitor with Trust Score = 0.15	15	Pass
9	Anomalies Trigger Override	Visitor with multiple anomalies detected; Trust Score = 0.05	15	Pass

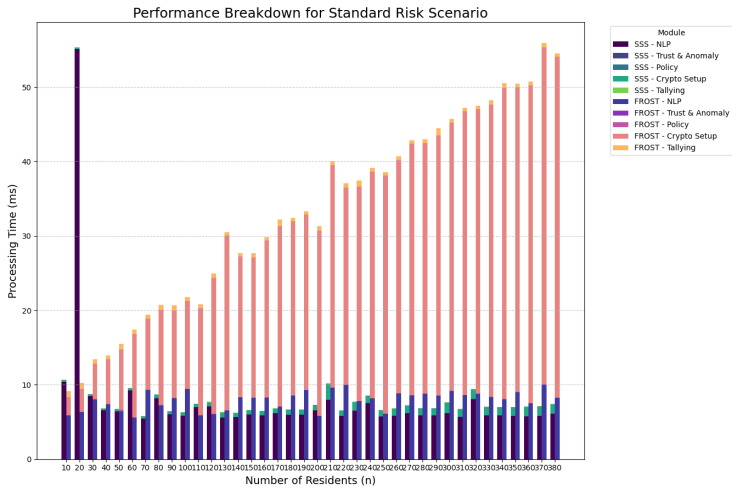


Fig. 2. Performance breakdown for the Standard-Risk scenario, where the voting threshold t is low and constant. Overall execution time remains low and scales minimally, dominated by the constant overhead of the intelligence modules and a small, fixed cryptographic cost.

residents may be inundated with requests, potentially leading them to approve requests without proper scrutiny or ignore them altogether. This human factor is a critical consideration, suggesting that in a production system, the dynamic policy engine would need to be carefully tuned to minimize unnecessary votes for trusted, low-risk scenarios.

3) Limitations: It is important to acknowledge the limitations of this proof-of-concept study. First, the system was evaluated in a simulated environment that does not account for real-world factors like network latency between the server and residents. Second, the intelligence modules for policy and anomaly detection are currently rule-based. While configurable, they may not capture the full complexity of real-world scenarios that a machine learning approach might. Finally, our implementation of the FROST protocol is a simulation for demonstrating the principles of a TSS; it does not include a true, interactive Distributed Key Generation (DKG) phase.

V. CONCLUSION

In this paper, we presented and evaluated a novel framework for adaptive access control that integrates Natural Language Processing, dynamic trust management, and threshold cryptography to address the security challenges of dynamic smart building environments. Our proof-of-concept system successfully demonstrated its ability to adapt its security posture in response to contextual risks, enforcing policies ranging from minimal thresholds for routine requests to a **75% supermajority** for high-risk visitors. This adaptive security was achieved with a minimal performance overhead of **10-50 ms** from the intelligence modules, confirming the framework's viability for real-time applications in communities of up to 120-180 residents. Our work shows that it is practical to replace static, credential-based access control with a more resilient paradigm based on context-aware, decentralized authorization. Future work will focus on implementing a full Threshold Signature Scheme with Distributed Key Generation and exploring the use of machine learning models to enhance the policy and anomaly detection engines.

VI. FUTURE WORK

While this paper demonstrates the feasibility of our proposed framework, several exciting avenues for future research exist:

- **Full TSS Implementation:** The next logical step is to replace the simulated FROST implementation with a full, cryptographically secure Threshold Signature Scheme. This would involve implementing a true, interactive Distributed Key Generation (DKG) protocol among resi-

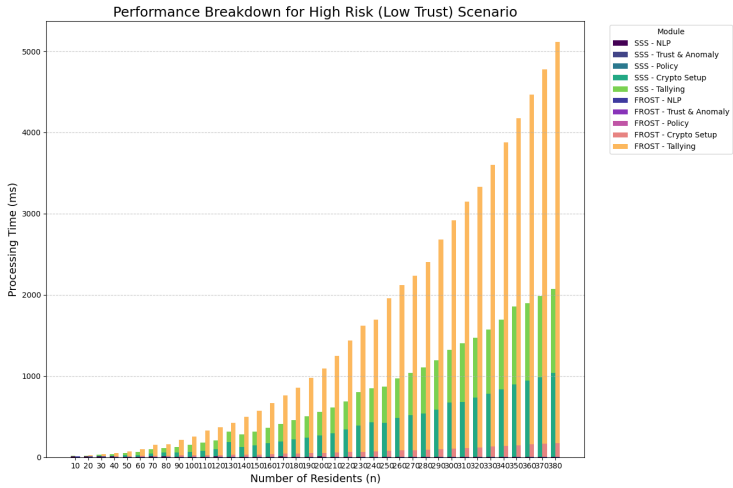


Fig. 3. Performance breakdown for the High-Risk scenario, showing server-side execution time as a function of the number of residents (n), where $t \approx 0.75n$. The cryptographic modules (Crypto Setup and Tallying) are the primary drivers of execution time, making the security threshold the main scalability bottleneck.

dents, completely eliminating the server as a trusted party for key management.

- **Machine Learning-Based Intelligence:** The existing rule-based policy and anomaly detection modules could be replaced with trained machine learning models. An isolation forest or autoencoder could identify more subtle anomalies, while a classification model could learn more sophisticated policy rules from historical data.
- **Formal Usability Studies:** To address the practical challenges of a resident voting system, formal user-centric studies should be conducted. This would be invaluable for assessing the usability of the interface and developing strategies to mitigate potential ”voting fatigue” in a busy environment.

REFERENCES

[1] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, pp. 612–613, 11 1979.

[2] Y. X. Kho, S. H. Heng, and J. J. Chin, “A review of cryptographic electronic voting,” *Symmetry*, vol. 14, 5 2022.

[3] N. Saxena, G. Tsudik, and J. H. Yi, “Threshold cryptography in p2p and manets: The case of access control,” *Computer Networks*, vol. 51, pp. 3632–3649, 8 2007.

[4] A. Cimatti, F. D. Sclavis, G. Galano, S. Giammusso, M. Iezzi, A. Muci, M. Nardelli, and M. Pedicini, “Dynamic-frost: Schnorr threshold signatures with a flexible committee,” *Journal of Mathematical Cryptology*, vol. 19, 1 2025. [Online]. Available: https://www.degruyterbrill.com/document/doi/10.1515/jmc-2024-0045/html

[5] S. H. Jayasundara, N. A. G. Arachchilage, and G. Russello, “Sok: Access control policy generation from high-level natural language requirements,” *ACM Computing Surveys*, vol. 57, pp. 1–37, 12 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3706057

[6] J. Slankas and L. Williams, “Access control policy extraction from unconstrained natural language text,” *Proceedings - SocialCom/PASSAT/BigData/EconCom/BioMedCom 2013*, pp. 435–440, 2013.

[7] R. Hegde, S. Keerthana, R. M. Sripriya, S. Raghu, and K. V. Kavin, “Natural language processing based intrusion detection interface for iot devices,” *IEEE International Conference on Recent Advances in Science and Engineering Technology, ICRASET 2024*, 2024.

[8] Y. He, D. Huang, L. Chen, Y. Ni, and X. Ma, “A survey on zero trust architecture: Challenges and future trends,” *Wireless Communications and Mobile Computing*, vol. 2022, p. 6476274, 1 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1155/2022/6476274 https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/6476274 https://onlinelibrary.wiley.com/doi/10.1155/2022/6476274

[9] W. Jiang, Z. Lin, and J. Tao, “An access control scheme for distributed internet of things based on adaptive trust evaluation and blockchain,” *High-Confidence Computing*, vol. 3, p. 100104, 3 2023.

[10] H. F. Atlam, A. Alenezi, R. J. Walters, G. B. Wills, and J. Daniel, “Developing an adaptive risk-based access control model for the internet of things,” *Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017*, vol. 2018-January, pp. 655–661, 7 2017.

AI-Driven Code Obfuscation: Enhancing Software Security using Machine Learning

Edra Tabaku
SRH Heidelberg University of Applied Sciences / Kadir Has University
Berlin, Germany
edratabaku@gmail.com

Prof. Dr. Alexander Iliev
M.Sc. Computer Science - Big Data & Artificial Intelligence
SRH Heidelberg University of Applied Sciences
alexander.iliev@srh.de

Dr. Tuğçe Ballı
Dept. Management Information Systems
Kadir Has University
Istanbul, Türkiye
tugce.balli@khas.edu.tr

Kendrick Bollens
School of Technology and Architecture
SRH Heidelberg University of Applied Sciences
Berlin, Germany
kendrick.bollens@srh.de

***Abstract*—Code obfuscation is a crucial technique in software security, making code harder to understand and reverse engineer while maintaining its functionality. This research conducts a comparative and experimental study of traditional rule-based code obfuscation tools that use variable renaming, dead code injection, control flow flattening and other techniques, and machine learning tools such as LLMs, neural networks and transformers to obfuscate JavaScript code. The study aims to evaluate the current state of AI code obfuscation and address its limitations.**

***Index Terms*—code obfuscation, artificial intelligence, machine learning, JavaScript.**

I. Introduction

The increasing number of software systems being used in every domain in the modern digital era has reshaped the approach towards opportunities and threats within the cyber domain. As the reliance on software increases, so does the necessity of protecting software from being exploited and protecting intellectual property[8]. In this context, code obfuscation became a crucial tool in cybersecurity to protect intellectual property of software and make code harder to reverse engineer[22]. Code obfuscation is a transformation of source code into a form that is syntactically valid yet semantically harder for humans to understand, therefore complicating the process of extracting and utilizing important algorithms and procedures that are part of the software product. By significantly increasing the time and effort required for attackers to reverse-engineer the code, obfuscation creates a significant barrier for potential attackers and enhances overall software security[22]. Javascript, in a dynamically typed, compiled "just-in-time" programming language, widely deployed and used for client-side scripting, with enormous amounts of unprotected source code being sent over the network and being available to the end user. Its interpretive nature and exposure in browser environments makes it particularly vulnerable to code inspection and tampering, therefore presenting unique challenges for obfuscation[20]. While code obfuscation has been used for many years as a defensive mechanism against reverse engineering and intellectual property theft, the growth of artificial intelligence (AI) and large language models (LLMs) introduces new possibilities for automating this process. Traditional Javascript obfuscators, such as Javascript Obfuscator and UglifyJS rely on rule-based transformations (identifier renaming, dead code insertion etc.[5] - effective but limited in terms of adaptability), recent advances in large language models suggest that AI-driven approaches can generate more resilient and semantically preserved obfuscated code. Despite the growing interest in AI-driven code obfuscation, a comprehensive comparative analysis between existing traditional tools and AI-driven methods remains scarce. Existing studies have primarily focused on the detection and deobfuscation of obfuscated code, with less emphasis on evaluating the effectiveness of obfuscation itself. This research aims to fill this gap by attempting to evaluate the current state of AI-driven obfuscation and comparing it to the capabilities of traditional tools.

A. Key definitions

Code obfuscation

Mathematically, we can show obfuscation with this definition: Code obfuscator is defined as a function f that transforms original source code P into an obfuscated version of source code P'. Formally, this can be represented as $f : P \to P'$. Where P is the space of all possible programs and P' is the space of all possible obfuscated programs. The obfuscation function f must ensure that the obfuscated program P' behaves identically to the original program P for all inputs. So, we will have:[19]

$$\forall x \in X,\ P(x) \simeq P'(x)$$

Types of obfuscation techniques

Initially, obfuscation techniques were simplistic and focused

on basic and simple transformations such as renaming variables and functions. Over time, they have advanced to include more sophisticated techniques in order to increase the difficulty of reverse engineering efforts. [22] defines the main three obfuscation techniques as:

- Lexical obfuscation: Consists of renaming identifiers such as variables, functions and classes to obscure their meaning. By replacing these identifiers with meaningless strings, this technique aims to make it more challenging to understand the code's purpose. This technique has minimal impact on the execution performance of the obfuscated code.
- Control flow obfuscation: Modifies a program's logical structure by adding dead code, false conditionals, loops in order to obscure the program's control flow. The goal is to make it difficult for attackers to trace the execution path of the program.
- Data obfuscation: Encoding and encrypting data values within code to make them unreadable, mostly used for credentials, API keys and configuration settings. At runtime, this data is decoded or decrypted, allowing the program to run normally while still protecting critical information.

Artificial intelligence, neural networks and transformers In broad terms, artificial intelligence comprises of any technique that allows computers to mimic human behavior and reproduce human decision-making to solve tasks with no or minimal human intervention[14].

Artificial neural networks, or neural networks consist of a mathematical representation of connected processing units called artificial neurons. Similarly to synapses in a brain, each connection between neurons transmits signals with different strengths (weights) that are continuously adjusted during the learning process. These neurons are organized into networks within different layers. The number of layers and neurons, among other choices such as the learning rate or activation functions should be set for the model as they constitute the model's hyperparameters[14].

Recurrent Neural Networks (RNNs) are a type of neural network architecture that is mainly used to detect patterns in sequential data[24]. This architecture allows internal feedback loops and therefore enables sequential pattern learning. Simple RNN architectures suffer from vanishing and exploding gradients.

The **transformer** is a model architecture that relies completely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence[28].

Supervised Learning and Reinforcement Learning **Supervised learning** is the paradigm where models are trained on labeled datasets, consisting of input-output pairs, to learn a mapping from inputs to outputs (e.g., X$\rightarrow$Y). The model aims to generalize from training examples to accurately predict unseen data. It optimizes a loss function (empirical risk minimization) and may incorporate regularization to balance the bias–variance trade-off [10]. The pairs of input and output data in the training set are then used to calibrate the open parameters of the ML model. Once the model has been successfully trained, it can be used to predict the target variable y given new or unseen data points of the input features x[14].

Reinforcement learning is a learning paradigm where an agent interacts with an environment, makes decisions (actions), receives feedback in the form of rewards, and learns a policy that maximizes cumulative reward over time. Instead of providing input and output pairs, we describe the current state of the system, specify a goal, provide a list of allowable actions and their environmental constraints for their outcomes, and let the ML model experience the process of achieving the goal by itself using the principle of trial and error to maximize a reward[14].

B. Research Gap

Previous researches has prioritized AI-driven deobfuscation over obfuscation, a focus largely driven by the critical need to counter malicious code in the field of cybersecurity. Artificial intelligence has been used to automate parts of the deobfuscation process that traditionally require manual effort. AI models can help in simplifying complex structure and therefore reducing the time and effort required to understand and deobfuscate code[22]. The dual nature of obfuscation, which is frequently used by malicious actors to hide malware and exploits, posing a direct threat to networks and users[15][3], which introduces a layer of ethical complexity that deobfuscation, as a countermeasure, does not. This conflict between creating a defensive tool and a potential weapon may explain the disparity in research.

Beyond ethical considerations, the technical challenges of generative obfuscation are substantial. Obfuscation does not necessarily follow a clear ruleset, making it a more open-ended problem and, therefore, fundamentally harder to produce new and effective obfuscation than to train it to detect known patterns. The diversity and complexity of obfuscation techniques mean that models need varied and extensive training data to be effective and must be able to generalize across different types of obfuscation which can be difficult given the lack of standardization in obfuscation methods[22]. An effective obfuscator must balance multiple metrics including potency, resilience, stealth, and cost[22]. Satisfying all of these criteria can be a technical challenge. Previous research highlights the difficulty of achieving 'semantic elasticity', where the obfuscated code retains its intended behavior while also being less readable[27]. Progress in research is further hindered by the lack of standardized benchmarks and evaluation metrics for AI-driven obfuscation. [19]

C. Research Questions

1. Can machine learning models effectively generate functionally correct JavaScript obfuscation?

In seeking to answer this question, we prepare a few experiments using different models in order to understand the

current state of AI obfuscation. This investigation aims to unravel the limitations of AI in terms of code obfuscation. Understanding these limitations is crucial for developing new approaches to code obfuscation.

2. How does the quality of AI-generated JavaScript obfuscation compare to a traditional rule-based tool?

This research entails exploring the differences between traditional, rule-based tools with AI obfuscators comparing them across different metrics. By comparing them, the research intends to uncover the advantages and disadvantages of each approach.

II. Related Work

Code obfuscation is a vital defensive mechanism, aiming to transform source code into a form that is significantly harder to understand and analyze, thereby challenging these malicious activities. By significantly increasing the time and effort required for attackers to reverse-engineer the code, obfuscation creates a significant barrier for potential attackers and enhances overall software security[22]. The application of LLMs to security-oriented code transformation tasks remains relatively unexplored, with important questions about their effectiveness and optimal usage patterns[27] . The application of machine learning to generate obfuscated code has evolved with the premise of treating code as a form of natural language, sequences of characters and tokens that can be transformed from one form to another while preserving underlying semantics.

A. Main sources

[22] start their research by introducing the development of code obfuscation and the various traditional techniques used. Their research highlights the growing application of machine learning in code deobfuscation, and how this application has simplified reverse engineering and efficiency. [27] perform an empirical study on the ability of LLMs to obfuscate Python source code and also introduce a new metric (semantic elasticity) to measure the quality degree of obfuscated code. They experiment with various LLMs and compare them. The study provides interesting findings such as the tendency of LLMs to reduce code complexity rather than increase it when performing obfuscation, contradictory to traditional approaches. [19] perform a systematic analysis of large language models into Assembly code obfuscation. They obfuscate Assembly code through three different obfuscation techniques: dead code insertion, register substitution and code flow alteration. To compare the performance of the selected LLMs they consider two metrics: character-wise Delta entropy and cosine similarity. [6] uses neural networks to convert plaintext source code into a cipher text, specifically using RNN encoder-decoders or sequence to sequence models. To compare the code samples (original vs. obfuscated) Levenshtein Distance is employed. The findings show significant improvement in stealth and execution cost compared to existing obfuscation methods.

B. The foundational role of Neural Networks

Several works[6][20] have leveraged deep learning architectures to automate and optimize obfuscation. DeepObfuscode[6] proposes a novel methodology of using a text-based recurrent neural network (RNN) encoder-decoder model to generate a "cipher text" of the original source code. This is a complex process as the model's architecture, with its randomly-set weights and propagating structure, compounds the randomness factor in the creation of the obfuscated code. A secondary RNN model is trained to deobfuscate the cipher text, and the resulting weights serve as the deobfuscation key. While unable to produce directly executable code just from obfuscation, this framework treats obfuscation as a sequence-to-sequence translation task, which is a core concept in natural language processing.

C. The rise of Large Language Models

The rise of LLMs has dramatically accelerated research in this area. Their deep contextual understanding and ability to process large amounts of data have made them a considerable candidate for obfuscation. A systematic analysis[19] shows evidence of the capability of LLMs to obfuscate assembly code. This study evaluates LLMs on different obfuscation techniques such as dead code insertion, register substitution and control flow changes. CODECIPHER[16] introduces a novel method to protect code privacy from LLMs. It works by transforming the LLM's embedding matrix to create a "token-to-token confusion mapping". While focusing on code translation rather than obfuscation, CoTran[13] uses a methodology that can be applied to obfuscation tasks. It fine-tunes an LLM using reinforcement learning with external feedback from a compiler and symbolic execution. This feedback loop allows the model to learn how to produce functionally correct code, which is a requirement for any obfuscation system.

D. The need of new metrics

The evolution of obfuscation has necessitated the development of more nuanced evaluation metrics. Delta Entropy[19] is used to quantify the change in code complexity and diversity, cosine similarity[19] is used to ensure that the similarity between the original and obfuscated code is within a plausible range; these metrics move beyond code size or runtime to measure the efficacy of obfuscation. Semantic Elasticity[27] measures the quality of obfuscated code by balancing structural transformations with the preservation of functional integrity. The findings of the work where the metric is used show the tendency of LLMs to reduce the cyclomatic complexity of code when obfuscating rather than increasing it.

III. Methodology

As the primary focus of this study is to gain an understanding of the effectiveness of AI-driven code obfuscation methods compared to the traditional ones, an experimental and comparative approach is used. This involves the development and evaluation of AI-based obfuscation systems, alongside a review of existing literature to support the experimental findings.

A. Dataset Selection

For the experimental evaluation, a subset of JavaScript code samples was obtained from the CodeSearchNet dataset on Hugging Face. CodeSearchNet provides a dataset that contains code samples in multiple programming languages, designed for machine learning research on source code. The dataset consists of samples collected from publicly available, non-form, open-source repositories on GitHub.[12] The creators applied filtering and partitioning strategies to reduce redundancy, such as ensuring that code from the same repository appears in only one split (train, test, or validation). From the total of 123,889 JavaScript scripts available, a corpus of 39,162 scripts was utilized for this study. This reduced subset was chosen due to computational resource constraints while considering data diversity for model training and evaluation. The code implementation for retrieving the dataset is available in the GitHub repository.

B. Evaluation Metrics

To evaluate the obfuscation quality the following metrics have been chosen, inspired by prior work on evaluating obfuscation techniques:

1. Variable Name Entropy is used to measure the lexical diversity and complexity of variable and function names. It is a variant of the Delta Entropy metric[19], which is used to quantify the change in code diversity between the original and obfuscated code pairs. A higher entropy value indicates a more successful lexical obfuscation.

2. Cyclomatic Complexity is a metric that quantifies the number of linearly independent paths through the source code.[17] Traditionally, obfuscators aim to increase this value to lower the readability of the code.

3. Functional Preservation is a fundamental requirement for any obfuscation technique. To evaluate this metric, a suite of unit tests was developed for each script to measure whether the obfuscated code maintains the same behavior as the original code. This metric is represented as a pass/fail flag for each script.

4. Time(cost) needed for obfuscation measures the computational overhead required for performing the obfuscation. Measuring performance is considered critical as high latency is not suitable for real-life applications and large-scale codebases.[22]

5. Levenshtein Distance quantifies the degree of textual change between the original and obfuscated code. This is a string metric that measures the minimum number of single-character operations required to transform one string into another.[21] For two strings *a* and *b* of length $|a|$ and $|b|$, the Levenshtein distance between the first *i* characters of *a* and the first *j* characters of *b* is defined as:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{min(i,j)=0,} \\ \min \begin{cases} lev_{a,b}(i-1,j)+1, \\ lev_{a,b}(i,j-1)+1, \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases} \quad (1)$$

where:

- $1_{(a_i \neq b_j)}$ is the indicator function, equal to 0 when $a_i = b_j$ and 1 otherwise
- The first case represents the distance if one string is empty
- The three options in the second case represent deletion, insertion, and substitution respectively

C. Experiment 1: Baseline Obfuscation with a Traditional Tool

The first experiment uses a traditional rule-based obfuscation tool: javascript-obfuscator, in order to establish a performance baseline. javascript-obfuscator is a free and open-source obfuscator for Javascript that applies a set of predefined transformations on the entire dataset, including control flow flattening, dead code insertion and string encoding. The results of this experiment are used as a point of comparison with the AI-drive approaches. The following code snippet performs three primarily transformations on inputCode (the original code snippet that we want to obfuscate):

Compaction - removing unnecessary whitespace and line breaks from the code to reduce code size

Control flow flattening - a transformation that alters the program's control flow by embedding logical statements in order to make it more difficult to understand execution order.

String array encoding - this replaces literal strings into references stored in a separate array.

```
const obfuscatedCode = JavaScriptObfuscator
.obfuscate(inputCode, {
    compact: true,
    controlFlowFlattening: true,
    stringArray: true
}).getObfuscatedCode();
```

The full code for obfuscation and evaluation is in the GitHub repository.

1) Results: Traditional obfuscators tend to increase the number of characters in the obfuscated output due to the insertion of conditional statements, string array encoding and dead code insertion. The figure plots the difference in number of characters between the original code snippet and its obfuscated version.

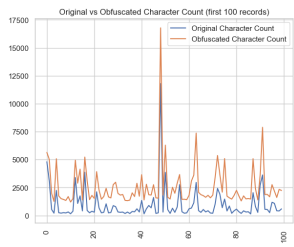


Fig. 1. Difference in number of characters between original and obfuscated code

On average, the character count increases by 1,580 characters, and the Levenshtein distance grows in proportion to the file's character count. The mean Levenshtein distance between the original and obfuscated versions is approximately 2,190.77, while the average time required for obfuscation is 13.69 milliseconds (ms). Additionally, the obfuscated code shows higher cyclomatic complexity, with an average increase of 4.07, and greater variable name entropy, with an estimated increase of 1.22.

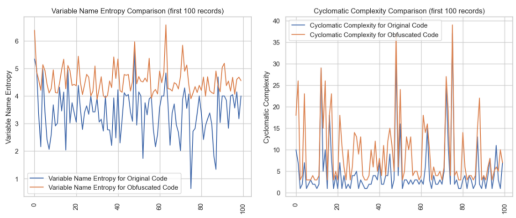


Fig. 2. Cyclomatic complexity and variable entropy changes

D. Experiment 2. LLM-based obfuscation

The second experiment explores the generative capabilities of a free and open-source large language model via prompt engineering. For this purpose, we used Ollama's Mistral model which is a free and open-source model. When using a simple prompt such as "Obfuscate the following javascript code", the model fails to provide a strong obfuscation and primarily performs variable renaming. However, when instructed to obfuscate JavaScript code while preserving functionality and increasing complexity, using a tailored prompt with defined obfuscation techniques, the model was able to generate more effective obfuscations.

1) Results: Consistent to prior research[27], the most noticeable change is the tendency of LLMs to reduce the character count, with outputs averaging 343.64 characters shorter than the original input. Obfuscation required a longer processing time, averaging 58.8 milliseconds, which increased proportionally with the input length. Similarly, the Levenshtein Distance between the original and obfuscated code also increased with input size, with an average of 558.57.

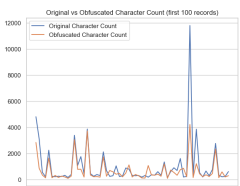


Fig. 3. Difference in number of characters between original and obfuscated code

Another interesting difference is the decrease in terms of cyclomatic complexity, being -0.26 on average while variable entropy exhibited only a small increase of 0.24.

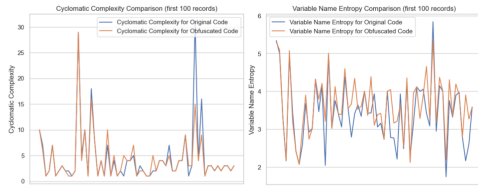


Fig. 4. Cyclomatic complexity and variable entropy changes

Despite these transformations, not all outputs were syntactically correct or functionally equivalent: 83.6% of the obfuscated code preserved the original functionality, whereas 16.4% either produced different results or failed unit tests.

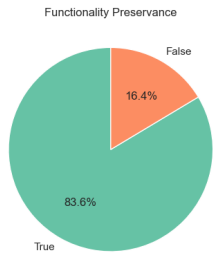


Fig. 5. Pie chart for functionality preservation

E. Comparative analysis

The comparative analysis conducted on 39,162 JavasScript scripts demonstrates that both traditional obfuscation and obfuscation produced by Large Language Models are able to produce functional code with lower human readability. However, the two differ in terms of performance and code structure. Traditional obfuscators employ aggressive techniques designed to maximize confusion and prevent reverse engineering. These techniques often include string encoding, insertion of self-modifying logic, control flow flattening using opaque predicates or mathematical obfuscation, and comprehensive variable/function renaming. The result is a transformation that, while behaviorally identical to the original, is extremely difficult for a human to interpret or refactor. The structural modifications introduced by these tools tend to be consistent and rule-based, leading to code that is unreadable but follows certain obfuscation patterns recognizable to experts. In contrast, LLM-generated transformations take a very different approach. These models, when prompted to rewrite or obfuscate code, tend to retain much of the original structure

and logic flow. While comments are generally removed and some variable names are altered, many names remain either fully or partially recognizable.

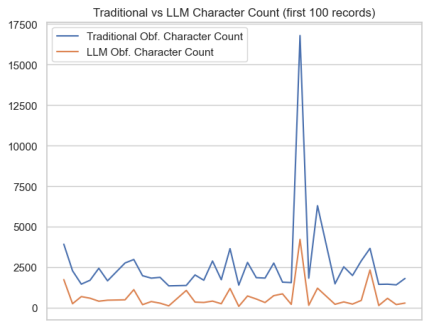


Fig. 6. Output length comparison between javascript-obfuscator output and mistral output

This distinction reflects the underlying design goals of each approach. Traditional obfuscators are explicitly built for security and code protection. Their primary objective is to hinder reverse engineering and intellectual property theft by making code as incomprehensible as possible. LLMs are not designed for security purposes. They apply natural language transformations such as paraphrasing, simplification, and occasional restructuring. These operations inadvertently create a version of the code that is harder to analyze than the original, but still relatively understandable.

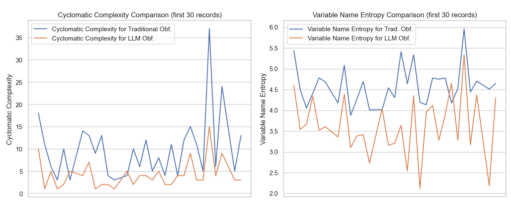


Fig. 7. Cyclomatic complexity and variable entropy comparison

In terms of performance and execution time, the differences for the time needed to obfuscate the code are also notable. Traditional JavaScript obfuscators are highly optimized and extremely fast, while generating an obfuscated version of a JavaScript file using an LLM model takes more time. This difference is largely due to the resource-intensive nature of model inference and the generative step-by-step prediction process that underlies LLM operations.

The results show that using general-purpose LLMs for obfuscation is not ideal for code protection, however, there are reasons why LLMs are being explored for these tasks.

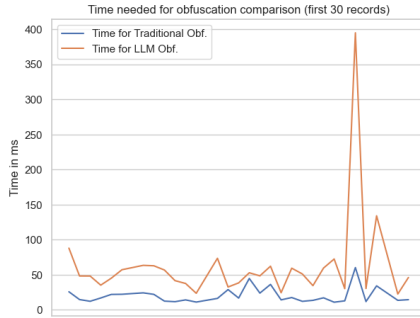


Fig. 8. Time needed for obfuscation comparison

As supported by this research and prior work[27] The current state of LLM code obfuscation shows that LLMs are able to generate functionally equivalent but structurally different code, which makes them a useful experimental starting point for more advanced AI-driven code obfuscation. Unlike traditional tools, machine learning models can be fine-tuned or prompted (in the case of LLMs) to perform obfuscation transformations that combine multiple obfuscation strategies.

F. Experiments with other models

For model selection, two primary paths emerge from the literature:

- **A sequence-to-sequence (seq2seq) neural network:** [6] trains a neural network from scratch and while taking a novel approach of producing an output that is not executable, it provides a starting point for a proof-of-concept. This idea of taking input sentences and converting them into an output with a trained model of weights for character-by-character prediction is the basis of considering this architecture for the proposed obfuscation model. These models have had growing applications in grammatical error correction and text summarization.
- **Fine-tuning a pretrained model:** This approach benefits from the power of existing models and involves fine-tuning a pretrained model specifically for the obfuscation task. [13] provides a blueprint for this by suggesting a reinforcement learning based feedback loop. In this framework, the agent receives rewards for generating code that passes tests (in case of obfuscation, generating functional correct code). Pre-trained BERTs have been widely applied to natural language processing (NLP) tasks and derivative models such as CodeBERT and GraphCodeBERT have been applied to the field of programming language processing (PLP) tasks. [25]

G. RNNs vs. Transformers

[6] uses recurrent neural networks to build a code obfuscator, although the output is not executable code. RNNs are unable to handle long-range dependencies. As the sequence

length increases, the gradients either shrink to zero or grow exponentially during backpropagation.[2] This makes it difficult for the model to capture and learn dependencies from parts of the code, as code obfuscation requires understanding the global context of the program, which RNNs struggle to maintain. Differently from transformer-based models that process tokens in a sequence simultaneously, RNNs process information sequentially, one token at a time, making RNNs perform slowly and inefficiently during training for large codebases. Transformers can process code tokens parallely and allow long-range dependencies and global context through self-attention, important for preserving program semantics[28]. RNNs treat code as a linear sequence of tokens without understanding its hierarchical structure while code obfuscation requires a semantic understanding of the code's abstract syntax tree (AST), making RNNs produce syntactically incorrect and broken code. Empirical evidence from pre-trained models like CodeBERT[7], CodeT5[29] demonstrates state-of-the-art performance in tasks such as code translation, summarization and refinement, all of which involve semantic-preserving code transformations similarly to code obfuscation tasks.

H. Fine-tuning transformer-based models

To carry out the following experiments, we adopted a two-stage fine-tuning strategy that integrates supervised learning and reinforcement learning. In the supervised learning stage, the dataset is labelled into input-output mappings, where each input consists of the original JavaScript code snippet and the corresponding output was its obfuscated version. This provides the model with examples of desired transformations, allowing it to learn the structural and syntactic patterns usually associated with code obfuscation. The reinforcement learning stage introduces a reward-driven optimization process to align the model's outputs with task-specific requirements. For this we design a reward function that aims to encourage the generation of obfuscated code that is both syntactically valid but also semantically equivalent to the original code. The model is positively rewarded when the generated code is executable and preserves the original functionality, which deviations result in negative feedback.

*1) **Fine-tuning CodeT5**:* This experiment attempts to fine-tune a CodeT5 model (Salesforce/codet5-base) in order to perform obfuscation. CodeT5 is a language model based on the T5 architecture and is specifically pre-trained on a vast corpus of code, making it effective for code-related tasks.[29]

Curriculum Learning: Curriculum learning is a set of learning strategies that mimics the way humans learn, starting with easy concepts and then progressing to more complex ones.[32] There are two main requirements for curriculum learning: a predefined metric which can compute the difficulty of the input examples and a curriculum schedule, the rate at which we can augment the training set with more complex samples.[26] In this dataset, we decided to calculate the complexity score (calculating the depth of the AST tree) and separate the dataset into three levels: easy, medium and hard. Based on the difficulty score, we start training with the easier samples for the first iterations and then proceed to the hardest samples in the later phases of training. Implementing curriculum learning introduces a risk of the model remembering the hardest samples better and failing to provide outputs for the easiest samples.

Tokenization: For tokenization, the experiment uses the RobertaTokenizerFast, which is a fast sub-word tokenizer based on the RoBERTa architecture, designed for converting natural language or code into numerical tokens (and vice versa) that the model can process. This tokenizer makes the process easier by handling important pre-processing steps such as tokenization, padding and truncation. .

Training: All levels are partitioned into training and validation sets using a 90/10 split. Some of the arguments of the training process are the number of epochs (10), choosing epochs as a validation strategy in order to choose the model with the lowest validation loss. Early stopping is enabled with a patience of 2 epochs, in order to prevent overfitting.

Performance evaluation: After the model is trained, we try to assess it on a set of unseen data and try to make it generate predictions. The time taken for each prediction is measured and stored along with the input and the predicted output in a file in order to calculate other metrics later. Omitting out the expected output was a choice made due to the importance of the code maintaining functionality and being executable, and the possibility of different obfuscated forms of the same code.

Results: While CodeT5 is optimized for code-to-code transformation tasks, limited token generation capacity was observed during the experiment, resulting in truncated outputs for longer scripts. [23] aims to address these limitations that are a known challenge in dealing with tokenizers for code models. These models are often trained in specific and fixed sequence lengths. When the original script is longer than the maximum number of tokens, the model will stop generating new tokens resulting in short and truncated output. [9] also mentions that even when properly prompted and fine-tuned, these models struggle with generating coherent code in complex scenarios. An open-ended task such as obfuscation, that requires maintaining semantic integrity while altering the readability of the code, may push the model beyond its capabilities, making it unable to generate valid long sequences.

An attempt to scale up to a larger version of this model was made (Salesforce/codeT5-large), however this experiment was met with significant computational resource limitation leading to frequent out-of-memory errors and output collisions.

*2) **Fine-tuning BigBird-Pegasus**:* BERT models use a full attention mechanism which is computationally expensive and limits them to short sequences, therefore we decided to try out fine-tuning BigBird, a transformer with a block sparse attention mechanism that reduces complexity from quadratic to linear[30]. BigBird is a transformer model that allows us to process sequences up to 4,069 tokens efficiently. Pegasus is a transformer model pre-trained for abstractive summarization. Its pre-training objective, called Gap Sentence Generation (GSG), involves masking and reconstructing entire sentences in order to generate a summary, which teaches the model to

focus on important information and synthesize it into an output.[31] The model selected, BigBird-Pegasus combines these two technologies. For this model experiment, techniques such as Low-Rank Adaptation (LoRA) and gradient accumulation were used. The dataset is split into a training set and a test set, with a standard 90/10 ratio. The input and target code snippets are tokenized, padding and truncation are applied in order to ensure uniform length for batch processing.

LoRA: is a parameter-efficient fine-tuning (PEFT) method that adapts a pre-trained model to new tasks without retraining all of its parameters. It freezes the pre-trained weights and injects smaller and trainable matrices into the model's layers[11]. This allows LoRA to reduce the number of trainable parameters, and allows us to fine-tune a large model such as BigBird-Pegasus-Large.

Tokenization: For tokenization, the experiment uses the AutoTokenizer, which handles potential padding tokens to ensure proper data formatting.

Training: Gradient accumulation is used to increase the batch size without requiring more GPU memory by accumulating gradients over multiple smaller batches before updating weights. Epochs are used as an evaluation strategy to monitor performance and detect potential overfitting.

Evaluation: Inference is performed on the test set to generate predictions. Beam search is used with an early stopping mechanism to try and generate a good output sequence.

Results: The primary reason why the model fails to generate valid output is the architectural mismatch from the pre-training objective of the model. BigBird-Pegasus is a model specifically pre-trained for summarization tasks. The model produces a shorter, abstract representation of the input so when fine-tuned to an obfuscation task, it tries to perform a function it was not designed for. What is noticeable in the produced outputs is the repetition of certain strings, which happens because the model defaults to a low-risk output since it cannot correctly map the dependencies between the input and output. Repetitive output is a well known issue that has been addressed with hyperparameters such as repetition_penalty.

I. Experimental environment

We use the Keras platform for conducting the experiments, which is a deep learning library implemented in Python. All code samples were run utilizing NVIDIA A100 GPU runtime in Google Colab. The experimental environment parameters are shown in the table below.

Category	Parameter
CPU	2 virtual CPUs
RAM	13GB
VRAM	40GB GDDR5
CUDA Cores	6,912
Tensor Cores	432
OS	Ubuntu Linux

TABLE I
EXPERIMENTAL ENVIRONMENT PARAMETERS

IV. RESULTS

From the experimental evaluation, it is noticeable that traditional obfuscation tools still outperform in terms of quality and efficiency. While LLMs are able to obfuscate code, in 16.4% of the cases, the LLM-generated code fails to pass functional tests. These results show that while LLMs can produce code-like outputs, maintaining functional correctness remains a key challenge. Similarly to previous research[27], we notice a drop in the cyclomatic complexity of the code as well as the output length in the case of LLM-generated transformations, that show us an output that is harder to analyze than the original code, but still more understandable than the code transformed using a specialized obfuscator. Performance analysis further reinforces this contrast. Traditional JavaScript obfuscators are lightweight, highly optimized, and capable of transforming scripts in near real-time. By comparison, LLM-based obfuscation is slower and more resource-intensive, as it relies on step-by-step generative inference. This computational overhead makes general-purpose LLMs less practical for large-scale obfuscation tasks. Experiments with transformer-based models reveal important limitations. Despite being optimized for code-to-code transformations, CodeT5 typically operates with a maximum sequence length of tokens, which often leads to restricted token generation capacity, therefore demonstrating truncation and loss of structural information in larger code segments. This is a known limitation for transformer-based code models that terminate generation once the maximum token threshold is reached. Even when fine-tuned, CodeT5 struggled to maintain semantic coherence and validity for complex obfuscation scenarios and attempts to scale up to larger variants were limited by computational resource bottlenecks. Similarly, experiments with BigBird-Pegasus produced underwhelming results. The model, originally engineered for handling long-document inputs, when fine-tuned for obfuscation shows repetitive outputs, incomplete transformations and frequent errors. Such issues are consistent with known challenges of generative models, defaulting to low-risk repetitive strings when they cannot reliably map input-output dependencies. Our evaluation shows that they are not optimized for code-specific tasks such as obfuscation due to their initial pre-training objectives. The outputs are frequently insubstantial or structurally invalid, producing little to no meaningful transformation. This aligns with observations that general-purpose long-context models often lack the requisite pretraining or structure-awareness for reliable code manipulation. Overall, the results demonstrate that while LLMs provide a novel and flexible experimental approach to obfuscation, traditional tools remain superior in terms of obfuscation quality, execution reliability, and computational efficiency. LLMs show promise as a foundation for future AI-driven obfuscation research, but their current limitations, particularly sequence length constraints, architectural mismatches, and functional instability, pose significant challenges for practical adoption.

V. Discussion

This research aims to contribute both theoretical and practical advancements to the field of software security, specifically code obfuscation. First, it provides a comparative analysis of traditional obfuscation tools and AI-based approaches. By comparing these techniques against a common set of evaluation metrics, this work aims to offer new insights into the effectiveness and limitations of each obfuscation strategy. The results show that using general-purpose LLMs for obfuscation is not ideal for code protection, however, previous research shows why LLMs are being explored for these tasks. The current state of LLM code obfuscation shows that LLMs are able to generate functionally equivalent but structurally different code, which makes them a useful experimental starting point for more advanced AI-driven code obfuscation. The findings of this study provide an evaluation of the current state of machine learning oriented code obfuscation, setting them against the established performance of traditional, rule-based obfuscation tools. The results show a significant gap in performance as well as readability and obscurity. In this section, we will interpret these findings supported by the existing literature.

A. Readability vs. Obscurity

Similar to previous work[27], we notice the tendency of LLMs to reduce metrics such as cyclomatic complexity, making code less complex than traditional obfuscators that aim to increase it. This is a counter-intuitive behavior that comes as a consequence of being trained in a vast corpora of clean, human-written code that is optimized for simplicity and maintainability. Therefore, when prompted to obfuscate code, LLMs default to their most learned pattern, simplifying the code. Traditional obfuscators excel in their specialized domain but lack adaptability to new strategies or programming languages without significant manual re-engineering. LLMs, on the other hand, demonstrate adaptability and potential for generalization across languages and domains, yet they currently fall short in delivering the reliability required in security-focused applications. Bridging this gap may require integrating reinforcement learning with execution-based feedback, as has been suggested in previous literature[4].

B. Unacceptable risk of functional errors and hallucinations

Beyond just efficiency, the results reveal an important concern: the unreliability of LLM-generated code. Unlike traditional obfuscators that guarantee functional equivalence through reliable transformations, LLMs introduce an unacceptable level of risk. The phenomenon of "hallucinations", where models generate syntactically plausible but nonsensical code, further expands this risk. These errors not only introduce bugs but also introduce the risk of creating new vulnerabilities, making LLMs impractical for professional application in code obfuscation tasks. Although LLMs can match abstract reasoning patterns, they fall short of true logical reasoning. Small changes in input tokens can drastically alter outputs, creating a strong token bias and suggesting that these models are highly sensitive and fragile[18].The models have been shown to heavily depend on variable and function names to infer the code's purpose, making them particularly vulnerable to misleading identifiers. The models struggle with transformations that alter the AST and with layered techniques, which are foundational components of traditional obfuscation tools. This lack of deep, logical understanding makes them prone to errors that might be difficult to detect and fix. This reinforces the challenge noted in other studies where LLMs, despite their generative strength, may lack a full understanding of program semantics and execution environments[1].

C. The need for domain-specific architectures

The case studies on CodeT5 and BigBird Pegasus provide evidence that the models' failures are not random but a consequence of their architectural design and training objectives. Although CodeT5, when fine-tuned on input-output obfuscation pairs, produced what seemed like reasonable transformations, it struggled to generate long sequences. This observation is in line with the documented challenges[29] of sequence-to-sequence models when handling long code inputs, where token truncation or vanishing gradients limit performance. CodeT5's token generation limitations make it unsuitable for code obfuscation tasks on large corpora while BigBird Pegasus's sparse attention mechanism is misaligned with the transformations required for obfuscation. BigBird Pegasus produced non-functional outputs, showing the importance of task-specific architectures and training techniques in code processing tasks. This suggests the need for developing new domain-specific architectures and training methodologies suited for the unique demands of code obfuscation.

VI. Threats to validity

In this section, we discuss the potential threats to the validity of this study and how we decided to treat and control these threats.

Internal validity. Internal validity refers to the extent to which the observed effects can be attributed to the experimental design and the manipulated variables rather than extraneous factors. The main threat is related to the choice of hyperparameters when fine-tuning models. To address this threat, we adopted configurations consistent with prior research and repeated training the models multiple times to mitigate any bias related to randomness. The dataset of original-obfuscated JavaScript pairs might not fully represent the diversity of real-world codebases that results in bias towards specific obfuscation styles. Nevertheless, the dataset employed is a large corpus specifically created for code-processing tasks, with its authors documenting efforts to reduce duplication and ensure richness in terms of variety. We obfuscated the code snippets using different tools such as both traditional and LLM tools and using different obfuscation strategies and techniques. Errors in tokenization or reward function definition during reinforcement learning may have influenced the model performance. For tokenization, rather than working only on token level obfuscation, we attempt to also employ Abstract

Syntax Tree representations to ensure preservation of code structure. The outputs of the model were printed after the supervised learning stage and reinforcement learning stage to check how the model was performing and to ensure the reward function was not giving false results.

External Validity. External validity relates to the generalizability of the study's findings beyond the specific context tested. The experiments are focused on JavaScript obfuscation so the results may not generalize to other programming languages or domains. The study uses a finite dataset and limited hardware so large-scale training or more computational resources might change the comparative outcomes.

Construct Validity. Construct validity assesses whether the study measures what it intends to measure. While we chose metrics that have been used to document obfuscation quality in previous studies, they may not fully capture the quality of obfuscation, which is a multifaceted construct. We ran JavaScript code in a virtual environment, but the evaluation of functional correctness might have missed edge cases, especially for long or complex scripts. JavaScript-Obfuscator was designed specifically for obfuscation whereas the models used were not so comparing them directly may introduce a construct bias.

VII. Limitations and Future Work

This research addresses some of the limitations of current models in code obfuscation tasks, but as mentioned, differences in training corpora, training techniques and hyperparameter selection might yield new results. Further training these models or developing new models with training objectives that explicitly prioritize obscurity and complexity, perhaps through techniques such as adversarial training or reinforcement learning, while still preserving functionality might offer new insights. Furthermore, exploring hybrid approaches that combine the deterministic nature of traditional rule-based tools with the generative capabilities of specialized machine learning models could provide new obfuscation techniques. Building models that are able to generate obfuscated code using one obfuscation technique on the abstract syntax tree of the code and combining the power of these models might be an approach that allows models to learn better. One of the limitations of this study is its focus on specific models and programming languages. Since the main focus of the research was not traditional obfuscation nor LLM obfuscation, different tools than the ones used might result in different results. In the current study, these were used as comparison benchmarks, but future research could expand this analysis to include a wider range of LLMs and rule-based tools to provide a more comprehensive understanding of the landscape. However, it is notable that the consistent and fundamental nature of the errors and architectural mismatches observed in this study, as well as supported by existing literature, suggests that these findings are broadly applicable to the current generation of general-purpose LLMs. The development of adaptive obfuscation schemes that can generalize effectively across different programming languages remains a significant challenge. A successful obfuscation tool that leverages the generative capabilities of artificial intelligence needs to move beyond simple transformations and perhaps adopt a hybrid model architecture that combines token and AST representations, using different learning techniques for training the model as well as adopting a multi-faceted evaluation framework that does not only focus on functional correctness and code complexity but also resilience to state-of-the-art deobfuscation tools. By approaching the problem with a clear understanding of the technical challenges as well as the ethical responsibilities involved, future work can contribute to the development of more resilient software protection, while also acknowledging the potential misuse. As mentioned previously, while code obfuscation is a legitimate technique for protecting intellectual property and proprietary algorithms it also raises important ethical concerns. The advancement of code obfuscation techniques that are resilient towards deobfuscation tools might encourage the development of more sophisticated polymorphic and metamorphic malware, posing a significant threat to traditional anti-virus systems. This raises profound ethical questions about the responsibility of researchers and developers in this field. In the case of code obfuscation, this means commitment to transparency about the employed methodology, willingness to acknowledge the potential of misuse and a focus on developing countermeasures. The goal is not to enable threat actors but to equip defenders.

Acknowledgements

I would like to express my deepest gratitude to the CyberMACS program for providing me with the opportunity, resources, and environment to pursue this research. I am especially thankful to all the people who work for the program. Their dedication, guidance, and continuous support have not only shaped the direction of this thesis but have also enriched my academic and personal growth. Without their efforts, encouragement, and expertise, the completion of this work would not have been possible. I would also like to extend my appreciation to my supervisors and mentors for their invaluable advice, patience, and encouragement throughout the course of this work. Their insights and expertise have guided me at every stage of the research process. I wish to thank my family and friends for their understanding, compassion, and support. Their encouragement has been my greatest strength and has sustained me throughout this journey.

References

[1] Wasi Uddin Ahmad et al. "Unified Pre-training for Program Understanding and Generation". In: *CoRR* abs/2103.06333 (2021). arXiv: 2103.06333. URL: https://arxiv.org/abs/2103.06333.

[2] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[3] Christian Catalano, Giorgia Specchia, and Nicolò G. Totaro. “Enhancing Code Obfuscation Techniques: Exploring the Impact of Artificial Intelligence on Malware Detection”. In: *Product-Focused Software Process Improvement*. Ed. by Regine Kadgien et al. Cham: Springer Nature Switzerland, 2024, pp. 80–88. ISBN: 978-3-031-49269-3.

[4] Mark Chen et al. “Evaluating Large Language Models Trained on Code”. In: *CoRR* abs/2107.03374 (2021). arXiv: 2107.03374. URL: https://arxiv.org/abs/2107.03374.

[5] C.S. Collberg and C. Thomborson. “Watermarking, tamper-proofing, and obfuscation - tools for software protection”. In: *IEEE Transactions on Software Engineering* 28.8 (2002), pp. 735–746. DOI: 10.1109/TSE.2002.1027797.

[6] Siddhartha Datta. *DeepObfusCode: Source Code Obfuscation Through Sequence-to-Sequence Networks*. 2021. arXiv: 1909.01837 [cs.CR]. URL: https://arxiv.org/abs/1909.01837.

[7] Zhangyin Feng et al. “CodeBERT: A Pre-Trained Model for Programming and Natural Languages”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 1536–1547. DOI: 10.18653/v1/2020.findings-emnlp.139. URL: https://aclanthology.org/2020.findings-emnlp.139/.

[8] Igor Golovko et al. “Obfuscation technologies of high-level source code using artificial intelligence”. In: *ICyberPhyS*. 2024. URL: https://api.semanticscholar.org/CorpusID:271822171.

[9] Linyuan Gong, Mostafa Elhoushi, and Alvin Cheung. *AST-T5: Structure-Aware Pretraining for Code Generation and Understanding*. 2024. arXiv: 2401.03003 [cs.SE]. URL: https://arxiv.org/abs/2401.03003.

[10] Abdul Mueed Hafiz, Mahmoud Hassaballah, and Adel Binbusayyis. “Formula-Driven Supervised Learning in Computer Vision: A Literature Survey”. In: *Applied Sciences* 13.2 (2023). ISSN: 2076-3417. DOI: 10.3390/app13020723. URL: https://www.mdpi.com/2076-3417/13/2/723.

[11] Edward J. Hu et al. “LoRA: Low-Rank Adaptation of Large Language Models”. In: *CoRR* abs/2106.09685 (2021). arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

[12] Hamel Husain et al. “CodeSearchNet Challenge: Evaluating the State of Semantic Code Search”. In: *CoRR* abs/1909.09436 (2019). arXiv: 1909.09436. URL: http://arxiv.org/abs/1909.09436.

[13] Prithwish Jana et al. “CoTran: An LLM-Based Code Translator Using Reinforcement Learning with Feedback from Compiler and Symbolic Execution”. In: *ECAI 2024*. IOS Press, Oct. 2024. ISBN: 9781643685489. DOI: 10.3233/faia240968. URL: http://dx.doi.org/10.3233/FAIA240968.

[14] Christian Janiesch, Patrick Zschech, and Kai Heinrich. “Machine learning and deep learning”. In: *Electronic Markets* 31.3 (Apr. 2021), pp. 685–695. ISSN: 1422-8890. DOI: 10.1007/s12525-021-00475-2. URL: http://dx.doi.org/10.1007/s12525-021-00475-2.

[15] Shan Jiang et al. *CASCADE: LLM-Powered JavaScript Deobfuscator at Google*. 2025. arXiv: 2507.17691 [cs.SE]. URL: https://arxiv.org/abs/2507.17691.

[16] Yalan Lin et al. *CodeCipher: Learning to Obfuscate Source Code Against LLMs*. 2024. arXiv: 2410.05797 [cs.CL]. URL: https://arxiv.org/abs/2410.05797.

[17] T.J. McCabe. “A Complexity Measure”. In: *IEEE Transactions on Software Engineering* SE-2.4 (1976), pp. 308–320. DOI: 10.1109/TSE.1976.233837.

[18] Iman Mirzadeh et al. *GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models*. 2024. arXiv: 2410.05229 [cs.LG]. URL: https://arxiv.org/abs/2410.05229.

[19] Seyedreza Mohseni et al. *Can LLMs Obfuscate Code? A Systematic Analysis of Large Language Models into Assembly Code Obfuscation*. 2025. arXiv: 2412.16135 [cs.CR]. URL: https://arxiv.org/abs/2412.16135.

[20] Vadym Mukhin et al. “Obfuscation Code Technics Based on Neural Networks Mechanism”. In: *2020 IEEE 2nd International Conference on System Analysis Intelligent Computing (SAIC)*. 2020, pp. 1–6. DOI: 10.1109/SAIC51296.2020.9239247.

[21] The Research Scientist Pod. *Levenshtein Distance: A Comprehensive Guide to String Edit Distance - The Research Scientist Pod*. 2025. URL: https://researchdatapod.com/levenshtein-distance/.

[22] Tomer Raitsis et al. “Code Obfuscation: A Comprehensive Approach to Detection, Classification, and Ethical Challenges”. American English. In: *Algorithms* 18.2 (Feb. 2025). Publisher Copyright: © 2025 by the authors. ISSN: 1999-4893. DOI: 10.3390/a18020054.

[23] Mootez Saad et al. *On the Effect of Token Merging on Pre-trained Models for Code*. July 2025. DOI: 10.48550/arXiv.2507.14423.

[24] Robin M. Schmidt. *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*. 2019. arXiv: 1912.05911 [cs.LG]. URL: https://arxiv.org/abs/1912.05911.

[25] Leo Song and Steven H.H. Ding. “Milo: Attacking Deep Pre-trained Model for Programming Languages Tasks with Anti-analysis Code Obfuscation”. In: *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. 2023, pp. 586–594. DOI: 10.1109/COMPSAC57700.2023.00084.

[26] Petru Soviany. “Curriculum Learning with Diversity for Supervised Computer Vision Tasks”. In: *CoRR* abs/2009.10625 (2020). arXiv: 2009.10625. URL: https://arxiv.org/abs/2009.10625.

[27] Lorenzo De Tomasi et al. *Simplicity by Obfuscation: Evaluating LLM-Driven Code Transformation with Se-*

mantic Elasticity. 2025. arXiv: 2504.14024 [cs.SE]. URL: https://arxiv.org/abs/2504.14024.

[28] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[29] Yue Wang et al. "CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 8696–8708. DOI: 10.18653/v1/2021.emnlp-main.685. URL: https://aclanthology.org/2021.emnlp-main.685/.

[30] Manzil Zaheer et al. "Big Bird: Transformers for Longer Sequences". In: *CoRR* abs/2007.14062 (2020). arXiv: 2007.14062. URL: https://arxiv.org/abs/2007.14062.

[31] Jingqing Zhang et al. "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization". In: *CoRR* abs/1912.08777 (2019). arXiv: 1912.08777. URL: http://arxiv.org/abs/1912.08777.

[32] Lei Zhou et al. "Self-Guided Curriculum Learning for Neural Machine Translation". In: *CoRR* abs/2105.04475 (2021). arXiv: 2105.04475. URL: https://arxiv.org/abs/2105.04475.

#	Title	Focus	Method	Contributions
1	DeepObfusCode: Source Code Obfuscation Through Seq2Seq Networks	Neural network–based source code obfuscation	Seq2Seq RNN encoder–decoder to transform code into non-readable form, paired with a de-obfuscator model	Demonstrated ML obfuscation with reversible transformations, introducing stealth and execution cost metrics
2	CoTran: An LLM-based Code Translator using RL with Compiler & Symbolic Execution Feedback	Code translation with correctness guarantees	LLM fine-tuned with RL, compiler, and symbolic execution feedback	Provides reinforcement learning setup to preserve semantics after heavy code transformations
3	CODECIPHER: Learning to Obfuscate Source Code Against LLMs	AI-based obfuscation to resist LLM code understanding	Transformer model trained adversarially against LLMs	Introduces LLM-vs-LLM obfuscation resistance evaluation
4	Can LLMs Obfuscate Code? A Systematic Analysis...	Evaluating LLM ability for assembly code obfuscation	Prompt engineering + LLM evaluation	Assesses effectiveness, weaknesses, and consistency of LLM-generated obfuscations
5	Simplicity by Obfuscation: Evaluating LLM-Driven Code Transformation...	Measuring semantic elasticity under obfuscation	LLM transformation pipeline with correctness verification	Highlights semantic preservation under extreme transformations
6	Designing a Code Obfuscation Scheme for Software Protection	Traditional obfuscation scheme	Static transformations: control-flow flattening, identifier renaming	Classic techniques and evaluation metrics for software protection
7	Code Obfuscation: A Comprehensive Approach...	Survey of detection/classification of obfuscation	Compilation of detection methods, ethical implications	Framework for identifying obfuscated code, relevant for evaluating AI outputs
8	Obfuscation Code Technics Based on Neural Networks Mechanism	Neural network–based obfuscation	Early NN model to transform source code	Precursor to DeepObfusCode, proof-of-concept for AI obfuscation

TABLE II
SUMMARY OF THE MOST RELEVANT PAPERS IN THE LITERATURE REVIEW

Method	Obfuscation Quality	Functional Correctness	Performance/sequence handling	Additional notes
Javascript-Obfuscator	High	100%	Moderate	Well-established tool
LLM (Mistral)	Moderate	Variable (fails occassionally)	Moderate	16.4% fail functional tests
CodeT5	Low	Poor on long code	Prone to truncation	Limited length
BigBird-Pegasus	Low	Invalid	Poorly aligned context	Not suited for code obfuscation

TABLE III
SUMMARY TABLE OF EXPERIMENT EVALUATION RESULTS

AI-Driven Cyber Threat Hunting Assistant: NL-to-Query Translation

Hamroz Gavharov
SRH Heidelberg University of Applied Sciences
Kadir Has University
Berlin, Germany
hamroz.gavharov@gmail.com

Prof. Dr. Reiner Creutzburg
Head of Computer Science
SRH Heidelberg University of Applied Sciences
Berlin, Germany
reiner.creutzburg@srh.de

Kendrick Bollens
School of Technology and Architecture
SRH Heidelberg University of Applied Sciences
Berlin, Germany
kendrick.bollens@srh.de

Rahim Dehkharghani
Assistant Professor
Kadir Has University
Istanbul, Turkey
rahim.dehkharghani@khas.edu.tr

***Abstract*—Security operations centers (SOCs) rely on large-scale log analytics to investigate alerts and proactively hunt for threats. Translating natural-language (NL) investigative intents into safe and effective Elasticsearch (ES) DSL is error-prone and time-consuming; naive NL→DSL generation can yield unsafe, unbounded, or semantically off-target queries. We present , a security-first, constrained pipeline combining schema-aware prompting, JSON-Schema gating, a rule-based validator (field/type discipline, hard time-window requirement, and cost control), and a security layer that detects adversarial prompts with calibrated abstention. The system is provider-agnostic (local and cloud LLMs) and evaluated on a 12-scenario bank across a standard logs index, CIC-IDS2017, a schema-drift variant, and differentially private (DP) indices. Under identical guardrails, the enhanced constrained method achieves macro-F1 0.91 (vs. 0.84 schema-grounded few-shot; 0.72 rules; 0.58 zero-shot). First-pass validator success is 78 %, rising to ≥95 % after two critique-guided retries with 3 % abstentions. Robustness under schema drift shows Δ F1 of −0.03 with 88 % one-retry recovery for unknown-field errors. Adversarial testing yields a malicious block-rate of 97.2 % at 3.1 % FPR. Privacy–utility curves degrade monotonically with stronger DP ($\varepsilon \in \{2.0, 1.0, 0.5\}$), e.g., macro-F1 0.90, 0.87, 0.82; numeric-only noise outperforms numeric+timestamp jitter at the same ε. We release an IEEE-style summary aligned to the full thesis, emphasizing safety-by-construction, reproducibility, and open evaluation assets.**

***Index Terms*—Threat hunting, Elasticsearch, NLIDB, LLM, validation, security, differential privacy, schema drift.**

I. Introduction

Security analysts routinely translate plain-language intents into ES DSL to triage incidents, investigate alerts, and hunt unknown threats. This process is non-trivial under operational constraints: queries must respect time windows, field types, and cost budgets, and they must be specific enough to avoid unselective scans. Although recent NL→query systems demonstrate impressive fluency, they rarely commit to strong guardrails that are essential in SOC settings where unsafe or costly queries can impact production systems.

This paper introduces , a security-first NL→DSL framework designed for SOC workflows. Our core premise is *safety-by-construction*: rather than treating safety as an afterthought, we integrate guardrails—schema-aware prompting, JSON-Schema gating, rule-based validation, and adversarial filtering—into the generation loop. The approach is model-agnostic and supports both local and hosted LLMs, enabling organizations with privacy or data-sovereignty requirements to deploy on-prem.

Research Questions. We study: RQ1—Can constrained generation improve execution accuracy while preserving safety under SOC guardrails? RQ2—How robust is the system to adversarial prompts and schema drift? RQ3—What privacy–utility trade-offs arise when logs are differentially privatized?

Contributions. (C1) , a constrained pipeline coupling index-aware prompting with JSON-Schema gating and a rule-based validator that enforces field whitelists, type correctness, time windows, and cost caps; (C2) a multi-layer safety envelope with an adversarial-prompt filter and calibrated abstention; (C3) a multi-dataset evaluation protocol and open artifacts; (C4) empirical studies on robustness and privacy (drift and DP).

II. Related Work

a) NL Interfaces to Databases (NLIDB): Classic surveys and systems in NLIDB focus largely on relational SQL with semantic parsing and schema grounding [1], [2]. Early work such as CHILL learned parsers from annotated data [3], while interactive systems like NaLIR emphasized user-in-the-loop disambiguation [4]. Large cross-domain benchmarks (e.g., Spider) drove rapid advances in text-to-SQL [5], including sequence models (Seq2SQL, SQLNet) [6], [7], execution-guided decoding [8], relation-aware schema encoders (RAT-SQL) [9], grammar-augmented pretraining (GraPPa) [10], and constrained decoding (PICARD) [11]. However, Elasticsearch DSL differs from SQL in operators, time semantics, and cost

behavior, making direct transfer of text-to-SQL techniques non-trivial in SOC settings.

b) LLMs for Program Synthesis and Tool Use: Foundation models enable few-shot program and query synthesis [12], with prompting strategies like chain-of-thought improving reasoning quality [13]. Code-focused evaluations and systems highlight both capability and brittleness [14], [15]. Tool-use paradigms such as ReAct and self-instrumentation approaches like Toolformer illustrate how models can plan and call tools under supervision [16], [17]. Instruction tuning and safety-aligned training further shape model behavior [18], [19]. Our work complements these by embedding strict, domain-specific guardrails (schema-aware prompting, JSON-Schema gating, and a validator) directly into the generation loop.

c) Safety and Robustness: Prompt injection and adversarial prompt families can subvert tool-augmented LLM systems [20]–[22]. Commercial SIEM assistants advertise NL-to-query capabilities (Splunk SPL, Chronicle Gemini, Rapid7 LEQL) but provide limited detail on formal guardrails for schema/type correctness, temporal bounds, or cost control [23]–[25]. Our setting treats the model as an untrusted proposer, enforcing mandatory validation and security filtering prior to execution; this aligns with broader guidance on operational safety for hosted LLM services [26].

d) Privacy in Log Analytics: Differential privacy provides formal protection for analytics [27], with noise calibrated to global sensitivity [28]. Systems research demonstrates practical DP mechanisms for databases and telemetry (e.g., PINQ and RAPPOR) [29], [30]. At population scale, production deployments (e.g., the Census TopDown algorithm) illustrate engineering trade-offs between privacy budgets and utility [31]. We quantify how DP perturbations impact bounded-time threat-hunting queries in Elasticsearch and report the resulting privacy–utility curves.

III. System Design

A. Architecture

Fig. 1 depicts the pipeline: (1) NL prompt enters a router; (2) constrained generator receives index profiles (field types, canonical examples, allowed filters/aggregations); (3) JSON-Schema gating rejects malformed outputs; (4) the validator enforces safety rules; (5) the security layer screens for adversarial or ambiguous inputs; (6) a critique loop retries up to two times; (7) approved queries execute with pre-flight sanity checks.

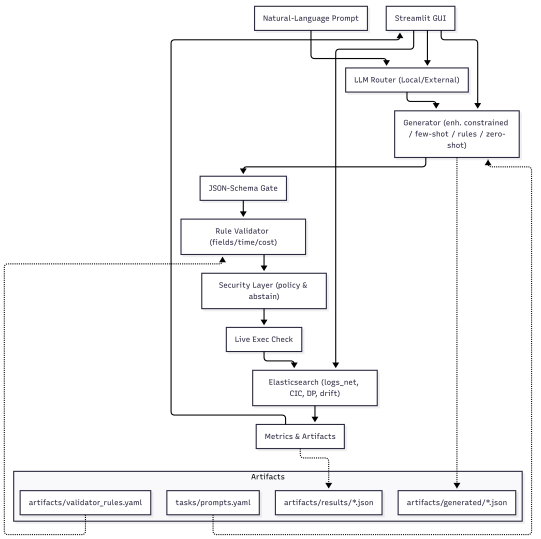
B. Index Profiles and Prompting

For each index we build a compact profile: top fields, types, value sketches, and canonical examples. Prompt templates bind the profile and enforce day-bounded windows unless a stricter bound is specified.

C. Validator and Policy

The validator implements deterministic checks:

- **Field whitelist/type discipline:** filters and aggregations must target known fields with compatible types.



Architecture. Router → Constrained Generator (+schema context) → JSON-Schema Gate → Rule-based Validator (types, fields, time window, cost) → Security Filter & Abstain → Critique/Retry → Execute.

Fig. 1. High-level architecture.

- **Temporal bounds:** enforced hard window (default ≤ 30 days) with day-level granularity.
- **Cost control:** pre-flight selectivity estimate; reject if predicted hits exceed a configured cap.

D. Security Layer

A lightweight classifier flags adversarial patterns (e.g., jailbreaks, role-confusion, code execution requests, or attempts to bypass time windows). If flagged, the system seeks clarification or abstains.

IV. Experimental Setup

A. Datasets and Scenario Bank

We evaluate on: (i) `logs_net` (enterprise-like network telemetry), (ii) CIC-IDS2017 mapped to ES, (iii) a schema-drift variant introducing semantic renames (e.g., `bytes_out`→`bytes_sent`, `label`→`classification`), and (iv) DP-perturbed versions at $\varepsilon \in \{2.0, 1.0, 0.5\}$. The 12-scenario bank covers common SOC tasks (lateral movement, beaconing, data exfiltration, privilege escalation).

B. Methods

We compare: constrained (), schema-grounded few-shot (no constrained decoding), rules-based, and zero-shot. All methods receive identical index profiles; only enforces JSON-Schema gating + validator control.

Algorithm 1 Validator and Retry Controller

```
function VALIDATE(dsl, profile, policy)
    if not JSONSCHEMAVALID(dsl) then return Reject(schema_violation)
    if not FIELDSKNOWN(dsl, profile) then return Reject(unknown_field)
    if not TYPESMATCH(dsl, profile) then return Reject(type_mismatch)
    if not HASTIMEWINDOW(dsl, policy) then return Reject(missing_time_window)
    if COSTTOOHIGH(dsl, policy) then return Reject(cost_violation)
    return Accept
end function
function CRITIQUELOOP(prompt, profile, policy, max_tries=2)
    for i = 0 to max_tries do
        dsl ← CONSTRAINEDGENERATE(prompt, profile)
        outcome ← VALIDATE(dsl, profile, policy)
        if outcome = Accept then return dsl
            prompt ← CRITIQUE(prompt, outcome)  ▷ explain failure and fix
        end if
    end for
    return Abstain
end function
```

TABLE I
MACRO PERFORMANCE ON `LOGS_NET`.

Method	F1	Jaccard	Precision	Recall
Constrained (ours)	0.91	0.87	0.92	0.90
Schema-grounded (few-shot)	0.84	0.79	0.85	0.83
Rules baseline	0.72	0.67	0.74	0.70
Zero-shot	0.58	0.52	0.60	0.56

C. Metrics and Statistics

Execution overlap (Jaccard), macro-Precision/Recall/F1, structural (AST-F1), validator outcomes (first-pass, retries, abstentions; failure taxonomy), latency (Q2 [Q1–Q3]), and adversarial block/FPR. We report 95% CIs and use paired non-parametric tests across scenarios.

D. Providers and Runtime

We route to GPT-class, Claude-class, Gemini-class, and a local Ollama model under identical prompts and guardrails. Latency includes prompt formatting, generation, gating, validation, and (for accepted queries) a cheap pre-flight execution check.

V. RESULTS

A. Main Accuracy on `logs_net`

Table I shows macro outcomes. The constrained method attains F1 0.91 (J 0.87, P 0.92, R 0.90), a clear lift over all baselines under the same safety envelope.

TABLE II
VALIDATOR FAILURE TAXONOMY ACROSS REJECTED GENERATIONS.

Category	Share (%)
`unknown_field`	34
`missing_time_window`	27
`schema_violation`	18
`type_mismatch`	11
`cost_violation`	6
`unselective_aggregation`	4

TABLE III
PROVIDER LATENCY AND FIRST-PASS SUCCESS (CONSTRAINED GUARDRAILS).

Provider	Latency Q2 [Q1–Q3] (s)	First-pass (%)
GPT-class	1.6 [1.3–2.0]	86
Claude-class	1.9 [1.5–2.4]	84
Gemini-class	1.5 [1.2–1.9]	83
Local (Ollama)	3.1 [2.6–3.8]	72

B. Validator Outcomes and Failure Taxonomy

First-pass success is 78 % for , reaching 95 % after two critique-guided retries with 3 % abstention. Table II reports failure categories across rejects.

C. Provider Performance

Table III summarizes latency and first-pass rates under the constrained pipeline. Notably, local inference is slower but competitive after retries.

D. Robustness to Schema Drift

Under drift, macro-F1 drops from 0.91 to 0.88 ($\Delta = -0.03$). First-pass rejects frequently cite `unknown_field`, and 88 % of those cases recover after one retry with targeted critique.

E. Adversarial Safety

On a red-team prompt suite, the security layer blocks 97.2 % of malicious prompts at 3.1 % FPR on benign prompts. End-to-end allowed-and-safe rates after validation are 93 % to 95 % across providers. Failures typically involve indirect attempts to remove time bounds; these are subsequently rejected by validator policy.

F. Privacy–Utility on DP Indices

Macro-F1 degrades monotonically with stronger privacy: non-DP 0.91; $\varepsilon = 2.0$ 0.90; $\varepsilon = 1.0$ 0.87; $\varepsilon = 0.5$ 0.82. Table IV separates numeric-only Laplace noise from numeric+timestamp jitter.

VI. COMPONENT ANALYSIS AND SENSITIVITY

We vary few-shot count, disable gating layers, and relax time-window policy.

a) Few-shot Count (0/2/8): Macro-F1 improves from 0.88 → 0.90 → 0.91; first-pass validity from 73% → 80% → 84% with diminishing returns at 8 shots.

b) Remove JSON-Schema Gating: Schema-grounded only reduces first-pass from 78% to 62%, raises abstentions from 3% to 9%, and lowers F1 from 0.91 to 0.84.

TABLE IV
PRIVACY–UTILITY ON DP INDICES (MACRO-F1).

Setting	ε=2.0	ε=1.0	ε=0.5
Numeric-only	0.90	0.89	0.85
Numeric+timestamp	0.90	0.87	0.82

c) Controller Without Live Schema: Using a controller with no live index context yields F1 0.86, dominated by `unknown_field` errors.

d) Relax Time Windows: Relaxing hard windows increases cost violations from 0.6% to 8.1% without improving F1 materially, supporting the *safety-by-construction* stance.

VII. Reproducibility and Artifacts

We release pinned environment manifests, index mappings, the 12-scenario task bank, and evaluation seeds. Artifacts include: (i) `mappings.json` (index schemas), (ii) `prompts.yaml` (templates and profiles), (iii) `validator_rules.yaml` (policy), (iv) `seeds.txt` (random seeds), and (v) `run.sh` (orchestration). Execution logs store per-try outcomes and the failure taxonomy label for rejected generations.

VIII. Threats to Validity

Internal validity: cache warming and shard layout can alter latency; we report medians with IQR and warm caches before timed runs. **External validity:** datasets are representative but not exhaustive; complex aggregations and cross-index joins are not covered. **Construct validity:** structural metrics reduce false negatives relative to execution-only metrics, but analyst-equivalent queries may differ in clause structure.

IX. Limitations and Ethics

We do not handle deeply nested aggregations or cross-index joins; free-text relevance is limited to basic terms/filters. The red-team suite, while diverse, cannot exhaust all adversarial families. Ethical posture: the system abstains on ambiguous or unsafe inputs, enforces bounded time windows, and restricts unselective scans by default.

X. Conclusion and Future Work

demonstrates that safety-by-construction NL→DSL can deliver high accuracy, calibrated safety, and privacy-aware behavior under realistic SOC constraints. Future work includes richer aggregation patterns, cross-index joins, broader datasets, user studies on workflow fit, and formal privacy guarantees.

References

[1] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, "Natural language interfaces to databases—an introduction," *Natural Language Engineering*, vol. 1, no. 1, pp. 29–81, 1995.

[2] Y. Li and D. Rafiei, "Natural language data management and interfaces: Recent development and open challenges," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1733–1744, 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3035918.3054783

[3] J. M. Zelle and R. J. Mooney, "Learning to parse database queries using inductive logic programming," in *Proceedings of AAAI*, 1996, pp. 1050–1055. [Online]. Available: https://www.cs.utexas.edu/~ml/papers/chill-aaai-96.pdf

[4] F. Li and H. V. Jagadish, "Nalir: An interactive natural language interface for querying relational databases," in *Proceedings of SIGMOD (Demo)*, 2014, pp. 709–712.

[5] T. Yu, R. Zhang, M. Yasunaga, and et al., "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," in *Proceedings of EMNLP*, 2018, pp. 3911–3921.

[6] V. Zhong, C. Xiong, and R. Socher, "Seq2sql: Generating structured queries from natural language using reinforcement learning," *arXiv preprint*, 2017.

[7] X. Xu, C. Liu, and D. Song, "Sqlnet: Generating structured queries from natural language without reinforcement learning," *arXiv preprint*, 2017.

[8] C. Wang, K. Tatwawadi, M. Brockschmidt, and et al., "Robust text-to-sql generation with execution-guided decoding," *arXiv preprint*, 2018.

[9] B. Wang, R. Shin, X. Liu, O. Polozov, M. Richardson, and Y. Su, "Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers," in *Proceedings of ACL*, 2020.

[10] T. Yu, C.-S. Wu, X. V. Lin, and et al., "Grappa: Grammar-augmented pre-training for language-to-sql," *arXiv preprint*, 2021.

[11] T. Scholak, N. Schucher, and D. Bahdanau, "Picard: Parsing incrementally for constrained auto-regressive decoding from language models," *arXiv preprint*, 2021.

[12] T. B. Brown, B. Mann, N. Ryder, and et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[13] J. Wei, X. Wang, D. Schuurmans, and et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[14] M. Chen, J. Tworek, H. Jun, and et al., "Evaluating large language models trained on code," *arXiv preprint*, 2021.

[15] Y. Li, D. Choi, J. Chung, and et al., "Competition-level code generation with alphacode," *Science*, vol. 378, no. 6624, pp. 1092–1097, 2022.

[16] S. Yao, J. Zhao, D. Yu, and et al., "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[17] T. Schick, J. Dwivedi-Yu, R. Dessì, and et al., "Toolformer: Language models can teach themselves to use tools," *arXiv preprint*, 2023.

[18] L. Ouyang, J. Wu, X. Jiang, and et al., "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[19] Y. Bai, S. Kadavath, S. Kundu, and et al., "Constitutional ai: Harmlessness from ai feedback," *arXiv preprint*, 2022.

[20] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising llms via prompt injection," *arXiv preprint*, 2023.

[21] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint*, 2023.

[22] K. Zhu, J. Wang, J. Zhou, and et al., "Promptrobust: Towards evaluating the robustness of large language models with adversarial prompts," *arXiv preprint*, 2024.

[23] Splunk Inc., "Spl ai assistant: Natural language to spl," https://www.splunk.com/, 2023.

[24] Google Cloud, "Gemini in chronicle security operations," https://cloud.google.com/, 2024.

[25] Rapid7, "Insightidr ai assistant: Natural language to leql," https://www.rapid7.com/, 2024.

[26] Microsoft, "Azure openai service: Content management and safety system (whitepaper)," https://learn.microsoft.com/azure/ai-services/openai/, 2023.

[27] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[28] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference (TCC)*, 2006, pp. 265–284.

[29] F. D. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in *Proceedings of SIGMOD*, 2009, pp. 19–30.

[30] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of CCS*, 2014, pp. 1054–1067.

[31] J. M. Abowd, R. Ashmead, R. Cumings-Menon, S. Garfinkel, M. Heineck, C. Heiss, W. Sexton, and et al., "The 2020 census disclosure avoidance system topdown algorithm," *Journal of Privacy and Confidentiality*, vol. 12, no. 1, 2022.

AI-driven Risk assessment in GDPR compliance: Real Time NLP and Machine Learning-based Gap Analysis of Data Protecting activities

Izuchukwu Patrick Udechukwu
SRH Heidelberg University of Applied Sciences
Berlin, Germany
pi.udechukwu.max@gmail.com

Knut Haufe
SRH Heidelberg University of Applied Sciences
Berlin, Germany
Knut.Haufe@de.ey.com

Reiner Creutzburg
SRH Heidelberg University of Applied Sciences
Berlin, Germany
Reiner.Creutzburg@gmail.com

E. Fatih Yetkin
Kadir Has University
Istanbul, Turkey
fatih.yetkin@khas.edu.tr

Adeopatoye Remilekun Jacobs
SRH Heidelberg University of Applied Sciences
Berlin, Germany
adeopatoye@gmail.com

***Abstract*— The General Data Protection Regulation (GDPR) necessitates efficient, scalable compliance tools. Traditional audits are time-consuming and error-prone, hindering the management of dynamic data processing activities. This research proposes an AI-powered real-time GDPR compliance auditing framework, using Natural Language Processing (NLP) to analyse legal texts and data. The system detects compliance gaps and provides actionable insights for compliance officers. By integrating NLP and Machine Learning, the framework ensures transparency, helping organisations streamline compliance and mitigate risks in a constantly evolving regulatory landscape.**

Keywords*— *GDPR Compliance, AI-driven Risk Assessment, Natural Language Processing (NLP), Compliance Automation

I. INTRODUCTION

The General Data Protection Regulation (GDPR) has introduced stringent data protection requirements for organizations operating within the European Union and beyond, making compliance a legal necessity and a key factor in maintaining organizational trust and reputation. Among the GDPR's most critical provisions is the obligation for organizations to document and continuously monitor their data processing activities. However, traditional compliance auditing methods are manual and reactive and struggle to keep pace with data operations' increasing volume and complexity. The inability to track real-time compliance, coupled with the rapid evolution of business processes and regulations, exposes organizations to potential fines, legal risks, and reputational damage. As the regulatory landscape becomes more dynamic, there is an urgent need for intelligent, automated tools that can provide real-time insights into compliance status and detect violations proactively. This research addresses these shortcomings by proposing an AI-powered framework for continuous, real-time GDPR compliance auditing. By integrating Natural Language Processing (NLP), Machine Learning, the system can interpret and analyze legal texts, analyse organisational data processing activities, and detect compliance gaps in real-time.

By providing actionable insights and enhancing transparency, this framework enables organisations to proactively manage compliance, reducing the risk of fines, reputational damage, and operational inefficiencies. Through this approach, the research aims to bridge the gap between legal requirements and real-time data operations, providing an innovative solution to the challenges organisations face in ensuring GDPR compliance.

II. LITERATURE REVIEW

A. *AI-Driven GDPR Compliance Diagnostics*

The integration of artificial intelligence (AI) into GDPR compliance has gained significant attention due to its potential to automate and enhance compliance audits. Traditional manual processes are insufficient in managing the complexity and volume of data operations under GDPR. Recent advancements in Natural Language Processing (NLP), machine learning (ML), and knowledge graphs provide innovative approaches to overcome these challenges. For example, studies by [1] demonstrated the effectiveness of a BERT-based NLP model for assessing the alignment of privacy policies with GDPR Article 13, achieving high precision and recall in policy analysis. This success highlights the potential of AI-driven tools for automating legal document interpretation and gap identification, which is central to the proposed research. However, gaps such as the applicability of these models in multilingual contexts and the integration of policy text interpretation with operational data remain unresolved.

B. *Explainable AI in Risk Assessment*

In the context of GDPR compliance, AI models are critical for assessing and automating compliance checks across data processing activities. However, many AI models' "black-box" nature—especially in legal and regulatory settings—poses a significant challenge. Explainable AI (XAI) ensures that AI-

driven decisions are transparent, interpretable, and auditable, making them suitable for legal and regulatory scrutiny. [2] emphasised the importance of transparency in decision-making processes, noting that AI models in compliance contexts must provide explainable outputs to be valid in legal audits. This aligns with the research objectives, which propose a framework that not only automates the detection of compliance gaps but also ensures that the reasoning behind these detections is comprehensible and verifiable by human stakeholders. Additionally, studies on the hybridisation of ML and deep learning [3] show that combining different models can enhance risk diagnostics, offering a pathway for more accurate, transparent, and adaptive compliance systems.

C. *Cognitive Graph Architectures for Real-Time Governance*

Recent contributions in cognitive graph architectures highlight their potential for real-time governance and compliance management. [4] introduced a cognitive graph-based framework that automates data lineage tracking across distributed environments, improving traceability and reducing manual compliance efforts. Cognitive graphs enhance the interpretability and scalability of compliance systems by offering dynamic updates and semantic enrichment. These frameworks also address key GDPR requirements, such as demonstrating lawful data processing and ensuring transparency. However, challenges in integration with legacy systems and computational costs remain.

D. *Hybrid Models and Adaptive Compliance Systems*

Hybrid AI models, which combine machine learning (ML), deep learning (DL), and NLP, are gaining traction for improving GDPR compliance. Studies by [3] and [5] demonstrated the effectiveness of such hybrid systems in risk assessment and anomaly detection. These models offer adaptability, real-time responsiveness, and interpretability, making them suitable for complex compliance environments like GDPR. Hybrid models can fuse structured data, such as financial indicators, with unstructured data, such as legal texts, to provide comprehensive compliance assessments. Furthermore, the integration of federated learning [5] in these models supports privacy-preserving compliance, which aligns with GDPR's emphasis on data minimization. Despite their promise, challenges in model interpretability and multi-jurisdictional compliance remain.

E. *Knowledge Graphs and NLP in Legal Interpretation*

The application of Knowledge Graphs (KGs) and Natural Language Processing (NLP) in legal interpretation, particularly for GDPR compliance, has seen significant development. Knowledge graphs allow for semantic reasoning across complex legal texts, while NLP techniques enable the extraction and analysis of compliance-relevant clauses. [6] developed a system called DERECHA, which employs NLP to verify Data Processing Agreements (DPAs) compliance against GDPR requirements. The system demonstrated significant improvements in precision and recall, outperforming conventional NLP models in legal text analysis. [7] highlighted the trend towards domain-specific NLP models for legal tasks, advocating for tailored pretraining and model explainability in compliance contexts. These findings underscore the importance of combining domain-specific NLP with KGs to enhance the accuracy and explainability of compliance systems. Despite these advancements, challenges such as ensuring multilingual support and aligning NLP outputs with legal norms remain key hurdles in developing fully automated GDPR compliance solutions.

III. PROBLEM STATEMENT

The General Data Protection Regulation (GDPR) mandates that organisations continuously document and assess their data processing activities to ensure compliance with legal standards. However, traditional auditing methods are manual, labour-intensive, and prone to errors, making it difficult for organisations to keep up with the fast-evolving data regulations. These approaches, relying on periodic documentation reviews and static policy assessments, often miss the dynamic nature of data flows, leaving gaps in compliance that can expose organisations to regulatory penalties and reputational risks. There is a critical need for automated, scalable, and real-time compliance tools to address these challenges. Existing rule-based systems are limited in interpreting legal texts and assessing ongoing data processing activities effectively, failing to provide a holistic view of GDPR compliance. This research proposes an AI-driven framework utilising Natural Language Processing (NLP), and Machine Learning to develop AI-driven system that can interpret legal documents, analyse data processing activities, detect compliance gaps in real time, and deliver actionable insights, while overcoming challenges related to legal language interpretation, scalability, and explainability for non-technical stakeholders.

IV. METHODOLOGY

The research methodology for this study is designed to develop and validate an AI-powered framework for real-time GDPR compliance auditing. The framework integrates Natural Language Processing (NLP), and Machine Learning to automate the detection of compliance gaps in data processing activities, linking regulatory requirements directly to operational data.

A. *Data Collection*

To build a comprehensive framework for GDPR compliance, various data sources will be collected and utilized. The full GDPR corpus and relevant legal commentary and amendments will serve as the foundational legal text. Privacy policies from GDPR-regulated organisations were collected and used to evaluate how well internal practices align with the regulatory framework. The privacy policy serves as a comprehensive statement of an organisation's data protection practices, covering data collection methods, purposes for processing, data sharing practices, and data retention periods. These policies are totally anonymised to protect sensitive information, focusing only on the clauses and terms related to data collection, processing purposes, and retention. After collection, the privacy policies were processed using NLP techniques to identify key components related to GDPR compliance.

B. *NLP-Based Text Preprocessing and Legal Interpretation*

The NLP component will focus on automating the extraction and analysis of compliance-relevant information from legal texts and organisational documentation:

- **Text Preprocessing**: The GDPR text and organisational documents will be preprocessed using standard NLP techniques, including tokenisation, part-of-speech tagging, and named entity recognition (NER).
- **Legal Clause Extraction**: The BERT-based transformer models will be fine-tuned on GDPR-specific corpora to identify key regulatory clauses, obligations, and compliance requirements. This will allow the system to map organisational documentation to GDPR clauses automatically.
- **Semantic Classification**: NLP models will classify the identified clauses into compliance categories, such as data subject rights, consent management, and data minimisation.

C. *AI Model Development for Gap Detection*

To identify compliance gaps in real time, machine learning (ML) models will be trained using labelled datasets of compliant and non-compliant data:

- **Model Training**: To detect compliance gaps, this research will utilise a Random Forest model. The Random Forest algorithm is selected for its intrinsic interpretability, as it provides a natural mechanism for determining feature importance, which will support the explainability claim of this research. This interpretability is crucial for ensuring compliance officers understand the factors driving the model's decision-making process.
- **Anomaly Detection**: In this research, anomalies refer to unexpected or rare instances where data processing activities deviate from the expected or compliant behaviour, which could indicate a potential GDPR compliance issue. These anomalies could include, for example, unusual patterns in data access or processing activities, unauthorised changes in data flows, or deviations from standard data retention policies. Isolation Forest will be applied to identify these outliers in the data. By detecting such anomalies, the system can flag potential compliance gaps that have not yet been captured through traditional rule-based audits, providing early alerts for preventive actions.
- **Explainable AI (XAI)**: For the Random Forest model, the importance of intrinsic features will be used to explain the decisions made by the model. This will allow us to identify which features (e.g., certain clauses in GDPR or specific data processing activities) contribute most to compliance decisions, enhancing transparency and interpretability. Since Random Forest inherently offers this level of explainability, using techniques like SHAP or LIME would be redundant.

D. *System Validation and Testing*

To validate the effectiveness of the proposed system, the framework will be tested using real-world data from GDPR-regulated organisations:

- **Real-World Data Integration**: For testing and validating the system, real-world data such as Data Protection Impact Assessments (DPIAs) and Records of Processing Activities (ROPAs) will be collected from organisations such as data processors (e.g., cloud service providers), financial institutions, and healthcare providers already subject to GDPR. The system logs from these organisations, detailing data flows and metadata changes, will also be used for real-time monitoring during the testing phase.
- **Accuracy and Precision Testing**: To assess the effectiveness of the AI-powered compliance detection system, accuracy and precision will be measured against traditional manual audits. The manual audits will serve as a baseline for comparison, involving compliance officers reviewing the same set of organisational documentation and data processing activities, as they would in a conventional compliance check. We will measure improvements in efficiency by calculating the time taken by the AI system to detect compliance gaps versus the time taken by human auditors. Additionally, precision and recall metrics will be used to evaluate how well the AI system detects true compliance gaps compared to the manual auditing process. This will show that the AI system offers more consistent and rapid detection of issues.
- **User Feedback**: Compliance officers and technical teams will provide feedback on the system's outputs, including the comprehensibility of the compliance reports and the actionable insights generated. This feedback will inform further refinement of the system's user interface and reporting capabilities.

E. *System Implementation and Dashboard Development*

Upon improvement of the system, a user-friendly dashboard will be developed to provide actionable insights to compliance officers:

- **Real-Time Monitoring**: The dashboard will display real-time compliance statuses, including detected gaps, risks, and recommendations for corrective actions.
- **Visual Analytics**: Data visualisations, such as compliance heatmaps and risk severity graphs, will be used to convey complex compliance information clearly and intuitively.
- **Reporting System**: The system will generate detailed reports, linking compliance gaps to specific GDPR clauses, and offer remediation recommendations. These reports will be designed for both legal and technical audiences.

V. RESULT

This chapter presents a detailed discussion of the findings from the research on the AI-driven GDPR compliance auditing framework. The primary objective of this study was to design and evaluate an automated system that leverages Artificial Intelligence (AI), particularly Natural Language Processing (NLP) and machine learning (ML), to monitor GDPR compliance and identify potential compliance gaps. The research explored how these technologies can streamline

compliance, reduce reliance on manual audits, and enhance organisational efficiency and transparency.

The findings discussed in this chapter provide insights into the system's effectiveness in detecting compliance gaps, automating legal text interpretation, and enabling proactive compliance management. Key outcomes include the system's ability to detect non-compliant practices accurately, the role of NLP in understanding complex legal texts, and the system's capacity for monitoring and reporting. Additionally, the chapter reflects on the practical implications of these findings, the theoretical contributions to the field of legal-tech, and the challenges faced during the implementation of the framework. The insights and interpretations drawn from these findings are intended to provide a comprehensive understanding of the potential impact of AI on GDPR compliance and its broader applications in regulatory automation.

Summary of Key Findings

This section summarises the key findings from the research, which evaluated the effectiveness of the AI-driven GDPR compliance system in automating the detection of compliance gaps, providing monitoring, and improving the accuracy of compliance assessments. The following points highlight the system's significant achievements and contributions.

Success in Detecting Compliance Gaps Using Machine Learning and NLP

A primary goal of the research was to assess the system's ability to detect compliance gaps using machine learning ML and NLP techniques. The findings indicate that the system effectively identified discrepancies between organisational data processing activities and GDPR requirements.

- **Machine Learning:** The system successfully applied supervised learning to detect common compliance issues, such as missing consent or data retention violations, by training models on labelled datasets of compliant and non-compliant activities. To further validate the research, multiple machine learning models were tested for compliance gap detection. The models were evaluated using **accuracy, precision, recall, and F1-score** on a labelled dataset of privacy policies.

Table 1: Model Accuracy Across Machine Learning Approaches

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Logistic Regression	81	79	83	80.5
Decision Tree	83	81	84	82
Random Forest	87	85	91	87.5
Isolation Forest for Anomaly Detection	77	76	80	78.5
Gradient Boosting	85	83	87	84

In the table above, Random Forest achieved the highest overall performance, balancing high accuracy and interpretability. Decision Trees provide clear decision pathways but slightly lower predictive accuracy compared to ensemble models. Additionally, Isolation Forest, used for unsupervised anomaly detection, identified subtle compliance gaps not present in the labelled dataset but had lower overall accuracy.

- **NLP:** Integrating NLP, particularly LegalBERT, allowed the system to automatically parse and interpret complex legal texts, mapping specific GDPR clauses to corresponding data processing activities. This process enabled the system to identify subtle gaps in compliance, such as incomplete privacy policies or vague descriptions of data subject rights, that could otherwise go unnoticed.

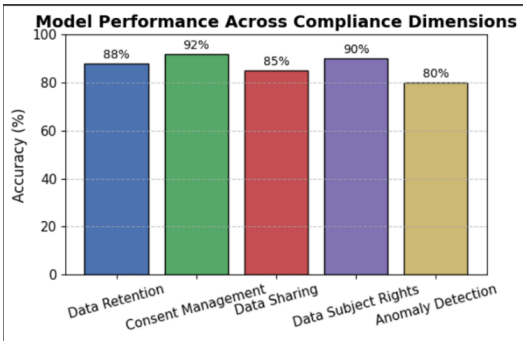


Figure 1: Model performance across Compliance dimension

Table 2: Model Performance Across Compliance Dimensions

Compliance Dimension	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Data Retention	88	85	90	87.5
Consent Management	92	90	95	92.5
Data Sharing	85	82	88	85
Data Subject Rights	90	88	92	90
Anomaly Detection	80	78	82	80

Consent Management and Data Subject Rights achieve highest accuracy, demonstrating the system's effectiveness in critical compliance areas.

The combined analysis of performance dimensions and model accuracy supports the research claims that:

1. The AI-driven framework outperforms traditional compliance audits in monitoring, predictive accuracy, and adaptability.
2. Random Forest is the most suitable model for GDPR compliance due to its balance between high predictive accuracy and interpretability.
3. Incorporating anomaly detection (Isolation Forest) allows the system to capture previously unseen gaps, increasing the robustness of the compliance monitoring.
4. Feature importance and decision pathway insights from Random Forest provide transparent reasoning for flagged non-compliant activities, which is critical for audit accountability.

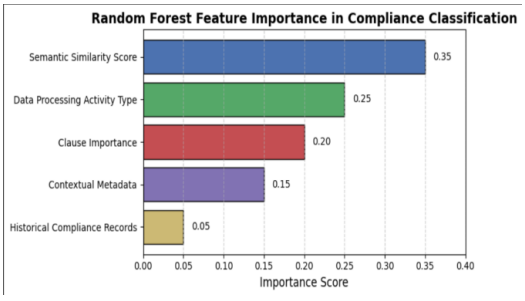


Figure 2: Random Forest Importance in the Compliance Classification

1. **Semantic Similarity Score (0.35)**: Most important feature, showing the system heavily relies on how closely a privacy policy matches GDPR clauses. Semantic similarity is the most influential feature, validating the NLP preprocessing strategy.
2. **Data Processing Activity Type (0.25)**: Shows the type of activity (collection, sharing, retention) strongly influences predictions.
3. **Clause Importance (0.20)** and **Contextual Metadata (0.15)**: Highlights that weighted critical clauses and contextual information matter in classification.
4. **Historical Compliance Records (0.05)**: Lower importance, indicating the model relies more on current document analysis than historical trends.

The Accuracy of the Compliance Gap Detection System

Accuracy was a central focus of this research, and the findings suggest that the AI-driven system demonstrated high accuracy in detecting compliance gaps, thanks to its sophisticated use of NLP and machine learning. The system's effectiveness was evaluated through internal validation, expert reviews, and comparisons with traditional manual audits.

- **Detection Precision:** The system was highly effective in correctly identifying compliance gaps, such as improper handling of sensitive personal

data, inadequate data subject consent mechanisms, and failure to comply with data retention policies. The machine learning models could learn from labelled datasets, improving their accuracy over time by distinguishing between compliant and non-compliant data processing activities.

- **Minimising False Positives:** One of the system's strengths was its ability to minimise false positives (i.e., incorrectly flagging compliant activities as non-compliant). The system improved its precision by continuously learning and refining its predictions based on a growing dataset, providing stakeholders with reliable compliance assessments.

- **Model Interpretability**: In addition to accuracy, the system also ensured interpretability through explainable AI techniques, allowing users to understand why certain activities were flagged as non-compliant. This transparency, combined with high detection accuracy, made the system reliable and user-friendly, enabling compliance officers to trust the system's findings and take appropriate action.

Ethical and Practical Limitations

- **Data privacy and security:** Sensitive data anonymized; risk of re-identification acknowledged.
- **Interpretability and trust:** Random Forest pathways and feature importance improve transparency, but complex models may still require expert review.
- **Generalization:** System optimized for GDPR; adaptation to HIPAA, CCPA requires retraining and rule adjustment.

The system demonstrated exceptional accuracy in detecting compliance gaps across a range of data processing activities, offering a high degree of precision in identifying real violations and minimising errors in compliance assessment. The system successfully detected compliance gaps using machine learning and NLP, providing monitoring capabilities that enable proactive compliance management. Furthermore, the accuracy of the compliance gap detection system was validated through rigorous testing, confirming its reliability and precision in identifying both evident and subtle non-compliance issues. These findings demonstrate the potential of AI to revolutionise GDPR compliance auditing, making it more efficient, accurate, and scalable for organisations of all sizes.

VI. DISCUSSION AND ANALYSIS

The AI-driven GDPR compliance system developed in this research successfully integrated NLP and machine learning techniques to identify compliance gaps in organizational privacy policies. The system demonstrated high predictive accuracy across key compliance dimensions such as Data Retention, Consent Management, and Data Subject Rights, while maintaining interpretability through Random Forest feature importance and decision pathways. Additionally, the framework provided monitoring capabilities, enabling proactive compliance management and actionable reporting for stakeholders

The findings of this study highlight several important insights regarding the automation of GDPR compliance auditing:

1. **Effectiveness of NLP in Compliance Interpretation**:

- NLP models, particularly **LegalBERT embeddings**, successfully extracted compliance-relevant information from GDPR texts and organizational documentation.
- Preprocessing techniques such as tokenization, lemmatization, and semantic vectorization allowed the system to map organizational practices to relevant regulatory clauses.
- Optimization strategies, including domain-specific fine-tuning and semantic similarity threshold adjustments, further improved classification performance.
- **Implication**: Properly optimized NLP can significantly reduce the manual effort required to interpret complex legal texts, supporting the research objective of improving compliance monitoring efficiency.

2. **AI Techniques for Compliance Gap Detection and Risk Assessment**:

- Random Forest effectively classified data processing activities as compliant, or non-compliant.
- Anomaly detection using Isolation Forest identified previously unseen gaps, enhancing risk assessment capabilities.
- Risk scoring methodology translated detected gaps into actionable metrics for prioritizing interventions.

The combination of supervised and unsupervised techniques provides a robust framework for both identifying known compliance gaps and discovering novel risks, aligning with the objective of effective AI-based compliance monitoring.

3. **Interpretability and Stakeholder Communication**:

- Feature importance scores and decision pathways from Random Forest provided transparency in decision-making.
- Automated report generation, coupled with visualizations such as accuracy tables and bar charts, allowed stakeholders to understand compliance risks and recommended actions efficiently.

These tools support actionable insights, helping compliance officers, executives, and regulators make informed decisions, fulfilling the objective of interpretable compliance insights.

4. **Monitoring and Practical Impact**:

- Continuous evaluation of data processing activities allowed for proactive detection of non-compliant actions.
- The system's scalability ensures that even large datasets across multiple departments or regions can be analyzed efficiently.
- Monitoring transforms compliance management from a reactive to a proactive approach, directly supporting the research objective of enhancing GDPR oversight.

METHODOLOGICAL LIMITATIONS

Despite the system's effectiveness, several limitations must be acknowledged:

1. **Dataset Scope**: The study analyzed 20 privacy policies, which may limit generalizability across organizations with vastly different privacy practices.
2. **Regulatory Focus**: The system was optimized specifically for GDPR; applying it to other regulatory frameworks such as HIPAA, CCPA would require model adaptation and retraining.
3. **Data Privacy Concerns**: Handling sensitive personal data, even in anonymized form, poses ethical risks and necessitates strict data governance.
4. **AI Interpretability**: While Random Forest improves transparency, some complexity remains in fully understanding ensemble model decisions.

These limitations indicate that while the findings are promising, caution should be applied when extrapolating results to broader populations or regulatory contexts.

COMPARISON WITH LITERATURE REVIEW

The study's findings generally align with prior research on AI-based compliance automation:

- Consistent with studies demonstrating NLP effectiveness in legal text analysis, the system accurately interpreted GDPR clauses and linked them to organizational practices.
- Aligns with literature on ensemble machine learning, showing Random Forest and Gradient Boosting provide high accuracy and reliable predictions for classification tasks.
- Divergence exists regarding monitoring: previous studies often focus on periodic audits, whereas this study demonstrates continuous compliance assessment, indicating a methodological advancement.

Underlying reasons for differences:

- The combination of supervised and unsupervised learning for both known and novel compliance gaps.
- The use of semantic similarity scoring with domain-specific embeddings, which enhances NLP performance beyond generic models.

TRANSFERABILITY OF RESULTS

The results of this study can be generalized to a broader range of organizations, industries, and regulatory contexts under certain conditions:

1. **Cross-Industry Application**: Organizations with structured privacy policies and formalized data processing practices can implement the system with minimal adaptation.
2. **International Adaptation**: While designed for GDPR, the methodology is adaptable to other data protection frameworks by retraining NLP models and adjusting regulatory clause mappings.
3. **Scalability**: monitoring and AI-driven reporting are applicable to both small enterprises and large multinational organizations with diverse datasets.

Limitations to transferability: Adaptation to industries with unstructured or highly customized data policies may require additional preprocessing and domain-specific model tuning. This study demonstrates that an AI-driven framework integrating optimized NLP, supervised and unsupervised machine learning, and interpretable reporting can effectively identify GDPR compliance gaps, assess associated risks, and deliver actionable insights to stakeholders. While methodological limitations and regulatory scope require careful consideration, the system represents a scalable, interpretable, and practical approach to automated GDPR compliance monitoring, contributing both theoretically and practically to the field of legal-tech compliance.

VII. CONCLUSION

The AI-driven GDPR compliance framework developed in this research represents a significant step forward in automating compliance auditing for organizations. By integrating NLP and ML, the system provides an efficient, and scalable solution for real-time compliance monitoring. Key findings from the study include:

1. ML and NLP for Gap Detection: The system detected both common and subtle compliance gaps by leveraging Random Forest and Isolation Forest ML algorithms, combined with LegalBERT for parsing and interpreting complex legal texts.

2. Proactive Compliance Management: Unlike traditional periodic audits, the system would enable proactive compliance, continuously monitoring and flagging potential violations before they result in regulatory breaches.

3. High Accuracy and Model Interpretability: The system demonstrated high precision and low false positives, making it a reliable tool for GDPR compliance auditing, with explainable AI

techniques ensuring transparency in decision-making.

4. Scalability and Efficiency: The AI-driven framework proved scalable, capable of handling large and diverse datasets, thus meeting the needs of organizations of varying sizes and complexities.

IMPACT ON INDUSTRY AND SOCIETY

The implications of this research extend beyond academic contributions and into the practical realm of data privacy and regulatory compliance. By automating compliance monitoring, the proposed AI-driven framework has the potential to significantly reduce the burden on organizations, both financially and operationally. As we understand, traditional compliance audits are time-consuming and prone to human error, often leading to missed violations and delayed detection. This AI-driven approach minimizes those risks, allowing for continuous monitoring of data processing activities in an organization.

For businesses, the ability to ensure proactive compliance means that they can mitigate regulatory risks, avoid costly fines, and build trust with their customers and regulators. By providing stakeholders with clear, actionable compliance insights, the system fosters a culture of transparency and accountability, aligning with evolving data privacy regulations. Customers benefit from enhanced data protection, knowing their personal information is being processed in compliance with stringent laws like GDPR, which in turn improves customer confidence. The societal impact is equally significant, as the framework contributes to greater privacy protection for individuals. As more organizations implement such AI systems, the overall safeguard of personal data improves, fostering a culture of privacy-conscious organizations and reducing the risk of data breaches.

LESSONS LEARNED

While the AI-driven compliance framework is a major advancement in regulatory automation, the research revealed several important lessons:

1. Data Quality is Key: The success of ML models heavily relies on the quality of the input data. Incomplete or inaccurate privacy policies or regulatory documents can hinder the effectiveness of the system. Continuous efforts are needed to ensure that data remains up-to-date and accurate.

2. Continuous Model Improvement: The AI system's ability to adapt and improve through continuous learning is crucial. As new regulations or updates to GDPR are introduced, the system must be retrained to accommodate these changes, ensuring its long-term effectiveness.

3. Collaboration Between Legal and Technical Teams: Bridging the gap between legal and technical domains is essential. The complexity of GDPR compliance necessitates a collaborative approach between legal experts and IT teams to ensure the framework remains effective across various organizational contexts.

METHODOLOGICAL STRENGTHS AND LIMITATIONS

The methodological strengths of this research lie in the integration of NLP for legal text interpretation and ML for compliance gap detection. These innovations allow for scalability, efficiency, and real-time monitoring, which are significant improvements over traditional compliance methods.

However, the research also encountered several limitations:

1. Regulatory Specificity: While the framework was developed with GDPR in mind, its generalization to other regulatory frameworks, such as HIPAA or CCPA, would require further adaptation. Each regulation has unique clauses and requirements that would necessitate retraining the AI models.

2. Interpretability of Complex Models: Although Random Forest models provide a high degree of interpretability, they still present challenges in fully explaining every decision. More research into explainable AI techniques for legal compliance is necessary to increase trust in these systems.

3. Data Privacy Concerns: Handling sensitive data, even with anonymization techniques, presents inherent privacy risks. Ensuring robust encryption, access controls, and data minimization practices is essential to mitigate these risks.

POTENTIAL FOR FURTHER RESEARCH

This research opens several avenues for future investigation in the field of automated compliance auditing:

1. Extension to Other Regulatory Frameworks: Future studies could focus on adapting the AI framework for other legal systems such as healthcare privacy laws (HIPAA) or financial regulations (PCI-DSS). Customizing the system for these frameworks could provide a more comprehensive solution for multi-regulatory environments.

2. Enhanced NLP Models: The integration of more sophisticated NLP models, such as transformer-based architectures, could improve the system's ability to understand legal context and semantics. Further exploration into contextual language understanding is crucial for improving compliance gap detection.

3. Improving AI Explainability: As AI-driven decision-making becomes more prevalent in regulatory settings, improving the explainability of AI models will be critical. Future research could explore advanced techniques like LIME, SHAP, and

counterfactual explanations to enhance the transparency of compliance gap detection.

4. Real-Time Adaptation to Regulatory Changes: The system could be enhanced to automatically adapt to changing regulations, ensuring continuous compliance without manual intervention. This could be achieved through the use of knowledge graphs or dynamic rule engines that update the system's compliance checks as legal requirements evolve.

It has been shown that AI-driven frameworks can significantly streamline GDPR compliance monitoring, transforming it from a reactive process into a proactive one. The ability to detect compliance gaps using machine learning and NLP not only improves efficiency and accuracy but also ensures organizations can act before non-compliance results in penalties. This research offers organizations a practical, scalable solution that reduces audit costs, mitigates risks, and enhances transparency and trust. As data protection regulations continue to evolve globally, the AI-driven compliance auditing framework provides a powerful tool for ensuring organizations stay ahead in meeting their regulatory obligations.

REFERENCES

[1] L. Zhang, N. Moukafih, H. Alamri, G. Epiphaniou, and C. Maple, "A BERT-based Empirical Study of Privacy Policies' Compliance with GDPR," Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.06778

[2] R. Sultana, "ARTIFICIAL INTELLIGENCE FOR DECISION MAKING IN THE ERA OF BIG DATA EVOLUTION." [Online]. Available: https://orcid.org/0009-0004-6636-7654

[3] H. Zhang and X. Yang, "Design and Implementation of an AI-Driven Hybrid Framework for Risk Assessment," in *2024 7th International Conference on Advanced Algorithms and Control Engineering, ICAACE 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 1454–1461. doi: 10.1109/ICAACE61206.2024.10548147.

[4] M. Raja Pulicharla, "AI-Augmented Data Lineage: A Cognitive Graph-Based Framework for Autonomous Data Traceability in Large Ecosystems," *Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal) Impact*, vol. 8, no. 1, p. 377, 2025, doi: 10.15680/IJMRSET.2025.0801055.

[5] A. Shaikhmuhammad and C. Researcher, "AI-Powered Cybersecurity Compliance: Bridging Regulations and Innovation."

[6] O. A. Cejas, M. I. Azeem, S. Abualhaija, and L. C. Briand, "NLP-Based Automated Compliance Checking of Data Processing Agreements Against GDPR," *IEEE Transactions on Software Engineering*, vol. 49, no. 9, pp. 4282–4303, Sep. 2023, doi: 10.1109/TSE.2023.3288901.

[7] D. M. Katz, D. Hartung, L. Gerlach, A. Jana, and M. J. Bommarito, "Natural Language Processing in the Legal Domain," Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.12039

Anonymous CTI Sharing: A Collaborative Model for Privacy-Preserving Threat Intelligence Exchange

Asem Mousa[1], Petre Lameski[1], Hasan Dağ[2], Ivan Chorbev[1]
[1]Faculty of Computer Science and Engineering, Ss Cyril and Methodius University, Skopje, North Macedonia
[2]Department of Management Information Systems, Kadir Has University, Istanbul, Turkey

***Abstract*—The increasing frequency and sophistication of cyberattacks underscore the need for adequate cybersecurity solutions, particularly by utilizing Cyber Threat Intelligence (CTI). CTI is critical for maintaining a proactive security posture, but its collaborative sharing is often hindered by concerns over data privacy, reputation risk, and the potential exposure of sensitive or proprietary information. To defeat these deficiencies, this research proposes a novel plan for privacy-preserving CTI sharing. Our approach is to include an OPRF-based Private Set Intersection (PSI) scheme as an external sidecar of OpenCTI, which is a popular open-source threat intelligence solution. With such an arrangement, two organizations are able to compute the intersection of their CTI datasets privately and only publish the common IoCs without disclosing the remainder of their CTI data. The architecture is built to interoperate with industry standard Structured Threat Information eXpression (STIX), enabling interoperability. The success and experimentation with this protocol prove its potential to enable secure, cooperative CTI sharing without compromising privacy, while supporting a trusted, intelligence-driven cybersecurity environment.**

***Index Terms*—Cyber Threat Intelligence, Private Set Intersection, Anonymous sharing, Privacy, CTI sharing**

I. Introduction

The number of reported cyberattacks, such as ransomware, phishing, and social engineering, has increased significantly in recent years, drawing the attention of both the public and commercial sectors [3]. Cyber Threat Intelligence (CTI) has emerged as a key player in modern cybersecurity defense mechanisms, allowing organizations to predict, detect, and respond to threats more effectively. Cyber Threat Intelligence (CTI) involves analyzing data to generate actionable insights about existing or emerging cyber threats targeting an organization. By understanding attackers' motives and capabilities, CTI enables organizations to proactively prevent or mitigate cyberattacks, shifting from a reactive to a proactive security approach. This approach improves decision making, strengthens defense mechanisms, and reduces potential financial and reputational damage [10].

The primary goal of CTI is to transform raw data into meaningful intelligence that can inform decision making and improve the security posture of an organization. By analyzing indicators of compromise (IoCs), threat actors' tactics, techniques, and procedures (TTPs), and broader trends in the cyber threat space, CTI empowers organizations to identify vulnerabilities, prioritize remediation process, and respond to incidents with greater speed and precision [5]. For example, CTI can reveal patterns in phishing campaigns, malware distribution, or ransomware attacks, allowing organizations to implement targeted defenses and reduce their risk exposure [25]. This intelligence-driven approach is particularly critical in an era where cyber threats not only are increasing in frequency but are also becoming more sophisticated and difficult to detect.

The value of CTI goes beyond mere threat detection; it also plays a crucial role in strategic planning and risk management. By understanding the motivations and capabilities of threat actors, organizations can better allocate resources, develop robust incident response plans, and strengthen their overall security infrastructure [10]. For example, CTI can help organizations identify which assets are most likely to be targeted, enabling them to implement proper security measures and reduce the likelihood of a successful attack. Furthermore, CTI fosters collaboration within the cybersecurity community, as organizations share threat data and insights to collectively combat cybercrime [25].

Despite its many benefits, the implementation of CTI is not without challenges. The large amount of data generated by modern IT environments can overwhelm security teams, making it difficult to distinguish between relevant threats and false positives. Furthermore, the dynamic nature of the cyber threat landscape requires organizations to continuously update their intelligence-gathering and analysis capabilities to stay ahead of adversaries [14]. To address these challenges, many organizations are turning to automated tools and platforms that leverage artificial intelligence (AI) and machine learning (ML) to streamline the CTI process and improve its accuracy [5].

Cyber Threat Intelligence reports contain many Indicators of Compromise (IoC) data, which might reveal sensitive information about the organization's infrastructure, such as IP addresses, file hashes and network domains. Moreover, it could violate the privacy of users by revealing usernames, emails, and IP addresses [11]. Additionally, companies hesitate to share CTI data because it may expose details about their security posture, which may lead to reputation loss [24].

This research addresses the above limitations by proposing a novel approach for privacy-preserving, CTI sharing built on Oblivious Pseudorandom Function (OPRF)-based Private Set Intersection (PSI). Specifically, it presents an implementation framework that integrates an OPRF-based PSI protocol as an external sidecar to OpenCTI, a popular open-source threat intelligence platform, with data feeds such as AlienVault's OTX. This allows two parties to compare CTI datasets (e.g.,

IoCs) and securely compute their intersection—revealing only the shared elements—without disclosing the rest of their threat data to each other.

The approach aligns with CTI standards such as STIX (Structured Threat Information eXpression) and TAXII (Trusted Automated eXchange of Indicator Information) enabling interoperability while preserving privacy at the data layer. The protocol is particularly suitable for structured data environments like OpenCTI, where observables (domains, IPs, URLs, hashes) can be extracted, normalized, and securely compared across parties.

The research objective is to address the following questions:

- What is the privacy risk of sharing cyber threat intelligence reports?
- How can Secure Multi-Party Computation (SMPC) preserve privacy in CTI sharing between entities?

Answering these questions requires deep investigation of the current literature and proposed solutions, and that will be discussed in the next section, which include literature review discussing different approaches to secure CTI sharing, and finally, a Proof of Concept of our implementation, along with a discussion of the results.

II. Literature Review

This section explores the landscape of Cyber Threat Intelligence sharing, beginning with surveys of the current approaches in sharing CTI data, and extending to various methods of implementing frameworks to share and process CTI feeds. We followed a scoping review methodology that aims to identify and evaluate key sources related to a specific research topic. The current review follows the guidelines set forth by [19] and draws upon previous systematic reviews. The reviewed studies are categorized based on their research area 1. Most of the research we reviewed included surveys and challenges related to CTI sharing, in addition to conducted studies on different approaches to implementing CTI.

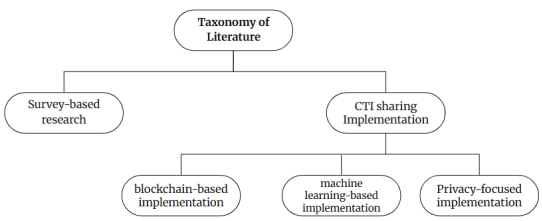


Fig. 1: Taxonomy of Literature

When choosing the papers for our literature review, we ensure that only relevant articles will be included in the scoping review. Each critically assessed study should cover at least one of the following topics:

- CTI challenges
- Safe Threat Intelligence sharing
- Secure Multiparty Computation
- Private Set Intersection

The studies included in this scoping review were collected from various online databases, including PubMed, IEEE Xplore, Science Direct, and Google Scholar. The search keywords employed were ("Cyber Threat Intelligence sharing" OR "Cyber Threat Intelligence" OR "Data sharing privacy"). The articles were extracted in March 2025. The search results yielded 1130 studies using these keywords. Among these, we chose the selected ones in accordance with our criteria, aligning with the primary objective of this review. In summary, the total number of included studies is 12. This comprise research articles from journals, books, and conference papers. The articles were meticulously analyzed to ensure the reliability of the findings. Table I summarizes the literature found in the area of CTI sharing.

III. Implementation

The model aims to achieve anonymity when sharing IoCs between two organizations. This is done by employing OpenMined [18] implementation of Private Set Intersection (PSI) protocol in the context of OpenCTI threat sharing platform [17]. In practice, Organization A resembles a Server that has a larger CTI data - could be obtained from public sources - and Organization B resembles a client that want to know the intersection of both CTI datasets. In OpenMined PSI, only the client knows the common elements between both parties. In this case, Organization B (client) does not reveal raw data, and Organization A (server) only share necessary knowledge, without compromising rest of the dataset.

This model is designed to work with OpenCTI platform, and tailored for STIX (Structured Threat Information eXpression) [16] observable objects, to easily integrate with other CTI platforms such as MISP [23]. The figure 2 below illustrates the workflow of the model.

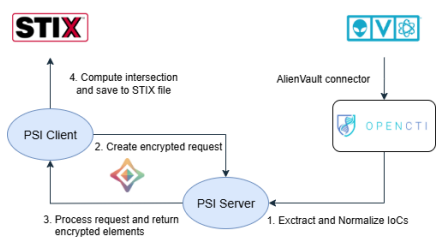


Fig. 2: PSI pipeline workflow

A. Experimental Setup

The test scenario was set to evaluate the integration of a Private Set Intersection (PSI) protocol with an OpenCTI-based Cyber Threat Intelligence (CTI) sharing process workflow. The deployment was a virtualized, containerized architecture to ensure portability, reproducibility, and component isolation.

TABLE I: Summary of Literature Review on Anonymous and General CTI Sharing

Reference	Year	Category	Contribution	Authors
Collective Threshold Multiparty Private Set Intersection Protocols for Cyber Threat Intelligence [12]	2024	Privacy-focused Implementation	Presents a generic composition that extends any threshold multiparty private set intersection (T-MPSI) protocol into a collective T-MPSI protocol for privacy-preserving cyber threat intelligence sharing.	C. Guan, J.S. van Assen and Z. Erkin
A Privacy-Preserving Cyber Threat Intelligence Sharing System [13]	2024	Privacy-focused Implementation	introduces a system to facilitate the secure exchange of cyber observables across trust boundaries without compromising the anonymity of the sharing entities	Philip Huff, Spencer Massengale, Tran Viet Xuan Phuong, Sri Nikhil Gupta Gourisetti
SeCTIS: A framework to Secure CTI Sharing [4]	2024	Blockchain-based Implementation	introduces a novel framework that securely shares Cyber Threat Intelligence using Swarm Learning and Blockchain	Dincy R. Arikkata et al.
Current approaches and future directions for Cyber Threat Intelligence sharing: A survey [2]	2024	Survey-based research	A comprehensive survey on CTI sharing, analyzing architectures, challenges, solutions, and future directions.	Poopak Alaeifar, Shantanu Pal, Zahra Jadidi, Mukhtar Hussain, Ernest Foo
Standards-based Cyber Threat Intelligence sharing using private Blockchains [20]	2023	Blockchain-based Implementation	Develops a decentralized CTI platform via Hyperledger Fabric using STIX/TAXII to enhance security, auditability, and trust.	Kimonas Provatas, Ioannis Tzannetos, Vassilios Vescoukis
Blockchain-Based Cyber Threat Intelligence Sharing Using Proof-of-Quality Consensus [7]	2023	Blockchain-based Implementation	Introduces PoQ consensus mechanism with validator selection based on reputation and quality to enhance trust.	Dimitrios Chatziamanetoglou, Konstantinos Rantos
A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience [21]	2023	Survey-based research	Proposes a CTI framework for detection and risk visualization to support organizational cybersecurity hardening.	Saqib Saeed et al.
A Democratically Anonymous and Trusted Architecture for CTI Sharing using Blockchain [8]	2022	Blockchain-based Implementation	Develops a novel blockchain-based architecture to securely share Cyber Threat Intelligence (CTI) data, addressing challenges of privacy, trust, and accountability	Kealan Dunnett et al.
LUUNU — Blockchain, MISP, Model Cards and Federated Learning Enabled Cyber Threat Intelligence Sharing Platform [6]	2022	Blockchain-based Implementation	Presents "LUUNU", a blockchain, MISP, Model Cards, and Federated Learning enabled platform for secure and private Cyber Threat Intelligence sharing, ensuring anonymity and data provenance	Eranga Bandara, Abdul Rahaman, Xueping Liang
Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing [22]	2022	Machine learning-based Implementation	Proposes botnet DGA detection using statistical features; blends XAI and OSINT for explainable, bias-resistant intelligence.	Hatma Suryotrisongko, Yasuo Musashi, Akio Tsuneda, Kenichi Sugitani
A Theoretical Review on the Importance of Threat Intelligence Sharing & The Challenges Intricated [1]	2021	Survey-based research	Highlights existing platforms and the need for automation and standardization in CTI sharing.	Sandhya Sukhabogia, M. Anushab
An Overview of Cyber Threat Intelligence Platform and Role of AI and Machine Learning [9]	2020	Machine learning-based Implementation	Proposes AI/ML-enhanced CTI framework for improved intelligence extraction.	Abir Dutta, Shri Kant

The PSI server is connected to OpenCTI instance running within Docker containers, as per the official OpenCTI deployment guidelines [17]. The threat intelligence content of the server originated from the AlienVault OTX connector [15], which imports Indicators of Compromise (IoCs) in Structured Threat Information Expression (STIX) format.

PSI server and client are both running on the same Linux Mint VirtualBox virtual machine to create a controlled network environment. PSI client was simulated locally and configured to read a prepared test STIX file containing IoCs imported from AlienVault pulse. The client securely transmitted this set of IoCs to the PSI server, which will be later compared against OpenCTI-stored IoCs without revealing non-intersecting parts, keeping data confidential.

B. PSI pipeline

The communication process between the PSI client and server followed a structured, privacy-preserving pipeline designed to ensure that only intersecting Indicators of Compromise (IoCs) were revealed [18]. The server 3 first accessed IoCs from OpenCTI using GraphQL API, which were first consumed using AlienVault OTX connector. The IoCs were normalized into a consistent format and subsequently encrypted using the private PSI key (s) of the server in preparation for sending it to the client for matching procedures.

```
(psi_env) asem@asem-VirtualBox:~/psi_projects python psi_server.py
2025-08-09 17:39:42,476 [INFO] Health check (platform version)...
2025-08-09 17:39:42,502 [INFO] Listing Identities with filters
2025-08-09 17:39:42,588 [INFO] Listing Indicators with filters
2025-08-09 17:39:46,432 [INFO] Extracted 244 IOCs from OpenCTI for author 'AlienVault'
2025-08-09 17:39:46,433 [INFO] OpenCTI IOCs extracted and normalized: 244
2025-08-09 17:39:46,526 [INFO] Server listening on localhost:65432
2025-08-09 17:40:08,295 [INFO] Client connected: ('127.0.0.1', 39746)
2025-08-09 17:40:08,300 [INFO] Sent PSI response to client
```

Fig. 3: Running PSI server

On the client side 4, a formatted STIX file of IoCs was imported and processed to create a PSI request. The client encrypted its collection of IoCs using its private key (c) and invoked the `create_request` function to create a PSI protocol-compatible request message.

```
(psi_env) asem@asem-VirtualBox:~/psi_projects python psi_client3.py
2025-08-09 17:40:08,281 [INFO] Loaded 33 IOCs from STIX file
2025-08-09 17:40:08,281 [INFO] Client IOCs loaded and normalized: 33
2025-08-09 17:40:08,295 [INFO] Connected to server at localhost:65432
2025-08-09 17:40:08,306 [INFO] PSI Matches found: 9
```

Fig. 4: Running PSI client

The encrypted request was sent to the server, which invoked its `process_request` procedure to perform the PSI computation. The server responded with an encrypted reply that contained the double-encrypted elements, in addition to the encrypted server's collection. The client, being the intended recipient, used `get_intersection` to decrypt and reveal only the IoCs common to both sets. It securely compared the server's encrypted data set to the encrypted client set without revealing non-common elements. This way, each side maintains the confidentiality guarantees of the PSI protocol. Table II lists the components of PSI protocol, each with its functional role in the protocol workflow.

TABLE II: Roles and functions in OpenMined PSI implementation

Component	OpenMined Function	Description
Client	`psi_client`	Has the smaller dataset. Creates and sends a request.
Server	`psi_server`	Has the larger dataset, processes client requests.
Request	`create_request`	Encrypted client dataset sent to server.
Response	`process_request`	Server's encrypted response sent back to client.
Intersection	`get_intersection`	Client computes intersection (common items).

IV. Results & Discussion

After running the PSI pipeline successfully, matched IoCs are saved in a separate file, which can be later imported into another OpenCTI instance on the client side, or shared with trusted partners for further analysis.

Table III shows a summary of the finding metrics. Note that this is a very small sample of IoCs, some datasets may include thousands of records or even more. Despite the small number of intersecting IoCs, it shows the high accuracy of PSI protocol when reducing the false positives rate. This is of course will increase the communication costs.

TABLE III: PSI Matching Summary

Metric	Value
Client IoCs (STIX file)	33
Server IoCs (OpenCTI)	244
Intersecting IoCs	9
Matching Rate	27%

An example of a matched IoC is the STIX object shown in Listing 1, representing the domain `canonicalconnect.com`, which appears in both the client and server datasets.

Listing 1: Example of a STIX Indicator in JSON format

```
{
  "created": "2025-08-08T17:08:32.000Z",
  "description": "",
  "id": "indicator--6973f41d-d337-4856-8137-81
     af9bd1b10e",
  "labels": [],
  "modified": "2025-08-08T17:08:32.000Z",
  "name": "OTX pulse_name=Exposed JDWP
     Exploited in the Wild:
   What Happens When Debug Ports Are Left Open
      ",
  "pattern": "[domain-name:value = '
     canonicalconnect.com']",
  "pattern_type": "stix",
  "pattern_version": "2.1",
  "spec_version": "2.1",
  "type": "indicator",
  "valid_from": "2025-08-08T17:08:32.000Z"
}
```

The same observable also appears on OpenCTI platform (server side):

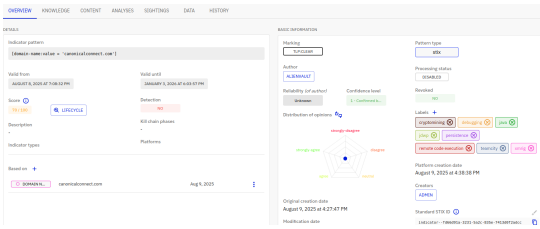


Fig. 5: Matched domain-name

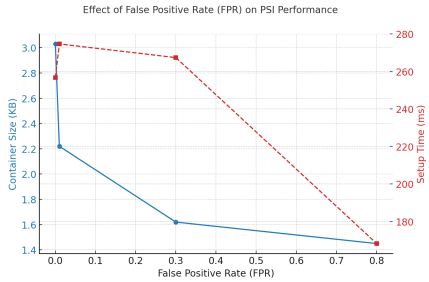


Fig. 6: FPR effect

A. Performance evaluation

When it comes to implementing PSI with Golomb Compressed Sets (GCS), False-positive Rate (FPR) is a key factor that affects the accuracy and communication cost, changing this parameter will significantly reflect on the matching process, especially with a larger dataset size. Here, we run the PSI pipeline again with a slightly larger and fixed sample with 999 IoCs loaded from the server, and 1773 IoCs loaded from the client, along with a variant FPR. Table IV displays the results of each test.

TABLE IV: Effect of False Positive Rate (FPR) on PSI Performance

FPR	Setup Time (ms)	Container Size (KB)	Matches Found
0.0001	256.83	3.03	136
0.01	274.70	2.22	136
0.3	267.43	1.62	136
0.8	168.23	1.45	137

As shown in the figure 6, changing false positive rate (FPR) directly affects the container size: low values of FPR produce larger containers, and higher values of FPR reduce communication overhead. Similarly, the server-side initialization time grows slightly for smaller values of FPR, but this effect is still not considerable for the dataset size used in this experiment. On larger data sets, however, the impact on setup time may become significant. These results highlight the fundamental trade-off between accuracy and communication cost in PSI protocol. It is noteworthy that with the highest tested FPR (0.8), one additional matched IoC was added, which implies a false positive. Although this variation is small in the current experiment, the impact may be noticeable when implementing the protocol on larger datasets.

B. Discussion

The PSI protocol implementation achieved its primary objective of enabling collaborative threat intelligence exchange without exposing entire datasets, utilizing Elliptic-curve Diffie–Hellman (ECDH) encryption and Golomb Compressed Sets for efficient set representation. Relevant IoCs from the client side was found in the server's OpenCTI repository via the AlienVault OTX connector. Using this approach limited full data disclosure and protected sensitive information from unwanted sharing, directly contributed in tackling the trust and confidentiality concerns that often hinder CTI sharing. The results also showed how the False Positive Rate (FPR) influences performance: smaller FPR values increase container size and setup time, while larger values improve efficiency but risk introducing more false positives. Although the effect was minimal in our experiment, these findings highlights the importance of carefully choosing the FPR in larger deployments to get the right balance between accuracy and performance.

V. Conclusion

Cyber Threat Intelligence (CTI) is key to modern cyber defense, but as attacks grow more intelligent and focused, sharing Indicators of Compromise (IoC) remains a challenge. The paper introduced a privacy-preserving mechanism for CTI sharing based on OPRF-based Private Set Intersection (PSI) protocol with OpenCTI platform, in which IoC matching can be done securely and anonymously without disclosure of non-intersecting data. Using open-source technologies and complying with standards like STIX, the solution provides a deployable model for collaborative sharing of CTI. Experimental deployment in OpenCTI showed smooth integration into existing workflows, identifying overlaps even with small datasets while reducing trust barriers and safeguarding sensitive information like Personally Identifiable Information (PII). Scaling to larger datasets and integration with other CTI systems remains a promising direction for standardization and wider adoption.

References

[1] S. S. et al. "A Theoretical Review on the Importance of Threat Intelligence Sharing & The Challenges Intricated". In: *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.3 (Apr. 2021), pp. 3950–3956.

[2] Poopak Alaeifar et al. "Current approaches and future directions for Cyber Threat Intelligence sharing: A survey". In: *Journal of Information Security and Applications* 83 (2024), p. 103786. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2024.103786. URL: https://www.sciencedirect.com/science/article/pii/S2214212624000899.

[3] Fatimah Aldauiji, Omar Batarfi, and Manal Bayousef. "Utilizing Cyber Threat Hunting Techniques to Find Ransomware Attacks: A Survey of the State of the Art". In: *IEEE Access* 10 (2022), pp. 61695–61706. DOI: 10.1109/ACCESS.2022.3181278.

[4] Dincy R. Arikkat et al. "SeCTIS: A framework to Secure CTI Sharing". In: *Future Generation Computer Systems* 164 (2025), p. 107562. ISSN: 0167-739X. URL: https://www.sciencedirect.com/science/article/pii/S0167739X24005260.

[5] Kurt Baker. *Cyber Threat Intelligence Explained*. 2025. URL: https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/.

[6] Eranga Bandara et al. "LUUNU — Blockchain, MISP, Model Cards and Federated Learning Enabled Cyber Threat Intelligence Sharing Platform". In: *2022 Annual Modeling and Simulation Conference (ANNSIM)*. 2022, pp. 235–245.

[7] Dimitrios Chatziamanetoglou and Konstantinos Rantos. "Blockchain-Based Cyber Threat Intelligence Sharing Using Proof-of-Quality Consensus". In: *Security and Communication Networks* 2023.1 (2023), p. 3303122.

[8] Kealan Dunnett et al. "A Democratically Anonymous and Trusted Architecture for CTI Sharing using Blockchain". In: *2022 International Conference on Computer Communications and Networks (ICCCN)*. 2022, pp. 1–7.

[9] Abir Dutta and Shri Kant. "An Overview of Cyber Threat Intelligence Platform and Role of Artificial Intelligence and Machine Learning". In: *Information Systems Security*. Springer Cham, Dec. 2020, pp. 81–86. ISBN: 978-3-030-65609-6.

[10] eccouncil. *What is Threat Intelligence in Cybersecurity?* 2024. URL: https://www.eccouncil.org/cybersecurity-exchange/threat-intelligence/what-is-cyber-threat-intelligence/.

[11] Gina Fisk et al. "Privacy Principles for Sharing Cyber Security Data". In: May 2015, pp. 193–197. DOI: 10.1109/SPW.2015.23.

[12] C. Guan, J.S. van Assen, and Z. Erkin. "Collective Threshold Multiparty Private Set Intersection Protocols for Cyber Threat Intelligence". In: *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*. 2024, pp. 1–6. DOI: 10.1109/WIFS61860.2024.10810671.

[13] Philip Huff et al. "A Privacy-Preserving Cyber Threat Intelligence Sharing System". In: *2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*. 2024, pp. 49–58. DOI: 10.1109/TPS-ISA62245.2024.00016.

[14] kaspersky. *What is threat intelligence? Definition and explanation*. 2025. URL: https://www.kaspersky.com/resource-center/definitions/threat-intelligence (visited on 03/25/2025).

[15] LevelBlue. *AlienVault OTX*. 2025. URL: https://otx.alienvault.com/.

[16] oasis. *STIX*. 2025. URL: https://oasis-open.github.io/cti-documentation/stix/intro.

[17] opencti. *OpenCTI Documentation*. 2025. URL: https://docs.opencti.io/.

[18] OpenMined. *OpenMined PSI*. 2025. URL: https://github.com/OpenMined/PSI.

[19] Danielle Pollock et al. ""How-to": scoping review?" In: *Journal of Clinical Epidemiology* 176 (2024), p. 111572. ISSN: 0895-4356. DOI: https://doi.org/10.1016/j.jclinepi.2024.111572. URL: https://www.sciencedirect.com/science/article/pii/S0895435624003287.

[20] Kimonas Provatas, Ioannis Tzannetos, and Vassilios Vescoukis. "Standards-Based Cyber Threat Intelligence Sharing Using Private Blockchains". In: *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*. 2023, pp. 649–656. DOI: 10.15439/2023F6880.

[21] Saqib Saeed et al. "A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience". In: *Sensors* 23.16 (2023). ISSN: 1424-8220. DOI: 10.3390/s23167273. URL: https://www.mdpi.com/1424-8220/23/16/7273.

[22] Hatma Suryotrisongko et al. "Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing". In: *IEEE Access* 10 (2022), pp. 34613–34624.

[23] Cynthia Wagner et al. "MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform". In: *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*. ACM. 2016, pp. 49–56.

[24] Thomas D. Wagner et al. "Cyber threat intelligence sharing: Survey and research directions". In: *Computers Security* 87 (2019), p. 101589. ISSN: 0167-4048. URL: https://www.sciencedirect.com/science/article/pii/S016740481830467X.

[25] Hemant Warier. *The Crucial Role of Cyber Threat Intelligence in Protecting Your Organization*. 2025. URL: https://www.cloudsek.com/knowledge-base/the-crucial-role-of-cyber-threat-intelligence-in-protecting-your-organization (visited on 02/13/2025).

Assessing Vulnerabilities in IoT Protocols: A Cross-Layer Approach

1st Berfin Ebrar Atabey
Faculty of Computer Science and Eng.
Ss Cyril and Methodius University
Skopje, North Macedonia
eb.atabey@gmail.com

2nd Dr. Sasho Gramatikov
Faculty of Computer Science and Eng.
Ss Cyril and Methodius University
Skopje, North Macedonia
sasho.gramatikov@finki.ukim.mk

3rd Dr. Mehmet Nafiz Aydın
Faculty of Economics and Admin. Sci.
Boğaziçi University
Istanbul, Turkiye
mehmetn.aydin@bogazici.edu.tr

***Abstract*—Security problems have grown as a result of the expanding use of Internet of Things (IoT) devices in industries, smart homes, and healthcare. Many of the IoT protocols, such as MQTT and CoAP, were created with portability and efficiency in mind, however they lack inherent security. Others, such as AMQP and HTTP, include stronger mechanisms but introduce more overhead, which makes them harder to use in constrained environments. Because of these distinctions, IoT systems are vulnerable to various attacks.**

This paper studies the vulnerabilities of four commonly used IoT protocols—MQTT, CoAP, AMQP, and HTTP—through a cross-layer perspective. Controlled experiments were carried out to test how these protocols behave under real attack scenarios at the application, transport, and network layers. Mitigation techniques, including authentication, access control, TLS/DTLS encryption, and rate limiting, were then applied to see how effective they were and what kind of performance cost they introduced.

The findings demonstrate that all of the evaluated protocols have cross-layer vulnerabilities, but they also demonstrate that light mitigations can be successful without always resulting in significant performance loss. The study emphasizes that the deployment context has a significant impact on selecting the "right" protocol and security configuration, and that layered and protocol-aware protection tactics are most effective for IoT systems. Also it shows that choosing the "right" protocol and security setup depends strongly on the deployment context, and that IoT systems benefit most from layered and protocol-aware defense strategies.

***Index Terms*—IoT, protocol vulnerabilities, MQTT, CoAP, AMQP, HTTP, cross-layer security, mitigation, performance**

I. Introduction

IoT technologies have seen rapid adoption in areas including healthcare, smart homes, and industrial automation [1]. Protocols like MQTT and CoAP have become popular in IoT systems due to their efficiency and suitability for low-power, resource-limited devices [2]. While AMQP and HTTP are widely supported and provide advanced features such as authentication and reliable delivery, their higher complexity makes them less suitable for constrained IoT environments [3]. However, many of these protocols were not designed with strong security in mind [4]. As a result, IoT systems face vulnerabilities ranging from data interception and unauthorized access to denial-of-service and replay attacks [4]. Weaknesses at the application, transport, and network layers can be chained together, creating cross-layer attack paths that traditional single-layer defenses fail to address [5]. Prior surveys (e.g., Naik [2], Tournier et al. [3]) have compared IoT protocols and highlighted their security limitations, but these works focus mainly on conceptual analysis rather than experimental validation. In addition, many studies propose security mechanisms for IoT protocols, although their performance implications in constrained environments are not systematically evaluated [6], [7].

This paper addresses these gaps through an experimental, cross-layer analysis of four widely used IoT communication protocols: MQTT, CoAP, AMQP, and HTTP. We identify and validate real attack scenarios across multiple layers, implement protocol-specific mitigation techniques, and evaluate their effectiveness under low- and high-load traffic conditions. The contributions of this paper consists of: a systematic cross-layer vulnerability mapping of MQTT, CoAP, AMQP, and HTTP, an experimental evaluation of protocol-specific mitigations, including authentication, access control, TLS/DTLS encryption, and rate limiting, a performance analysis quantifying the trade-offs between security and efficiency in constrained IoT environments. By combining vulnerability validation with performance measurements, this work provides practical insights for deploying protocol-aware, layered defenses in IoT systems.

II. Related Work

The security and application domain of IoT protocols has been widely studied in the past decade. Among the most referenced surveys in IoT research is Al-Fuqaha et al. [1], which reviews enabling technologies and protocols but does not address security in detail. Naik [2] compared messaging protocols including MQTT, CoAP, AMQP, and HTTP, identifying trade-offs in efficiency and functionality, but stopping short of analyzing their vulnerabilities in depth. Tournier et al. [3] extended this perspective with a survey of IoT protocols through a generic stack model, highlighting security issues across layers but without experimental validation.

Other works have examined other aspects of IoT security. Butun et al. [4] reviewed vulnerabilities, attack vectors, and countermeasures, showing the broad range of threats facing IoT systems, while Mustafa et al. [5] proposed a cross-layer security framework to address these threats. However, these contributions remain at the theoretical level and do not include

protocol-level experimentation. Similarly, Singh et al. [6] examined lightweight encryption methods for IoT devices, and Munir et al. [7] proposed authentication and defense strategies, but both do not systematically assess the performance impact of these mechanisms under resource constraints.

Protocol-specific problems have also been the subject of recent research. For example, Hintaw et al. [8] analyzed vulnerabilities in MQTT implementations, identifying vulnerabilities such as weak authentication and retained message misuse. Dandotiya et al. [9] proposed enhancements for CoAP security through theoretical proposals. McAteer et al. [10] discussed AMQP vulnerabilities in IoT environments, focusing on security gaps in RabbitMQ deployments.

In conclusion, eventhough these studies provides a perspective of how important IoT and protocol security, they still lack three important aspects; (i) most surveys remain conceptual without experimental validation, (ii) the performance implications of security mechanisms are rarely examined in constrained environments, and (iii) protocol-specific vulnerabilities are studied in isolation rather than through a unified, cross-layer approach. This paper addresses these gaps through experimental analysis of four widely used IoT protocols, systematic vulnerability–mitigation mapping, and performance evaluation of defense mechanisms under realistic traffic conditions.

III. Methodology

A. Protocol Selection

Four widely used IoT communication protocols were selected for analysis: MQTT, CoAP, AMQP, and HTTP. MQTT and CoAP were chosen as lightweight protocols optimized for constrained environments, while AMQP and HTTP represent more feature-rich options commonly used in cloud-integrated or enterprise IoT systems [2], [3]. This combination allows comparison between protocols that prioritize efficiency and those that emphasize functionality and reliability.

B. Attack Scenarios

The experiments were designed to replicate realistic protocol-specific vulnerabilities that are frequently reported in IoT environments. For MQTT, the study investigated cases of missing authentication, retained message abuse, and topic flooding, reflecting misconfigurations that leave deployments exposed [8]. CoAP experiments focused on spoofing and replay attacks achieved through crafted UDP packets, which tested server robustness and client response validation [9]. For AMQP, the analysis targeted two common weaknesses in RabbitMQ: the use of default guest credentials and message queue flooding [10]. HTTP testing considered both transport and application-level issues, including plaintext credential leakage in unencrypted sessions and susceptibility to CSRF-style exploits [4]. All experiments can be seen in Table I were carried out in a controlled laboratory environment using protocol clients and servers, together with tools such as Wireshark, Scapy, MITMproxy, and RawCap. Each attack was validated through packet captures and system-level observations such as broker crashes, performance slowdowns, or credential exposure.

TABLE I
Attack Scenarios Evaluated

Protocol	Vulnerability Tested	Tools Used
MQTT	No authentication, retained message abuse, flooding	Mosquitto, Paho client, Wireshark
CoAP	Spoofed responses, replay, malformed packets	CoAPthon, Scapy, RawCap
AMQP	Default guest account, queue flooding	RabbitMQ, Pika client
HTTP	Plaintext login, CSRF attacks	Python requests, MITMproxy

C. Mitigations Applied

For every identified vulnerability, a security mechanism was implemented to evalute if the attack feasibility will reduce. MQTT and AMQP were tested with enforced authentication and fine-grained access control lists, while TLS and DTLS encryption were enabled across all protocols to prevent credential interception and message tampering. Broker-level defenses such as rate limiting and queue length restrictions were introduced to mitigate flooding-based attacks. For CoAP, lightweight replay protection was applied by adding response validation and caching mechanisms. These mitigations were selected according to existing standards and recommendations [5]–[7], and were deployed in minimal configurations to reflect the constraints of resource-limited IoT environments.

D. Performance Metrics

To quantify the trade-offs between added security and system efficiency, each protocol was evaluated under two configurations: an unsecured baseline with default settings and a secured setup with mitigations enabled. Similar to prior IoT protocol studies [2], [5], performance was assessed using packet analysis to measure network overhead, retransmissions, and latency. System resource usage, including CPU and memory utilization, was also monitored in line with previous evaluations of lightweight cryptographic and defense mechanisms [6], [7]. Tests were conducted under both low-load (50 packets) and high-load (5000 packets) conditions to capture the scalability of security measures.

IV. Results

A. Vulnerability Validation and Mitigation

1) MQTT: With default configurations, the broker allowed anonymous connections, enabling unauthorized clients to publish and subscribe. Retained message abuse was confirmed, as malicious payloads persisted in the broker and were delivered to new subscribers even after the attacker disconnected. Topic flooding overwhelmed the broker under high load, leading to increased latency and dropped messages. When authentication and access control lists (ACLs) were enforced, unauthorized access was blocked and retained message abuse was no longer possible. Rate limiting also reduced the effect of flooding by constraining throughput.

2) CoAP: Sending malformed packages to CoAP resulted in server shutting down even if the package was legit. Replay and spoofing attacks were successfully executed by injecting crafted UDP packets. In the default configurations, servers processed duplicate requests and accepted forged messages, highlighting weaknesses in minimal packet validation. Two mitigation strategies were tested. First, AES encryption was applied to secure messages, which successfully blocked spoofed packets but increased packet counts and message size, particularly under 5000 messages, and introduced noticeable latency overhead. Second, a replay cache was implemented, which effectively filtered duplicate packets without altering traffic volume or CPU consumption.

Although DTLS is the standard security mechanism for CoAP in the literature [9], it was not implemented in this study due to incompatibility with DTLS libraries and deployment constraints in the experimental testbed. Instead, AES encryption was selected as a practical lightweight alternative that could be integrated into the existing setup without requiring full DTLS support. This approach enabled the validation of message confidentiality and integrity while still allowing performance measurements under constrained conditions.

3) AMQP: In RabbitMQ's default configuration, the guest account allowed administrative access when not disabled, enabling full unauthorized control of queues. Queue flooding was also validated, as the broker became unstable when processing sustained high-volume traffic, eventually exhausting memory and dropping connections. Mitigations were applied by disabling the guest account and enforcing user authentication, which eliminated unauthorized access. Queue length restrictions prevented flooding, and unlike other mitigations, it did not affect performance. TLS encryption was also applied to secure credentials, preventing their exposure in plaintext traffic.

4) HTTP: With default configurations a simple HTTP packet was sent and verified that all conversation happens in plaintext. A CSRF attack was also demonstrated, where forged requests were accepted without additional validation. Migrating to HTTPS prevented credential leakage during transmission, while the addition of CSRF tokens ensured that forged requests were rejected. Together, these mitigations eliminated the risks of both eavesdropping and request forgery in HTTP-based IoT communication.

In the Table II all summary of vulnerabilities and mitigations applied can be seen.

TABLE II
SUMMARY OF VULNERABILITIES AND MITIGATIONS

Protocol	Vulnerability	Mitigation
MQTT	No authentication, retained message abuse, flooding, plaintext traffic	Authentication & ACLs; Retained message restriction; Rate limiting; TLS
CoAP	Malformed packets, replay, spoofing, no encryption	Exception handling; Replay cache; AES-GCM encryption
AMQP	Default guest account, queue flooding, no TLS	User hardening; Queue limits; TLS
HTTP	Plaintext login, CSRF, spoofing	HTTPS/TLS; CSRF tokens; Static ARP entries

B. Performance Impact Evaluation

The assess the efficieny of the determined mitigations according to the vulnerabilities, each protocol was tested under high load(5000 messages) and low load(50 messages). For every run, packet counts and byte volumes were captured alongside server-side CPU usage, using the same controlled environment as vulnerability testing to ensure comparability.

1) MQTT: Authentication & ACLs: Enforcing client authentication and topic-level ACLs introduced negligible network overhead. In fact, authentication improved efficiency: at 5,000 messages, CPU usage dropped by 25.8% (1.24% $\rightarrow$ 0.92%), and total bytes decreased by $\tilde{4}$%, since authenticated sessions benefit from cached authorization and optimized broker pathways. Strict ACLs added a modest 7% increase in bytes, with no measurable CPU penalty.

DoS Mitigation: Connection limits, queued message caps, and packet size restrictions provided strong resilience with virtually no performance cost. CPU overhead stayed below 1%, and packet/byte counts remained stable. So, no performance difference observed before or after mitigation application.

Retained Message Control: Restricting retained messages introduced measurable overhead, increasing MQTT bytes by 19% and total traffic by 7% across both low and high loads. CPU usage also rose slightly (0.1% at 50 messages; +17.7% at 5,000 messages). This mitigation carries a consistent performance cost in both bandwidth and processing but its still negligible.

TLS Encryption: TLS emerged as the most resource-intensive mitigation for MQTT. At 50 messages, MQTT bytes increased by +228.8% (2,310 $\rightarrow$ 7,576), driving a +44.9% rise in total bytes, while CPU usage increased only slightly to 0.27%. At 5,000 messages, the impact scaled predictably: MQTT bytes grew by +66.5% (238,910 $\rightarrow$ 397,642) and total bytes by +24.9%, with CPU usage rising modestly from 1.24% to 1.60% (+29%). These results show that TLS imposes a substantial bandwidth penalty in MQTT but only a minor CPU cost, making encryption feasible in many deployments but potentially challenging for bandwidth-constrained IoT networks.

2) CoAP: Exception Handling (Malformed Packets): At low load, CPU usage dropped 29% due to early rejection of invalid input, but at high load, validation overhead increased CPU by +66%, revealing scale-dependent trade-offs. Network overhead was negligible.

Replay Protection: The anti-replay mechanism showed a clear scale-dependent effect. At low load (50 messages), it actually improved efficiency, with CPU usage dropping from 0.82% to 0.60 (-27%) because structured message validation replaced the less optimized baseline handling. At high load (5,000 messages), however, the server needed to constantly check and update its replay cache, pushing CPU usage from 4.74% to 6.75% (+42%). Network overhead remained minimal

(¡1% byte increase), confirming that the cost of replay protection is almost entirely computational. In practice, this means replay defense is lightweight for small IoT deployments but becomes progressively heavier as traffic volume grows.

AES-GCM Encryption: Applying AES-256-GCM introduced predictable overhead in both network traffic and CPU usage. At 50 messages, total bytes rose from 5,290 to 12,450 (+135%) and server CPU usage increased from 0.82% to 1.31% (+59.8%). At 5,000 messages, the cost scaled further, with bytes increasing from 538,890 to 1,245,249 (+131%) and CPU usage nearly doubling from 4.74% to 9.23% (+94.9%). This overhead stems from the added 28-byte nonce and authentication tag appended to each packet and the cryptographic operations required for encryption and verification. While AES-GCM ensures confidentiality and integrity, the results demonstrate that it is among the most resource-intensive mitigations, particularly for CPU-constrained IoT deployments.

UDP Flood Mitigation: Iptables rate limiting and reverse-path filtering imposed no measurable cost under normal traffic, since filtering activated only under abnormal packet bursts.

3) AMQP: User Hardening & Queue Policies: Removing the default "guest" account and applying queue length limits had zero measurable performance impact in normal operation, since these mitigations only trigger under attack. Flood protections (max-length, overflow rules) did not alter throughput in baseline scenarios.

TLS Encryption: TLS significantly increased overhead. At 50 messages, bytes rose +28% and CPU usage +40%. At 5,000 messages, byte overhead nearly doubled (+93%) and CPU surged +163% (1.9% → 5.0%), reflecting heavy cryptographic processing and RabbitMQ's enterprise-grade TLS stack. This makes TLS protective but resource-intensive, less suitable for constrained IoT nodes.

4) HTTP: CSRF Protection: Token-based CSRF mitigation added virtually no overhead (¡0.5% CPU variance, zero packet change), proving highly efficient. TLS/HTTPS: Encryption produced the highest overhead of all protocols. At 50 messages, total bytes rose +642% and CPU increased slightly (+23%). At 5,000 messages, bytes rose +261% while CPU appeared to decrease (1.94% → 1.09%); this anomaly was traced to throughput reduction, not efficiency gains. In practice, TLS on HTTP imposes very high network cost, making it the least efficient option in bandwidth-sensitive IoT.

To better illustrate cross-protocol differences, Figure 1 summarizes the relative encryption overhead on total bytes under high load. As shown, MQTT's TLS overhead remains moderate compared to the much higher costs in CoAP, AMQP, and HTTP.

C. Cross-Protocol Comparison

1) Security Implementation Difficulty:: MQTT and AMQP were relatively straightforward to secure. With MQTT's simplicity authentication and ACLs were easy to configure, though the risk of forgetting defaults is high. AMQP provided rich built-in options like a user interface which made everything easier. CoAP proved the hardest to secure: DTLS support was

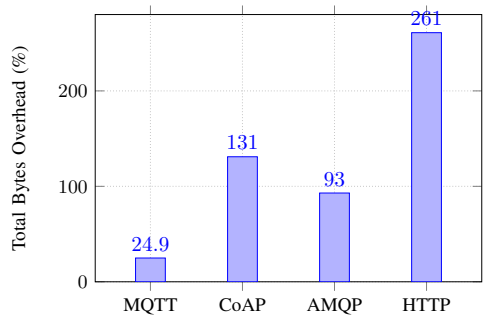


Fig. 1. Encryption overhead on total bytes at high load (5,000 messages): MQTT (TLS), CoAP (AES-GCM), AMQP (TLS), HTTP (TLS).

unstable, AES-GCM had to be substituted, and lightweight servers often crashed on malformed input. HTTP was easy to secure at the transport layer via TLS, but higher-layer protections like CSRF tokens, ARP defenses had to be added manually.

2) Performance in Constrained Environments:: MQTT proved to have thelowest overhead after adding the mitigtions, making it a great fit for low-power devices. CoAP, although lightweight by design, became unstable and inefficient under security measures, limiting its viability. AMQP had strong safety features but paid large CPU and bandwidth expenses, making it primarily appropriate for edge or backend systems. HTTP consistently had the heaviest overhead due to verbose message formats and TLS, making it the least practical for constrained IoT .

3) Security vs. Overhead Trade-off:: MQTT achieved the best balance, with mitigations like authentication even improving CPU efficiency by rejecting unauthorized clients early. AMQP offered robust resilience but at high cost, with TLS alone raising CPU usage by over 160% under load. CoAP required non-standard workarounds (AES-GCM), and overhead scaled disproportionately with traffic. HTTP lacked IoT-specific defenses and relied heavily on TLS, which inflated bandwidth usage massively.

4) Resilience to Cross-Layer Attacks:: All protocols showed cross-layer weaknesses. MQTT and AMQP without authentication allowed network-layer floods to escalate into unauthorized application actions. CoAP's UDP base made it highly vulnerable to spoofing, replay, and malformed packets crashing servers. HTTP, without encryption or access control, remained open to MITM, credential theft, CSRF, and injection. These results confirm that effective IoT security must be layer-aware, as failures at one layer often cascade to others.

Overall, these observations are consolidated in Table III, which provides a qualitative comparison of the four protocols. The matrix highlights that MQTT offers the best balance of security and efficiency for constrained IoT devices, while CoAP becomes fragile under mitigations, AMQP is too heavy for resource-limited settings, and HTTP remains the least

TABLE III
CROSS-PROTOCOL SUITABILITY MATRIX

Protocol	Easy to Secure?	Light Overhead?	Secure Defaults?
MQTT	Yes	Yes	No
CoAP	Hard	Mixed	No
AMQP	Medium	Heavy	Partially
HTTP	Easy	Heavy	No

practical despite its ease of use.

V. CONCLUSION AND FUTURE WORK

This work developed a cross-layer methodology to identify, exploit, and mitigate vulnerabilities in four IoT protocols—MQTT, CoAP, AMQP, and HTTP—and to quantify the performance cost of these mitigations under realistic workloads. The study confirmed that attacks at one layer can propagate to others, and that all four protocols are insecure by default. Yet, in the case of MQTT lightweight measures such as authentication, early request rejection, and broker-side throttling not only improved resilience but in some cases reduced CPU load, challenging the assumption that "security always costs performance." TLS and AES-GCM provided strong confidentiality but introduced significant network and CPU overheads, highlighting the importance of context-aware deployment.

Three main contributions emerge: (i) a structured, repeatable framework for cross-layer testing, (ii) empirical evidence that some mitigations can be both secure and efficient, and (iii) practical insights into deployment realities such as DTLS instability and AMQP cluster risks. Together, these findings provide practitioners with actionable guidance on which mitigations to prioritize and protocol designers with evidence for secure-by-default enhancements.

Future work should extend this methodology to physical IoT hardware to capture timing, energy, and radio-layer effects, expand protocol coverage to ecosystems like LoRaWAN, Zigbee, and DDS, and integrate automated orchestration tools for large-scale attack/mitigation testing. Another promising direction is to prototype a secure-by-default MQTT profile with built-in authentication, bounded state, and AEAD encryption, and to apply fuzzing and stricter flow-control rules for resilience against protocol misuse.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in Proc. IEEE Int. Syst. Eng. Symp. (ISSE), Vienna, Austria, 2017, pp. 1–7, doi: 10.1109/SysEng.2017.8088251.

[3] J. Tournier, F. Lesueur, F. Le Mouël, L. Guyon, and H. Ben-Hassine, "A survey of IoT protocols and their security issues through the lens of a generic IoT stack," Internet of Things, vol. 16, p. 100264, 2021.

[4] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures," IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 616–644, 2020.

[5] R. Mustafa, N. I. Sarkar, M. Mohaghegh, and S. Pervez, "A cross-layer secure and energy-efficient framework for the Internet of Things: A comprehensive survey," Sensors, vol. 24, no. 22, p. 7209, 2024.

[6] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," J. Ambient Intell. Humaniz. Comput., vol. 15, no. 6, pp. 1625–1642, 2024.

[7] A. Munir, I. A. Sumra, R. Naveed, and M. A. Javed, "Techniques for authentication and defense strategies to mitigate IoT security risks," J. Comput. Biomed. Informatics, vol. 7, no. 1, pp. 377–388, 2024.

[8] A. J. Hintaw, S. Manickam, M. F. Aboalmaaly, and S. Karuppayah, "Mqtt vulnerabilities, attack vectors and solutions in the internet of things (iot)," IETE Journal of Research, vol. 69, no. 6, pp. 3368–3397, 2023.

[9] N. Dandotiya, A. S. Dandotiya, R. Dubey, S. Gupta, S. Sharma, and A. K. Sharma, "Enhancing coap for secure iot communication," in 2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N), Greater Noida, India, 2024, pp. 1167–1172

[10] I. N. McAteer, M. I. Malik, Z. Baig, and P. Hannay, "Security vulnerabilities and cyber threat analysis of the AMQP protocol for the internet of things," in Proceedings of the 15th Australian Information Security Management Conference. Perth, Western Australia: Edith Cowan University, 2017, pp. 70–80. [Online]. Available: https://ro.ecu.edu.au/ism/203

Behavioral Authentication: Evaluation of Reliability in Contemporary Web Security

Kebal Prasad Bhandari
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, Republic of North Macedonia
kebal.prasad.bhandari@students.finki.ukim.mk

Ivan Chorbev
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, Republic of North Macedonia
ivan.chorbev@finki.ukim.mk

***Abstract*—Cyberattacks are becoming increasingly sophisticated day by day. As a result, the Authentication of users is very important in a secure system. The authentication is performed by something that a real user knows, has, or is. The last one is called biometrics, which is most linked with fingerprint and face modalities. You can also authenticate a user by his or her behavior, called behavioral biometrics. Behavioral authentication based on user behavior patterns like keystroke dynamics and mouse interactions has become a promising technique to enhance web security. This paper argues that the long-term reliability dimension operationalized as the ability to maintain consistent performance in real-world environments with diverse conditions has not been thoroughly explored. In this paper, through a systematized review of multiple studies related to behavioral authentication, we examine the impact of behavioral drift, variability in environmental conditions and attacks on the precision and reliability of behavioral authentication systems. Additionally, from a network and internet perspective, issues such as packet loss, latency, and insecure communication channels can further challenge and affect real-time performance and data quality, reducing the reliability confidence. Also, other user's psychological factors can also affect and impact the user behavior. The outcome shows that single-modal techniques suffer a reduction of up to 15% in open environments, while multi-modal approaches (keystroke, mouse movements or other behavioral signals) maintain error rates below 2% consistently.**

***Index Terms*—Behavioral authentication, keystroke dynamics, mouse dynamics, multi-modal fusion, reliability**

I. Introduction

Over the last two decades, measures for behavioral authentication systems like keystroke, mouse dynamics and touch interactions have progressed from preliminary feasibility studies [1] to full-scale investigations of the methods used [2] which clearly demonstrate their progress. At the same time, evaluating a continuous authentication system has also proven vital to accurately demonstrate its effectiveness. Bours and Mondal performed extensive performance assessments with a focus on long-term reliability and environmental factors, thereby emphasizing the requirement of a flexible testing paradigm [3]. In contrast, Wang et al. enriched the discussion by proposing state-of-the-art evaluation methods that are marked by significantly robust statistical measures with the ability to transmit genuine real-world performance and reduce systematic errors in the field of behavioral authentication research [4]. Additionally, Zhou et al. further developed this perspective by shedding light on ongoing shortcomings in different evaluation methods, which, as a result, clearly signals the need for standardized benchmarks and detailed error rate investigations in order to better understand real-world performance scenarios [5], [6]. It's also important to consider other non-technical aspects like psychological factors and environmental conditions if we want better reliability of the system. For example, stress and sadness can reduce precision and speed, while a positive mood at the same time can improve coordination [7]. Different environmental conditions like cold or hot weather can also impact the user's cognitive and behavioral abilities, which are crucial to behavioral biometrics systems [8].

A. Background

The growing sophistication of cyber-attacks seen over the last few years has placed even more demands on advanced security on web security infrastructures. Traditional approaches such as passwords and physical token-based methods have been found vulnerable against a wide range of attacks like phishing, credential compromise and brute-force attacks [9]. These static credential-based techniques show limited effectiveness against adaptive attackers. Hence, the search for more dynamic alternatives is a must. Behavioral authentication presents itself as a strong replacement by leveraging unique patterns of human-computer interaction like keystroke dynamics [10], mouse movement [11], and graphical user interface (GUI) interaction [12] to provide a non-intrusive and persistent security solution that efficiently balances protection and user experience altogether [3], [13].

Biometrics is divided into two main categories: behavioral and physiological. Physiological Biometrics contains features like fingerprints and facial characteristics, which have long dominated in authentication research. However, behavioral biometrics focuses on the user's actions by looking at patterns that emerge from how people interact with the computer systems. For example, keystroke dynamics capture the rhythm and time of typing metrics such as key hold time and flight time rather than the content typed [14]. These characteristics are influenced by a blend of physiological factors such as hand geometry and motor skills as well as human neurological factors which include typing proficiency and cognitive habits rendering them distinct across individuals [15], [16].

Li and Yu proposed privacy-preserving methodologies for behavioral biometric authentication utilizing encryption techniques, substantially mitigating risks associated with data leakage and misuse in continuous authentication scenarios [17]. Moreover, predictive analytics techniques examined by Drouin and Boyd have been instrumental in proactively managing authentication systems demonstrating significant benefits while detecting and mitigating suspicious user behaviors [13].

To illustrate, Fig.1 depicts the pattern of keystroke dynamics for two different successive key presses, key A and key B. It includes critical metrics like, *Hold Time* (duration a key is pressed), *Press-Release Time* (interval from pressing to releasing a key), *Release-Press Time* (interval between releasing and pressing keys), *Press-Press Time* (interval between consecutive key presses) and *Release-Release Time* (interval between consecutive key releases). These characteristics as discussed by [15], provide a solid foundation for identifying users based on their typing rhythms which are unique and influenced by physical and cognitive characteristics. Likewise, mouse dynamics record movement speed, click sequences and cursor trajectories, forged by similar physical and experiential variables [16].

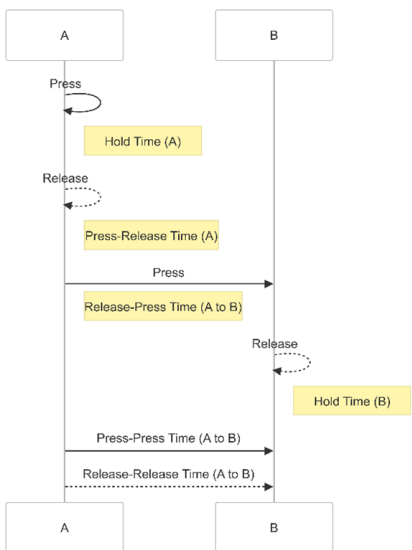


Fig. 1. Important Features of Keystroke Dynamics. Adapted from [14].

Behavioral authentication systems can be static, one-time authentication or continuous - continuous session monitoring (such as anomaly detection during user activity) [18]. Static systems require only a little data and hence are adequate for shallow verification whereas continuous ones require richer and more intricate data to effectively model the behavior of a user over the long term offering an active defense against various threats like session hijacking or unauthorized access [3]. Putting together different behavior patterns like how people type, move their mouse and interact with computer interfaces into one system makes it more dependable, as Fig. 2 shows. This figure displays a full check-in process: collecting data from typing, mouse usage and interface actions then extracting key features and preparing them to select a model (using deep learning or hybrid models) combining all the signals deciding who's using the system, and evaluating its performance using metrics like how often it allows access to the wrong person or denies access to the right one. In the end, it figures out if the user is who they say they are.

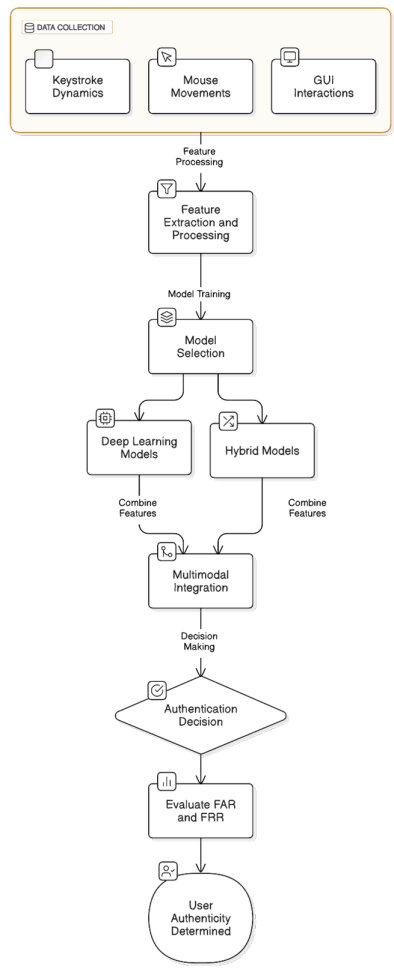


Fig. 2. Multi-modal Behavioral Authentication System.

Dabrowski et al. [19] reported that combining keystroke data alongside mouse movements data can greatly decrease

error rates and enhance accuracy, which helps to boost system reliability and accuracy. Mondal and Bours study also suggested a hybrid strategy that combines several data sources to address user behavioral drift. As a result, the behavioral authentication system obtains better accuracy and consistency, even for a single modal-based system [20]. Roy et al. study investigated the robustness and effectiveness of multi-modal behavioral systems in different dynamic situations [21].

At the same time, advances in machine learning algorithms have significantly enhanced the performance of behavioral authentication systems in recent years. Alshehri and Traoré research showed that recurrent neural networks can exhibit high resilience against behavioral drift, thus ensuring high reliability in continuous authentication scenarios [22]. Chen and Lai's research showed that the combination of convolutional and recurrent neural networks can learn complex user patterns and help enhance the system to be better [23]. Such complex models illustrate the ways in which deep learning methods can enhance behavioral authentication systems. However, such complex behavioral information also brings about issues of privacy that need to be resolved through proper protection mechanisms [17]. Kar et al. (2023) went a step further in this respect by suggesting continuous user authentication through multi-modal systems for enhancing security and reliability [24], whereas Panasiuk and Saeed (2017) suggested a risk-based static authentication system using behavioral biometrics and session context analysis [25]. Despite these advances, the attainment of long-term reliability, evinced by stability in the face of diverse unpredictable real-world situations, remains a major impediment to large-scale deployment [26]. As cyber threats continue to evolve, it is very crucial that such critical systems maintain accuracy and robustness outside controlled environment settings.

B. Problem Statement

Behavioral authentication system shows promise in laboratory-controlled environments by recording an outstanding accuracy of 91.67% by utilizing deep multilayer perceptrons in keystroke dynamics analysis [27]. Nevertheless, application to real-world contexts reveals significant discrepancies due to multiple factors such as behavioral drift and environmental fluctuation, along with network constraints and susceptibility to different attacks [3], [26]. The evidence implies that single modal systems, relying only on keystroke dynamics or mouse movements are subject to declines in performance up to 15% when put in uncontrolled environments [28], thus leaving serious doubts over their scalability and reliability in heterogeneous and dynamic contexts [29]. Environmental and psychological variations like cold, render typing and mouse behavior unpredictable. Multi-modal systems, however, tend to be more robust and reliable. For example, Bhattacharya et al. [30] found error rates to be below 3% using keystroke and mouse behavior collectively, which shows that the multi-modal system is effective and reliable. Through multi-modal based approaches and using sophisticated machine learning techniques promises well [19], [31], their performance over the longer term and in uncontrolled environments is inadequately studied, which only points towards the intensive exploration of their long-term effectiveness.

C. Research Questions and Objectives

This review synthesizes findings from 32 scientific research papers to systematically assess the reliability of behavioral authentication systems primarily focusing on web security and addressing two major questions:

1) How do factors such as behavioral drift, environmental variability, network challenges and attacks impact the long-term reliability of behavioral authentication methods?
2) How do single-modal systems compare to multi-modal systems in terms of performance across diverse, real-world datasets and varying environments?

The main focus of our study is to evaluate and understand how these systems work in a real-world environment and find key differences between progress in theory and actual results without proposing new solutions. By looking into these areas, the study aims to build a solid foundation to understand the advantages and disadvantages of behavioral authentication systems.

D. Significance of the Study

This study responds to the growing demand for a reliable and secure authentication system that addresses the challenges posed by modern web security threats. This provides an important implication for both researchers working to improve these systems and practitioners looking for the best and most effective ways to use them. As a result, helps to understand the right balance between theoretical and practical realization. It also lists the important factors affecting it, which makes it possible for more research to be done in this area. Finally, this study encourages the use of adaptive user modeling to reduce the effects of behavioral drift. It shows that we need standardized ways to test how well they work in different real-world situations. It also highlights the need for standardized evaluation methods to establish the reliability under various practical applications.

II. Methodology

This study conducts a systematic literature review(SLR) from 32 scientific research papers focusing on behavioral authentication systems based on systematic research guidelines. The methodology follows a scientific, transparent, and comprehensive analysis of the reliability of behavior-based authentication systems, primarily focusing on web application security contexts. This research focused on these few key questions: what factors are affecting system reliability, how do single-modal and multi-modal systems compare and how do performance metrics like accuracy, FAR, FRR and EER reflect on system effectiveness and reliability. In addition, this study also examines ML Techniques, testing environments, different datasets, real-world challenges and different types of attacks.

A. Scientific Framework and Metrics

For the comparison and contrast of the effectiveness and performance of behavior-based authentication systems, several standard test measures are employed:

- **False Acceptance Rate(FAR):** The probability of a malicious or an unauthorized user being authenticated inappropriately, defined by:

$$\text{FAR} = \frac{\text{Number of false acceptances}}{\text{Total number of unauthorized attempts}} \times 100\% \quad (1)$$

  A low FAR is critical for security, ensuring unauthorized users are rejected [3].
- **False Rejection Rate(FRR):** The probability that an authorized user is incorrectly rejected, calculated as:

$$\text{FRR} = \frac{\text{Number of false rejections}}{\text{Total number of genuine attempts}} \times 100\% \quad (2)$$

  A low FRR increases usability by reducing real legit user rejections [3].
- **Equal Error Rate(EER):** The point where FAR is equal to FRR, indicates an equitable system performance. Lower EER means greater reliability, as it provides the best trade-off between security (low FAR) and usability (low FRR) [18].
- **Accuracy:** The typical performance metric for behavioral authentication classification models is given as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3)$$

  where True Positives(TP), True Negative(TN), False Positive(FP) and False Negative(FN) are correct and incorrect results. Higher accuracy here means better overall system performance.

B. Scope and Limitations

This work focuses on behavioral authentication systems for web security, not including physiological biometrics like fingerprints or face recognition. We focus on keystroke dynamics, mouse movements and User Interface interaction patterns, which are better suited for continuous desktop or web-based authentication settings [23], [30]. Single-modal systems (e.g. keystroke-only or mouse-only) achieve an average accuracy of 85-92% and do not perform well in uncontrolled environments because of noise and behavioral variability [19], [27]. Multi-modal systems are more consistent, commonly with FAR $< 1\%$ and FRR $< 2.5\%$ [19], [30]. Focusing on continuous authentication in web applications, we accept that our findings might not explicitly generalize to different modalities (e.g., mobile sensor-based behavior) or other various types of deployment settings.

Furthermore, our SLR methodology relies on the quality of results, experimental settings, datasets, testing approaches, threat models, etc, reported in the papers we have chosen to proceed with. We work around this by normalizing data where possible and by comparing qualitative trends instead of absolute numbers. Second, even as we outline possible improvements and protection mechanisms (like privacy-preservation solutions or defenses against adversarial attacks), recommendations or speculation of novel solutions are outside the scope of our inquiry [23], [30]. Our aim is to discover current shortcomings and reliability-related issues in order to inform future research efforts.

C. SLR Process and Study Selection

Following best practices for SLR, we defined specific research questions and inclusion criteria before searching relevant digital libraries (IEEE Xplore, ACM Digital Library, etc.). The keywords used included combinations of "behavioral authentication", "continuous authentication", "keystroke dynamics", "mouse dynamics", "web security", "long-term reliability" and "behavioral biometrics". We initially retrieved 53 papers and then applied different screening criteria: we included only those studies focusing on behavioral authentication with empirical evaluations of performance or reliability (excluding purely theoretical works). After title and abstract screening, 43 papers were eligible for full-text review, from which we further excluded those not centered on web/desktop environments or lacking sufficient experimental detail. This resulted in 32 primary studies that form the basis for our analysis. Each of the selected papers was reviewed carefully in detail and extracted data regarding their methodology, dataset, features, classifier and results. Since every study that was chosen for this analysis was released between 2005 and 2023, our examination includes both classical (such as early keystroke studies) and the most recent modern developments (such as deep learning-based methodologies). Based on their main objectives performance evaluation, multi-modal system design and tackling certain particular issues (such as user behavior drift or various threats), we have grouped the studies.

D. Data Analysis

We employed both quantitative and qualitative analysis techniques to synthesize the findings:

- **Quantitative Synthesis:** We studied different performance metrics like accuracy, FAR, FRR, EER, etc. from each study under comparable scenarios and visualized trends. For example, we compared reported error rates for single-modal with multi-modal systems across studies to identify the typical performance gaps. We also studied and analyzed how performance degrades over time or under different attack simulations (like replay attacks) as reported from the extracted data and various psychological and environmental situations.
- **Qualitative Synthesis:** We performed an analysis of discussion sections in the studies to extract common challenges and proposed solutions. Recurring themes included the impact of user behavior change over time, the benefits of combining modalities, the influence of environmental factors and the trade-offs between security and usability.

The combination of these analyses allowed us to answer the research questions comprehensively: quantitative results illus-

trate the extent of reliability issues and improvements while qualitative insights explain why those issues are occurring and how researchers are aiming to address them.

E. Scope and Limitations

The current research explores behavioral authentication systems in the field of web security, highlighting the importance of long-term reliability, accuracy and robustness for real-world applications under various critical circumstances. Prominent limitations include reliance on reported results, inconsistency in dataset quality, and the lack of real-time testing for some models. However, by combining controlled and real-world investigations, this research offers a balanced view of both theoretical and practical performance.

III. Results and Discussion

This section synthesizes findings from a systematic review of 32 studies on behavioral authentication systems, categorized into performance metrics, machine learning techniques, multi-modal and single-modal approaches, different testing environments, vulnerabilities, trends, and their implications.

A. Performance Metrics

Behavioral authentication systems are evaluated using accuracy, FAR, FRR and EER. Multi-modal systems combining keystroke dynamics, mouse movements and GUI interactions can achieve over 98% accuracy, with FAR below 1% and FRR under 2% [23], [30]. Single-modal systems (e.g. keystroke-only or mouse-only) average 85-92% accuracy, which struggles in uncontrolled settings due to noise and behavioral variability [19], [27]. Multi-modal systems report EER as low as 0.5%, which highlights their effectiveness and robustness.

B. Machine Learning Techniques

Machine learning in behavioral authentication systems has evolved from traditional static approaches to the latest dynamic deep learning methods.

1) **Traditional Algorithms**: Support Vector Machines (SVM), Random Forests and XGBoost dominate early studies in the field. They perform really well on small datasets but really struggle with complex, sequential data achieving $\sim$85% accuracy in single-modal setups.
2) **Deep Learning Techniques**: Recent studies focus on RNN, LSTM and Transformers for capturing and monitoring various temporal patterns and achieve over 96% accuracy, while Transformers excel in scalability and better applicability.

C. Multi-Modal vs Single-Modal Approaches

Multi-modal system uses multiple traits at a time. This helps them be better than a single-modal system.

- Accuracy averages 98% by manipulating complementary features (keystroke, mouse, GUI) [23], [30].
- FAR ($< 1\%$) and FRR ($< 2\%$) are lower due to enhanced robustness [23], [30].

Single-modal systems are simpler but more prone to noise, environmental variations and adversarial attacks [3], [32]. Keystroke-only models or Mouse dynamics-only models degrade significantly across devices and network latencies [19], [27].

D. Testing Environments and Datasets

Performance varies across environments:

1) **Controlled Environments**: Controlled Lab settings achieve over 95% accuracy due to minimal noise and static-controlled environment settings.
2) **Real-World Environments**: Accuracy drops to 85-90% due to network latency, device heterogeneity and various environmental noise which underscores the need for advanced adaptive models.
3) **Datasets**: Public datasets like CMU Keystroke Benchmark and Mouse Dynamics Challenge are common but lack demographic and environmental diversity, limiting broad applicability. Proprietary datasets yield higher accuracy but lack various standardizations.

E. Vulnerabilities and Challenges

Key issues include:

1) **Behavioral Drift**: User behavior changes over time, which increases FAR and FRR. Accuracy can drop 10-15% within months without retraining [32].
2) **Adversarial Attacks**: Replay and imitation attacks increase FAR by up to 40% in open systems. Neural networks are vulnerable to adversarial attacks without proper training datasets [3].
3) **Environmental Variability**: Behavioral Authentication System performance can be significantly impacted by various devices, network latency, users interactions, variations in mood, and changes in weather conditions like hot and cold temperatures, etc., that degrade the system's reliability.

TABLE I
Threat Model Matrix for Behavioral Authentication Systems

Threat	Example Attack	Reported Effect
Imitation	Mimicking typing/mouse rhythm	FAR $\uparrow$ by 20-30%
Replay	Reusing recorded input traces	FAR $\uparrow$ by up to 40%
Device Variability	Different keyboards/mice	Accuracy $\downarrow$ 10-15%
Behavioral Drift	Long-term user change	FRR $\uparrow$ after months

F. Emerging Trends and Insights

1) **Machine Learning Evolution**: Deep learning (e.g., Transformers) outperforms traditional models by capturing complex patterns and at the same time improving the system's reliability.
2) **Multi-modal Systems**: Combining modalities enhances resilience and accuracy with advanced feature fusion techniques showing realistic promise.
3) **Privacy-Preserving Methods**: Federated learning and homomorphic encryption address privacy concerns enabling secure model training without compromising user data.

G. Implications and Recommendations

Behavioral authentication offers a secure and advanced user-friendly alternative to traditional authentication methods. Future research should:

- Develop adaptive models to counter user's behavioral drifts.
- Adopt privacy-preserving and highly secure techniques to meet ethical and regulatory standards.
- Standardize evaluation metrics for consistent study comparisons.

IV. Conclusion

Behavioral authentication systems using keystroke dynamics, mouse movements, and GUI interactions present an encouraging supplement to conventional security systems by implementing ongoing, user-specific authentication. This compilation of 32 papers concludes that multi-modal solutions tend to have more than 98% accuracy, FAR less than 1%, and FRR less than 2%. Single-modal solutions tend to deteriorate in practical applications because of environmental variation and behavioral drift. More sophisticated deep models like RNNs, LSTMs, and Transformers have demonstrated improved resilience by extracting more complex temporal and contextual features, and privacy-preserving techniques (like federated learning) prevent data leakage attacks. However, there are still some challenges: system resilience is compromised by adversarial attacks and limited dataset diversity, and network problems (like latency, packet loss, or insecure communication channels) can compromise real-time performance and data integrity. User's psychological factors and environmental variability can also impact the implementation and long-term reliability of the system. Future work must address adaptive user modeling to counteract drift, privacy-focused resilient architectures, and test protocols standardized to accept heterogeneous network conditions and threat models so that the transition from controlled laboratory settings to real-world deployments is effortless.

References

[1] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, 2000.

[2] P. S. Teh, S. Yue, and J. Hu, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, vol. 2013, p. 408280, 2013.

[3] P. Bours and S. Mondal, "Performance evaluation of continuous authentication systems," *IET Biometrics*, vol. 4, no. 4, pp. 217–223, 2015.

[4] D. Wang and P. Wang, "Behavioral drift and its impact on biometric systems," *Computers Security*, vol. 111, p. 102664, 2021.

[5] J. Zhou and Y. Zhou, "Evaluating the reliability of multimodal behavioral biometric systems," *ACM Transactions on Applied Perception*, vol. 16, no. 4, pp. 23–37, 2019.

[6] K. Zhou and F. Li, "Evaluating behavioral biometrics under real-world conditions," *Journal of Information Security*, vol. 14, no. 3, pp. 120–135, 2021.

[7] P. Zimmermann, S. Guttormsen, B. Danuser, and P. Gomez, "Affective computing—a rationale for measuring mood with mouse and keyboard," *International journal of occupational safety and ergonomics*, vol. 9, no. 4, pp. 539–551, 2003.

[8] M. Falla, A. Micarelli, K. Hüfner, and G. Strapazzon, "The effect of cold exposure on cognitive performance in healthy adults: a systematic review," *International journal of environmental research and public health*, vol. 18, no. 18, p. 9725, 2021.

[9] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 3, pp. 165–179, 2007.

[10] D. Gunetti and C. Picardi, "Keystroke analysis of free text," *ACM Transactions on Information and System Security*, vol. 8, no. 3, pp. 312–347, 2005.

[11] T. Anusas-Amornkul and T. Wangsuk, "Behavioral biometric authentication based on mouse dynamics," *Procedia Computer Science*, vol. 60, pp. 100–107, 2015.

[12] K. O. Bailey, J. S. Okolica, and G. L. Peterson, "User identification and authentication using multi-modal behavioral biometrics," Air Force Institute of Technology, Tech. Rep., 2014.

[13] M. Drouin and D. Boyd, "Predictive analytics in continuous authentication systems," *Computers Security*, vol. 93, p. 101795, 2020.

[14] J. Jenkins, Q. Nguyen, J. Reynolds, W. Horner, and H. Szu, "The physiology of keystroke dynamics," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8058. SPIE, 2011.

[15] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *2009 IEEE/IFIP International Conference on Dependable Systems Networks*. IEEE, 2009, pp. 125–134.

[16] Y. Zhong and Y. Deng, "A survey on keystroke dynamics biometrics: approaches, advances, and evaluations," *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics*, vol. 1, no. 1-22, p. 2, 2015.

[17] Y. Li and W. Yu, "Privacy-preserving behavioral biometric authentication for continuous user verification," *Future Generation Computer Systems*, vol. 97, pp. 189–200, 2019.

[18] S. Mondal and P. Bours, "A study on continuous authentication using a combination of behavioral biometrics," *Pattern Recognition Letters*, vol. 82, pp. 144–150, 2017.

[19] M. Dabrowski, P. Panasiuk, M. Szymkowski, and K. Saeed, "A multimodal biometric user identification system based on keystroke dynamics and mouse movements," in *Biometric User Identification*. Springer, 2021.

[20] S. Mondal and P. Bours, "A hybrid approach to continuous authentication using keystroke dynamics," *Computers Security*, vol. 56, pp. 77–89, 2016.

[21] P. Roy and S. Mondal, "Hybrid authentication frameworks using behavioral biometrics," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3040–3052, 2021.

[22] A. Alshehri and I. Traore, "Robust behavioral biometric authentication using recurrent neural networks," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 3948–3960, 2017.

[23] S. Chen and H. Lai, "Deep learning approaches for behavioral authentication," *ACM Transactions on Information Systems*, vol. 39, no. 4, p. Article 37, 2021.

[24] P. Kar, K. Bamotra, P. Duvvuri, and S. Mohanan, "Continuous user authentication using multimodal biometrics," *Pattern Recognition Letters*, vol. 170, pp. 140–149, 2023.

[25] P. Panasiuk and K. Saeed, "Risk-based static authentication in web applications with behavioral biometrics and session context analytics," in *Web Application Security*. Springer, 2017.

[26] T. Parker and J. Fraser, "Evaluation of behavioral biometric systems in uncontrolled environments," *Computers Security*, vol. 92, p. 101710, 2020.

[27] A. Andrean, M. Jayabalan, and V. Thiruchelvam, "Keystroke dynamics based user authentication using deep multilayer perceptron," *International Journal of Machine Learning and Computing*, vol. 10, no. 1, pp. 134–139, 2020.

[28] C. D. Murphy, J. Huang, D. Hou, and S. Schuckers, "Shared dataset on natural human-computer interaction to support continuous authentication research," Clarkson University, Tech. Rep., 2022.

[29] W. Shi, C. Wang, Z. Zheng, and X. Cao, "Continuous authentication using mouse dynamics," *Computers Security*, vol. 121, p. 103870, 2022.

[30] S. Bhattacharya and P. Roy, "Continuous authentication using multimodal behavioral biometrics," *Information Sciences*, vol. 607, pp. 563–577, 2022.

[31] S. M. Mousavi and M. H. Abidi, "Enhancing behavioral biometric systems through hybrid models," *Journal of Information Security and Applications*, vol. 66, p. 103158, 2022.

[32] X. Liu and C. Ma, "Adversarial attacks on behavioral authentication models," *IEEE Access*, vol. 8, pp. 12 871–12 880, 2020.

Classification of Web-Based Cyberattacks via IoT

Aysu Maden
Management Information Systems Department
Kadir Has University
Istanbul, Turkey
aysu.maden@stu.khas.edu.tr

Hasan Dağ
Center for Cybersecurity &
Critical Infrastructure Protection (CCIP)
Kadir Has University
hasan.dag@khas.edu.tr

***Abstract*—This study proposes a focused and explainable multi-class intrusion detection system (IDS) to detect four web-based attacks (DDoS, SQL Injection, Cross-Site Scripting (XSS), and Brute Force) that are frequently encountered in IoT environments. Using the ML-Edge-IIoT dataset containing realistic and heterogeneous traffic scenarios, the relevant attack types are meticulously selected and relabeled to create a targeted classification environment. Five different machine learning models (Random Forest, XGBoost, LightGBM, TabNet, and LSTM) are implemented, and the highest success is achieved by the Random Forest model. The decisions of the model are interpreted by SHAP-based explainability analysis, and the protocol-level deterministic features that affect the detection performance are revealed. The system has proven its real-time operation with low latency and minimal resource consumption; in this context, the paper offers a structure that can be integrated into edge devices. To the best of our knowledge, this study presents the first explainable and multi-class IDS solution that focuses on these four web attack types in IoT systems.**

***Index Terms*—IoT, web attack types, SQLi, XSS, Brute Force, DDoS.**

I. Introduction

With the acceleration of digitalization, the number and complexity of cyberattacks have increased significantly, and IoT devices, in particular, have become primary targets [1], [2]. Distributed denial of service (DDoS), SQL injection (SQLi), script injection (XSS), and brute force (brute force) are prominent among these attacks. Despite their static controls, traditional security solutions are inadequate in the face of evolving attack techniques [3], [4]; attacks occurring at the web application layer, in particular, lead to serious data breaches and service disruptions [5].

Most existing intrusion detection systems (IDS) address the problem only as a binary classification (normal/attack) or focus on a single attack type. Furthermore, many studies rely on simulation-based or incompletely labeled datasets that do not reflect the real attack environment. In contrast, the Edge-IIoTset dataset provides multi-layered and realistic IoT traffic, enabling more reliable model development.

This study proposes an explainable, multi-class IDS model that focuses on four critical web-based attacks: DDoS, SQLi, XSS, and Brute Force. Machine learning algorithms were applied to reconstruct the dataset, and the highest performance was achieved with the Random Forest model (macro F1 = 0.94). Furthermore, SHAP-based explainability analysis revealed the protocol-level characteristics that guide the model's decisions. This focused and interpretable approach fills a significant gap in the literature by providing a practical IDS solution tailored to IoT environments.

II. Related Work

The literature on the four attack types focused on in this study (DDoS, SQL Injection, XSS, and Brute Force) can be generally examined under three headings: (i) binary classification-based intrusion detection systems, (ii) multi-class classification approaches, and (iii) web-based attack detection. Most existing studies either only distinguish normal/attack or focus on a single attack type. The evolving threat landscape has increased the need for explainable, multi-class models that can distinguish attack types.

A. Binary Classification-Based IDS

First-generation intrusion detection systems (IDS) classified network traffic solely as "normal" or "attack." While these methods offer advantages such as low computational cost and fast decision-making, they have critical limitations such as their inability to distinguish attack types and their inability to effectively handle class imbalance [6], [7]. For example, a study using classical models such as random forest (RF) and linear discriminant analysis (LDA) after dimensionality reduction with principal component analysis (PCA) achieved over 99% accuracy on CICIDS2017. However, rare attack types such as SQL injection were classified with high error rates, necessitating the need for balancing methods such as UDBB [6]. Newer hybrid methods (e.g., Transformer-CNN, Autoencoder-CNN) have achieved over 99.9% accuracy [7], [8]. However, these binary approaches fall short of capturing the diversity of low-frequency classes.

B. Multi-Classification-Based IDS

With the diversification of attack vectors in IoT environments, binary IDS structures that only detect anomalies are insufficient for comprehensive threat analysis. Therefore, interest in multi-class approaches that can distinguish different attack types has increased. For example, Trifa and Abdullah succeeded in classifying 15 different attack types with 99% accuracy on the IoT-23 dataset with an ANN-based model. However, they emphasized that classical balancing methods such as SMOTE did not significantly improve accuracy on rare classes [9]. In [10], the CSE-CIC-IDS2018 dataset was

used to evaluate CNN, Random Forest, LGBM, and their hybrid combinations. The models were tested on Normal, Bot, Brute Force, XSS, DDoS, and SQL Injection attacks. The hybrid approaches, especially CNN + Random Forest and LGBM + Random Forest, achieved the highest performance with an F1-score of 0.98. Similarly, the RF algorithm has been reported to demonstrate higher accuracy and robustness compared to methods such as support vector machines (SVM) and Naive Bayes [11]. Recently, hybrid methods have gained prominence: CNN-LSTM, CNN+XGBoost, and CNN+RF-based models achieved over 99% accuracy on the CICIDS and CSE-CIC-IDS2018 datasets, particularly enhanced by feature selection and balancing steps [12], [13]. These findings demonstrate that multi-class IDS approaches not only increase accuracy but also real-world applicability.

C. Detection of Web-Based Attacks

The web application layer is one of the most frequently targeted components of IoT networks, capable of inflicting the most devastating damage. DDoS attacks are known to cause significant resource consumption and service disruptions, especially in SDN-based environments [14]. SQL injection attacks are annually included in the OWASP Top 10 list, and studies using SVM and LSTM-based methods have reported F1 scores above 99% [15]. In a study [16], a two-layer LSTM-based Web Application Firewall (WAF) model was proposed to enhance intrusion detection. The architecture consisted of two sequential detection layers trained separately for specific attack types. The first layer was dedicated to DDoS detection and achieved an accuracy of 97.57% , while the second layer focused on SQL injection and XSS detection, reaching an accuracy of 89.34% . By tailoring each layer to a distinct attack category, the model aimed to improve prediction accuracy and provide a more reliable defense mechanism against diverse web-based threats.

In a study [17], a multi-layer LSTM model with self-attention achieved detection accuracies of 94.06% for Brute Force, 99.95% for DDoS, and 93.57% for Web attacks, with corresponding F1 scores of 0.89, 1.00, and 0.44. Another study [18] evaluated an LSTM-based hybrid RNN–CNN model on the CSE-CICIDS2018 dataset, reaching an accuracy of 98.15% for DDoS attacks. XSS attacks remain a prevalent threat, and character vectors and NLP-based approaches have been shown to yield successful results [19], [20]. Brute force attacks target authentication systems, and studies on the MQTT protocol in the IoT context have achieved success rates above 99% with RF and LSTM algorithms , [21], [22].

While effective methods have been developed for each attack type, the literature remains limited in providing a multi-class, explainable framework that simultaneously classifies DDoS, SQL injection, XSS, and brute force attacks. In this context, our study aims to fill a significant gap in the literature by developing an explainable and operationally applicable framework that focuses on four critical web-based attack types.

III. Methodology

In this study, a multi-class intrusion detection system (IDS) was developed to detect four web-based attacks commonly encountered in IoT environments (DDoS, SQL Injection, XSS, and Brute Force). Focusing on these attack types was chosen to increase both the interpretability and predictive accuracy of the model. It is known in the literature that comprehensive models that simultaneously address all attack types face various challenges in terms of class imbalance and explainability. Therefore, the proposed approach offers a more focused, scenario-based, and realistic framework. The model's training and testing processes were conducted on the Edge-IIoTset dataset, specifically designed for IoT and IIoT environments.

The ML-Edge-IIoTset is a comprehensive dataset containing 63 features and 157,800 samples. The dataset includes both network layer (IP, TCP, UDP) and application layer (HTTP, MQTT) information, enabling simultaneous analysis of protocol behavior at different levels. The dataset contains a total of 15 classes; These include attacks such as normal traffic, DDoS, SQL Injection, XSS, Ransomware, Man-in-the-Middle, Backdoor, and Port Scanning. In this study, only four critical attack types were selected and a five-class problem was defined along with normal traffic. Thus, the model was restructured to focus on distinguishing attack types in a realistic scenario.

In the preprocessing process, DDoS subtypes (TCP, UDP, HTTP, ICMP) were first combined into a single class due to their structural similarities. Off-target attacks were removed from the dataset, leaving only the four selected attack types and normal traffic. At this stage, repetitive or meaningless columns were cleaned, and some columns were converted to numerical format. To ensure efficient data processing by machine learning algorithms, categorical variables were represented using One-Hot Encoding, and any missing values that might occur during this process were filled with zeros to maintain integrity. The class imbalance observed in the dataset was compensated for using the SMOTE method to ensure that classes with few examples, such as SQL Injection and Brute Force attacks, were not overlooked during the training process. These steps resulted in a dataset that is both suitable for distinguishing attack types and provides fair representation across classes.

Random Forest-based importance scores were used during the feature selection phase to increase the model's efficiency and ensure it focused only on significant variables during the training process. This method was chosen because it provides high classification accuracy and allows it to measure the relative contribution of features in an embedded manner. The analysis revealed that the 20 most significant features out of 63 were identified. TCP port information, ACK and SEQ flags, HTTP version information, and some flags specific to the MQTT protocol were particularly prominent. The selected features were scaled between 0 and 1 using the Min-Max normalization method, preventing variables with different scales from dominating the model. Finally, the data was split into 80% training and 20% test, making it ready for the modeling

process.

The dataset generated by all these steps has been optimized to both distinguish the targeted attack types with high accuracy and minimize class imbalance. Thus, the proposed method provides the necessary infrastructure for developing an IDS model with a focused classification structure, high interpretability, and direct application to realistic IoT scenarios.

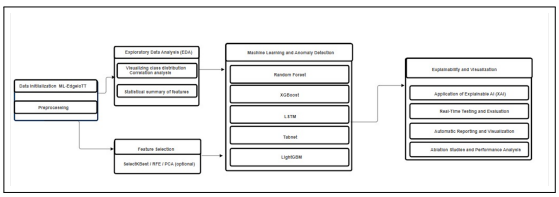


Fig. 1. Flowchart of the proposed method

To summarize the overall workflow, Figure 1 illustrates the proposed intrusion detection pipeline. Starting from data initialization with the ML-EdgeIIoTset dataset, the process continues with preprocessing, exploratory data analysis, and feature selection. Machine learning and deep learning models (Random Forest, XGBoost, LSTM, TabNet, LightGBM) were trained and compared. Finally, explainability, real-time performance testing, automated reporting, and ablation studies were carried out to ensure transparency and operational applicability.

IV. Classification Framework

In order to evaluate the intrusion detection performance, five different classifier models were trained, three of which are classical machine learning (Random Forest, XGBoost, LightGBM) and two of which are deep learning based (LSTM, TabNet). The model selection was made to evaluate the flexibility and generalizability of the system with different data structures and learning strategies. In this section, the general strategy of each model is summarized; then, the outputs of the best-performing Random Forest model are detailed.

A. Overall Evaluation of Model Performances

In this study, five classifier models representing different learning paradigms were tested: Random Forest, XGBoost, LightGBM, LSTM and TabNet. Each model was used to test its robustness against the multidimensional and imbalanced structure of the Edge-IIoT dataset and to analyze the discrimination between various attack types.

XGBoost is a powerful decision tree ensemble algorithm based on boosted learning. Hyperparameter optimization was performed with RandomizedSearchCV; 95.2% accuracy and 93.8% F1 score were obtained by obtaining the appropriate parameter combination. The model showed high success especially in Normal (100% F1), DDoS (97% F1) and SQL Injection (87% F1) classes. However, some confusion was observed between SQL Injection and Password attacks, which revealed the discrimination difficulties due to common traffic characteristics.

LightGBM is a classifier with a similar boosted tree structure. By optimizing the hyperparameters of the model, satisfactory performance was achieved with 95% accuracy and 93% macro F1 score. When the confusion matrix was examined, while high accuracy rates were observed in Normal and DDoS traffic, some overlapping predictions were made between the Password and SQL Injection classes. This situation indicates situations where the model's discrimination power is limited due to the behavioral similarities of the classes.

The LSTM model has the potential to analyze temporal patterns in network traffic thanks to its structure specific to time series data. The model's accuracy rate was 90.6%, and the macro F1 score was around 84%. Successful results were produced in classes with a strong number of examples such as DDoS (94% F1), XSS (88% F1) and Normal (100% F1); however, performance decreased in complex and less representative classes such as Password (62% F1) and SQL Injection (78% F1). This situation shows that deep models may be more vulnerable to class imbalance.

TabNet was chosen as a neural network architecture that can work with the attention mechanism and is suitable for tabular data structure. Hyperparameter optimization was performed with Optuna library, the model achieved 91.4% accuracy and 85.7% macro F1 score. Although high accuracy rates were observed in normal, DDoS and XSS traffic, a low success rate of 49.5% was observed in the Password class; this class is often confused with SQL Injection and DDoS. The behavior of the model revealed that it had difficulty distinguishing between classes with low sample numbers and semantic overlap.

The results obtained from these four models show that the tendency to make errors increases, especially in low-frequency and similar attack types. The Random Forest model, which showed the highest success among the five models and offered an advantage in terms of interpretability, was analyzed in more detail in the final stage of the study.

TABLE I
Overall Performance comparison of classifier models

Model	Accuracy (%)	Macro F1 (%)
Random Forest	96.0	94.4
XGBoost	95.2	93.8
LightGBM	95.0	93.0
LSTM	90.6	84.0
TabNet	91.4	85.7

The comparative performances of the five evaluated models are summarized in Table 1. Random Forest achieved the highest performance with 96% accuracy and a macro F1 score of 94.4%, making it the most suitable model for the proposed IDS framework. XGBoost and LightGBM also showed strong results, with accuracies above 95% and macro F1 scores above 93%. Deep learning-based models, LSTM and TabNet, provided competitive but relatively lower results, particularly in minority classes, due to sensitivity to class imbalance. These results confirm that while ensemble-based tree models are more stable across heterogeneous traffic, deep models may

require larger and more balanced datasets to achieve similar robustness.

B. Random Forest

In this study, the Random Forest (RF) model is used for multi-class attack detection. The main reasons for choosing this model are its ability to achieve high accuracy rates, its robustness against class imbalance, and its ability to internally evaluate the importance of features. Before model training, label data was converted to numerical form with LabelEncoder and SMOTE method was applied to the training data to reduce class imbalance. In this way, minority classes (e.g. SQL Injection, XSS, Brute Force) were better represented by the model. Random Forest model was trained with hyperparameters n estimator = 100 and random case = 42 and evaluated on test data. The accuracy rate of the model was measured as 96%. Precision, recall, and F1 score metrics calculated for each of the five classes are given below: While high success was achieved especially in Normal and XSS classes, balanced and acceptable results were also obtained in classes with relatively fewer examples, such as SQL Injection and Password. This performance shows the success of resampling and correct feature selection with SMOTE. The success metrics of the model after training and testing are visualized with the classification report and confusion matrix. In this study, Random Forest was preferred as the final model because it gave the highest and most balanced results compared to other models.

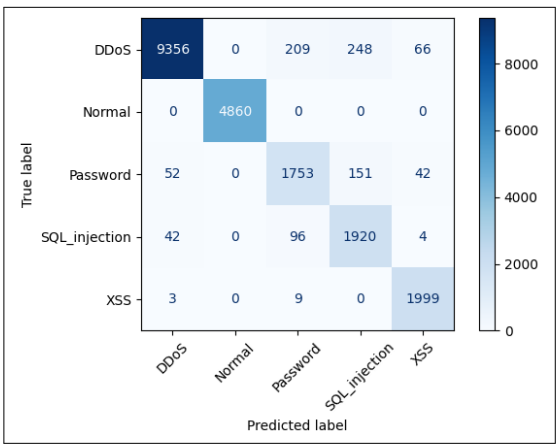


Fig. 2. Confusion matrix of random forest

In Figure 2, the confusion matrix shows how accurately the random forest model classifies DDoS, Normal, Password, SQL injection, and XSS attacks. The model achieved 100 percent success in the "Normal" class and correctly classified 4860 samples. Secondly, it correctly predicted the vast majority of "DDoS" attacks with 9356 samples, but confused them slightly with 209 samples as Password, 248 samples as SQL injection, and 66 samples as XSS. However, when looking at the Password and SQL injection classes, more confusion was observed. In SQL injection attacks, 96 samples were classified as Password. In Password attacks, 151 samples were classified as SQL injection and 42 samples as XSS. This reveals the possibility that attacks have common characteristics in terms of traffic.

V. Model Evaluation and Applied Analysis

The success of the model is not limited to the accuracy rate; explainability, system resource usage, automatic reporting infrastructure and understanding the contribution of the model's components are also of great importance. In this direction, the developed system was evaluated under four headings: (i) explainable artificial intelligence (XAI), (ii) latency and memory usage, (iii) automatic reporting, (iv) component effect with ablation analysis.

A. Explaining Decision Processes with Explainable AI (XAI)

SHAP (SHapley Additive Explanations) method was applied in order to make the decisions of the intrusion detection system interpretable. SHAP was analyzed especially on the Random Forest model and the effect of the features on the decision was calculated for each example.

Two basic criteria stood out within the scope of this analysis:

Fidelity: The level of overlap of SHAP explanations with model decisions was calculated as 0.96, which shows the reliability of the explanations.

Sparsity: On average, 19.6 features played an active role in the decision process, which shows that the model is not dependent on too many features at the same time.

Additionally, a SHAP summary plot was created to visualize the global impact on all samples (See Figure 3). In this plot, the average SHAP values of the most influential features for samples belonging to different classes are shown in color. In particular, it was observed that protocol-specific variables such as http.request.version_0_0, mqtt.conack.flags_0_0 and tcp.dstport had a discriminatory power between classes on model decisions.

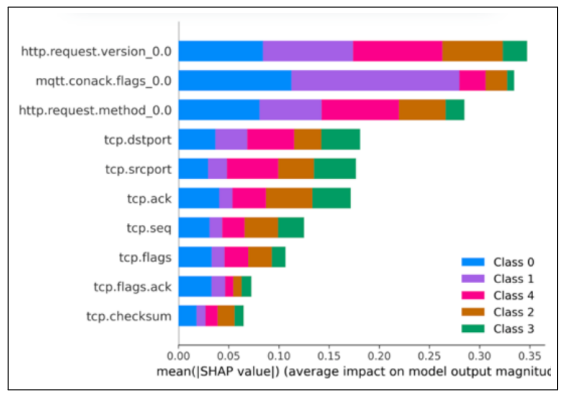


Fig. 3. SHAP values based on the classes

B. Latency and Memory Usage Measurement

In order to evaluate the potential usability of the developed system, the average latency and memory usage were measured on 100 samples. The average prediction time of the Random Forest model for each sample was determined as 0.0865 ms and the average memory consumption was determined as 12.08 MB. These results show that the model can be used in resource-constrained environments such as edge devices.

In the tests conducted with synthetic data, it was observed that the model successfully learned structures similar to real data. The average minimum sample distance was calculated as 1.23, and the correlation difference between the features was calculated as 0.1844, which showed that the synthetic data structurally represented the real data [23].

C. Automated Reporting of Classification Results

A Python-based automatic reporting system was developed to report the classification results in a regular, reproducible and understandable manner. The reporting workflow consists of the following steps:

Calculating the basic metrics with the compute_metrics function,

Converting these metrics to a stylized HTML table with the make_report_html function,

Saving them in an automatic folder structure (reports/auto/YYYY-MM-DD.html) based on daily dates.

In the reported examples, the Random Forest model achieved 96% accuracy and 94% macro F1 score.

D. F1-Based Ablation and False Positive Investigation on the Random Forest Model

The effect of the SMOTE method was evaluated with ablation analysis to understand the contribution level of the model components. The F1 score of the model trained without SMOTE was measured as 0.9443, while the full model was measured as 0.9387. The p value obtained in the statistical analysis performed with the McNemar test was $= 1.46 \times 10^{-8}$, indicating that the difference was significant.

In addition, SHAP-based analyses were performed on false positive examples and it was observed that features such as tcp.srcport, tcp.ack, mqtt.conack.flags played a dominant role in the decision process. This analysis revealed that the model successfully interpreted low-level protocol patterns, but tended to classify some normal examples as attacks.

VI. Conclusion and Future Work

This study presents a multi-class attack detection system that can accurately detect four basic web application layer attacks in IoT network traffic — DDoS, SQL Injection, XSS, and Brute Force. Addressing these attacks under a single classifier eliminates the need to develop separate binary models for each attack and increases the efficiency of the system in terms of distribution and resource usage. In addition, this integrated structure enables learning common malicious traffic patterns among different attack types, producing outputs that quickly and directly determine the attack type for real-time security measures.

The seven-layer ML-Edge-IIoTset dataset, which simulates a realistic and heterogeneous environment, allowed the models to be tested in complex scenarios. Many preprocessing steps such as label merging, missing value management, one-hot coding, and Random Forest-based feature selection were applied during the model development process. The class imbalance problem was addressed with the SMOTE method and fairness in representation was achieved among attack types.

Five models representing different learning paradigms (Random Forest, LightGBM, XGBoost, LSTM and TabNet) were comparatively evaluated. Among these, the Random Forest model showed the highest success with a macro F1 score of 94.4%, while it was the most resource-efficient model with a latency of 0.0865 ms/sample and 12.08 MB memory usage. While LightGBM and XGBoost produced similarly strong results, LSTM successfully captured especially time-dependent patterns, and TabNet offered a balanced structure in terms of interpretability and performance.

The decision processes of the model were explained with the SHAP (SHapley Additive Explanations) method, and it was shown that a strong agreement was achieved between model decisions and explanations with a fidelity value of 96%. It was observed that protocol-specific variables, especially tcp.dstport and mqtt.conack.flags, played a decisive role in distinguishing between classes. The SHAP analysis performed on false positive samples revealed that there was confusion due to the overlap of some patterns with other attack types, especially in the Brute Force class.

In the ablation analysis performed to examine the effect of the SMOTE application, a remarkable result was obtained: The model without SMOTE gave better results with a macro F1 score of 94.43% and this difference was found to be statistically significant with the McNemar test ($p = 1.46 \times 10^{-8}$). This finding shows that although it provides efficiency for minority classes, oversampling methods can blur the decision boundaries.

In addition, the developed system provides reproducibility and traceability by automatically reporting the results obtained in each run as timestamped files in HTML format. This structure provides an important infrastructure especially for integration into the production environment or continuous monitoring processes.

As a result, this study not only suggests an attack detection framework with a high success rate; it also offers a strong structure in terms of interpretability, efficiency and operational applicability. The developed system meets the basic requirements of modern intrusion detection systems operating in today's layered and complex IoT/web environments and creates a solid foundation for sustainable security monitoring processes.

Although the findings are promising, some limitations of the study should be considered. First of all, although the ML-Edge-IIoTset dataset reflects realistic traffic patterns, it was

obtained from a simulation-based environment. Furthermore, the research was limited to only four attack types; although this increases the interpretability and performance of the model, it would be useful to address a wider range of attacks such as Ransomware or Man-in-the-Middle (MITM) in the future for general validity. Finally, although SHAP-based explainability analyses provide meaningful insights, it may be necessary to turn to more efficient methods due to computational costs in big data scenarios.

REFERENCES

[1] Symantec, "Internet security threat report," Symantec Corporation, Vol. 23, 2018, accessed: 2025-08-20. [Online]. Available: https://www.symantec.com/security-center/threat-report

[2] K. DeMedeiros, A. Hendawi, and M. Alvarez, "A survey of ai-based anomaly detection in iot and sensor networks," *Sensors*, vol. 23, no. 3, p. 1352, 2023.

[3] R. Ejjami *et al.*, "Enhancing cybersecurity through artificial intelligence: Techniques, applications, and future perspectives," *Journal of Next-Generation Research 5.0*, 2024.

[4] M. A. I. Mallick and R. Nath, "Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments," *World Scientific News*, vol. 190, no. 1, pp. 1–69, 2024.

[5] W. Al-Kahla, A. S. Shatnawi, and E. Taqieddin, "A taxonomy of web security vulnerabilities," in *2021 12th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2021, pp. 424–429.

[6] R. Abdulhammed, M. Faezipour, H. Musafer, and A. Abuzneid, "Efficient network intrusion detection using pca-based dimensionality reduction of features," in *2019 International symposium on networks, computers and communications (ISNCC)*. IEEE, 2019, pp. 1–6.

[7] H. Kamal and M. Mashaly, "Advanced hybrid transformer-cnn deep learning model for effective intrusion detection systems with class imbalance mitigation using resampling techniques," *Future Internet*, vol. 16, no. 12, p. 481, 2024.

[8] J. Zhang, L. Pan, Q.-L. Han, C. Chen, S. Wen, and Y. Xiang, "Deep learning based attack detection for cyber-physical system cybersecurity: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 377–391, 2021.

[9] M. B. Musthafa, S. Huda, Y. Kodera, M. A. Ali, S. Araki, J. Mwaura, and Y. Nogami, "Optimizing iot intrusion detection using balanced class distribution, feature selection, and ensemble machine learning techniques," *Sensors*, vol. 24, no. 13, p. 4293, 2024.

[10] M. K. Pehlivanoğlu, R. Atay, and D. E. Odabaş, "İki seviyeli hibrit makine öğrenmesi yöntemi ile saldırı tespiti," *Gazi Mühendislik Bilimleri Dergisi*, vol. 5, no. 3, pp. 258–272, 2019.

[11] R. Liu, Y. Feng, and K. Sakurai, "An approach to multi-class intrusion detection based on feature subspaces and weighted fusion," in *2024 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2024, pp. 139–146.

[12] N.-A. Stoian, "Machine learning for anomaly detection in iot networks: Malware analysis on the iot-23 data set," B.S. thesis, University of Twente, 2020.

[13] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in iot security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[14] R. Alguliyev and R. Shikhaliyev, "Computer networks cybersecurity monitoring based on cnn-lstm model," in *2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, 2024, pp. 1–6.

[15] A. F. Al-zubidi, A. K. Farhan, and S. M. Towfek, "Predicting dos and ddos attacks in network security scenarios using a hybrid deep learning model," *Journal of Intelligent Systems*, vol. 33, no. 1, p. 20230195, 2024.

[16] B. R. Dawadi, B. Adhikari, and D. K. Srivastava, "Deep learning technique-enabled web application firewall for the detection of web attacks," *Sensors*, vol. 23, no. 4, p. 2073, 2023.

[17] S. Volkov and I. Kurochkin, "Network attacks classification using long short-term memory based neural networks in software-defined networks," *Procedia Computer Science*, vol. 178, pp. 394–403, 2020.

[18] A. A. Hagar and B. W. Gawali, "Deep learning for improving attack detection system using cse-cicids2018," *NeuroQuantology*, vol. 20, no. 7, pp. 3064–3074, 2022.

[19] A. K. Balyan, S. Ahuja, U. K. Lilhore, S. K. Sharma, P. Manoharan, A. D. Algarni, H. Elmannai, and K. Raahemifar, "A hybrid intrusion detection model using ega-pso and improved random forest method," *Sensors*, vol. 22, no. 16, p. 5986, 2022.

[20] D. Tang, Y. Yan, S. Zhang, J. Chen, and Z. Qin, "Performance and features: Mitigating the low-rate tcp-targeted dos attack via sdn," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 428–444, 2021.

[21] D. Mehta, H. Suhagiya, H. Gandhi, M. Jha, P. Kanani, and A. Kore, "Sqliml: A comprehensive analysis for sql injection detection using multiple supervised and unsupervised learning schemes," *SN Computer Science*, vol. 4, no. 3, p. 281, 2023.

[22] A. F. Otoom, E. E. Abdallah, *et al.*, "Deep learning for accurate detection of brute force attacks on iot networks," *Procedia Computer Science*, vol. 220, pp. 291–298, 2023.

[23] Y. Lu, M. Shen, H. Wang, X. Wang, C. van Rechem, T. Fu, and W. Wei, "Machine learning for synthetic data generation: a review," *arXiv preprint arXiv:2302.04062*, 2023.

Comparative Security and Performance Analysis of Session-Based and JWT-Based Web Session Mechanisms

Abdaal khan khattak
CyberMACS
SRH Heidelberg University of Applied Sciences
Berlin, Germany
0009-0002-4844-7402

Prof. Vladimir Stantchev
School of Technology and Architecture
SRH Heidelberg University of Applied Sciences
Berlin, Germany
vladimir.stantchev@srh.de

Prof. Reiner Creutzburg
School of Technology and Architecture
SRH Heidelberg University of Applied Sciences
Berlin, Germany
0000-0001-7522-5990

Prof. Hasan Dağ
Management Information Systems
Kadir Has University
Istanbul, Turkiye
0000-0001-6252-1870

Muhammad Abubakar Bajwa
Engel & Voelkers Technology GmbH
Engel & Voelkers GmbH
Hamburg, Germany
muhammad.abubakar@engelvoelkers.com

Abstract*—Modern web sites run on plain HyperText Transfer Protocol (HTTP). Two common approaches are to achieve maintained user-state: the traditional session that lives on the server and the newer JSON Web Token, or JWT, that lives only in the browser and gets checked by a signature. The big question here is: which one is more secure, and which one has better resource utilization? To answer this, two Node.js apps were built in a local environment. One app used express-session along with Redis, an in-memory database. The other used jsonwebtoken library with an HS256 signing algorithm that uses the secret "thesis-secret-123". To test security, user request was sent using Postman to the apps through OWASP ZAP's proxy. When ZAP ran its scans, it reported 6 different alert types, about 42 and 44 for session-based and JWT-based web session systems respectively. A manual check with Burp Suite was used to manipulate the user requests. Tempering the connect.sid cookie to something random gave a 401 error. Same with JWT token, the JWT-based server responded with a 401 error. That tells the server-side storage can spot an invalid cookie, but the token can be peeked at by anyone who cares to decode it. Performance was measured with JMeter, simulating 1000 users sending requests to each server in total duration of $20\,\mathrm{s}$. The JWT version answered in about $279\,\mathrm{ms}$ on average, while the session version took roughly $7870\,\mathrm{ms}$. Throughput followed a different pattern: about 3.39 requests per second for each server app. Grafana graphs made the gap look clear. The system stats showed JWT using a little less CPU ($0.127\,\%$ vs $0.128\,\%$) and a bit less memory ($59.3\,\mathrm{MB}$ vs $76.1\,\mathrm{MB}$). These insights underscores that the sessions give you tight control on the server, which can stop some attacks, but they add a little lag. JWTs run fast, but it requires a robust secret management.

***Index Terms*—session management, JWT, authentication, security, performance, OWASP ZAP, JMeter, Grafana, Prometheus, Node.js, Express.js, Burp Suite, Redis**

I. Introduction

The web application's backbone of modern digital interactions, from e-commerce platforms to banking systems, all rely on the HyperText Transfer Protocol (HTTP) to facilitate communication between clients and servers. When it comes to relaying messages back and forth, however, the stateless nature of HTTP—that is, each request made and answered is independent—means that there must be very robust session management mechanisms to ensure that that all-important user authentication state is maintained when the user interacts with the system. In terms of popular approaches used to solve this problem, two modes of authentication dominate public discourse on the web: session-based authentication and JWT-based authentication. Both have distinct pros and cons. This thesis compares the performance and security of two methods of user authentication: session-based and JSON Web Token (JWT)-based. The research involves building two prototypes using Node.js and the Express framework. The first prototype uses traditional session management with express-session and Redis, an in-memroy database, as the session store. The second prototype uses JSON (JavaScript Object Notation) Web Tokens and the jsonwebtoken library to sign and verify tokens. The two prototypes were put through both security and performance testing.

A. Subject Relevance

Safeguarding the web app's sessions ensures users can trust the app and ensures the app can scale. Second, session-based authentication is good at doing that because it ensures only the right people can access the app. Third, the web app developers who build such authentication systems need to understand where their systems can be vulnerable (as in sending unprotected messages over the internet), and where the session or JWT based authentication systems be implemented.

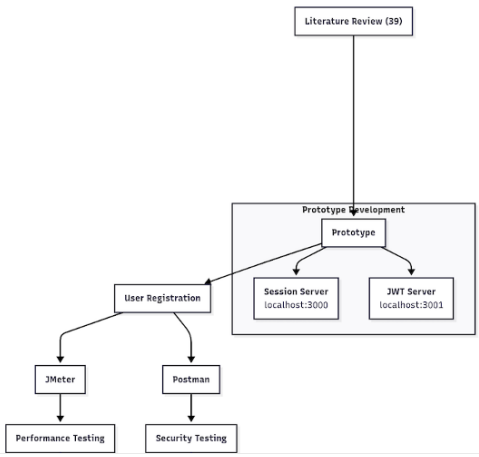


Fig. 1. Research Workflow

B. Research Gap

While session management is well covered in the literature, there is a gap in direct comparisons between session-based and JWT-based session management systems in empirical studies. They propose that the payload should be encrypted to mitigate those dangers, but they put forward no real-world tests for this assertion. [2] tackles the much larger question of performance in distributed systems. He also seems to think that this is a much larger question of theoretical framework issues, rather than one that could be solved with a proof of concept. [18] and [35] have a lot to say about session hijacking and how to make cookies safer. But their proposed methods are from a time before JWTs were common.

C. Objectives and Goals

This thesis primarily aims to empirically test and compare, from security and performance perspective as shown in Fig. 1, the two ways of session management systems across stateless servers. The first way, session-based, has been in use for a long time. The second way, using JSON Web Tokens (JWTs), has become popular in recent years. I have created two prototypes using Node.js and Express web framework: a server that manages client sessions with Redis, and the other uses JWT for session management. Using OWASP ZAP and Burp Suite to evaluate security and find vulnerabilities. Evaluating performance with JMeter alongside Prometheus/Grafana metrics. Examining pros and cons to guide the development of safe and streamlined authentication. Our objectives are to produce developer-friendly insights that they can act upon. This is nothing less than a hope to achieve session management ideal place.

D. Research Questions

This research investigates the subsequent questions: How both session management systems compare regarding security vulnerabilities, such as token tampering and CSRF. Authentication via sessions and authentication via JWTs exhibit different behaviours due to their differing characteristics? While a session's lifespan will typically extend until the user explicitly logs out or until a session-expiration event occurs, a typical JWT will remain valid until its expiration time is reached. However, a session can be kept alive through the use of a refresh token (e.g., a cookie). A JWT that has not expired but is somehow compromised can cause serious issues, but a compromised session also causes serious problems. So, from the point of view of using sessions versus using JWTs, both have their pros and cons when it comes to security and expiration. How do session-based and JWT-based systems perform (in terms of latency and throughput) under a load of 1,000 users? These trade-offs mean real things in the world for web application design. What are the practical implications on the functionality of a web application if implemented either?

E. Scope & Limitations

This study's scope is confined to a self-hosted environment local environemnt (macOS, Redis, Node.js) with no-cost tools with free-tier plans (Postman, OWASP ZAP, Burp Suite Community, JMeter, Prometheus, Grafana). The prototypes (server-session.js, server-jwt.js) employ overly basic configurations (e.g., a weak secret) to stand in for actual, hazardous-to-your-application configurations that you might find in the real world. Limitations include:

Localhost Limitations: Absence of network latency can lead to an underestimation of the performance disparities in production environments. Simplified Secrets: Using a weak secret makes JWT vulnerabilities look worse than they are, implying that they are far more dangerous than they actually are and that they apply to much more than just JWTs. Scale: Testing with 1,000 users may not fully capture high-volume scenarios. Limitations of the Tools: Tools at the free-tier may not possess the many advanced features found in their paid-tier plans or enterprise-level counterparts.

F. Structure/Overview

This dissertation has the following structure:

Review of the literature: examines important works that pertain to the management of user sessions. It calls out weaknesses that have been identified in these works and looks to see if there are any performance trade-offs that might account for the identified session management vulnerabilities. Methodology: details the prototype development, security testing and performance testing. Finding: shows results from security (ZAP reports, burp suite tests) and performance (latency, throughput) studies. Discussion: analyzes trade-offs and implications for web application design. Conclusion: key findings and future research directions are summarized. References: cited works relevant to the traditional session and jwt session mechanisms are listed.

II. Literature Review

Web session management serves as a fundamental pillar in the architecture of secure web applications, ensuring that

user interactions remain authenticated, authorized, and stateful across multiple requests in inherently stateless protocols like HTTP. Effective session management mechanisms are critical for maintaining user privacy, preventing unauthorized access, and mitigating a wide array of cyber threats, including session hijacking, fixation, and replay attacks. As web applications evolve to support distributed systems, mobile integrations, and high-traffic environments, the choice of session management technique has profound implications for both security and performance. Traditionally, two primary paradigms dominate this domain: server-side session management, which relies on storing session data centrally on the server and referencing it via identifiers (often cookies), and stateless token-based approaches, exemplified by JSON Web Tokens (JWTs), which embed session information directly into client-side tokens secured through cryptography.

Session management in web applications has been changing. While it once relied on a traditional session-based approach, it is now moving toward stateless token-based methods that bear no resemblance to sessions—like JWTs—to meet the demands of large-scale distributed systems. Early sessions managing systems were forced to work around the limitations of HTTP, which was not designed with the notion of state in mind. They achieved this via cookies, for instance. Many servers and many users, each of whom the server must keep track of for only as long as the user is logged in, posed a somewhat a huge dilemma that was only solved by viably scaling up the server. This was managed by using key/value stores that maintained user sessions (like Redis, for example) and insecurely kept the tokens (or keys) that allowed the server to perform Cross-Site Request Forgery (CSRF)-unsafe operations on behalf of users.

Server-side session management has long been the standard in web development, where a unique session identifier (SID) is generated upon user authentication and stored in a cookie or URL parameter. The server maintains the associated session data in memory, databases, or caches, allowing for straightforward revocation, monitoring, and updates. This approach excels in environments requiring tight control over session lifecycle, such as invalidating sessions upon logout or detecting anomalies in user behavior [10]. However, it introduces scalability challenges, particularly in distributed architectures, as session data must be synchronized across servers, often necessitating shared storage solutions like Redis or databases, which can become bottlenecks under high load [2]. Moreover, vulnerabilities such as session fixation—where an attacker presets a session ID—or hijacking via intercepted cookies remain prevalent if secure transport (e.g., HTTPS) and cookie flags (e.g., HttpOnly, Secure) are not rigorously enforced [29].

In contrast, JWT-based session management represents a shift toward stateless, client-side paradigms, where session data is encoded into a compact, self-contained token comprising a header, payload, and signature. Introduced as a standard in RFC 7519, JWTs leverage cryptographic signing (e.g., HMAC-SHA256 or RS256) to ensure integrity and authenticity without requiring server-side storage for each session [33]. This enables seamless scalability in microservices, cloud-native applications, and cross-origin scenarios, as servers validate tokens independently without database queries [5] [4][12] [34]. Recent advancements highlight JWTs' efficacy in high-volume contexts, such as e-government systems and RESTful APIs, where they facilitate token rotation, refresh mechanisms, and integration with protocols like OAuth 2.0 and OpenID Connect [30] [14] [17] [8]. For instance, implementations using HMAC-SHA512 for signing have demonstrated robust resistance to tampering, while custom handlers enhance session persistence and security [22] [13].

Studies highlighted both the benefits and the vulnerabilities associated with using JSON Web Tokens (JWT). For instance, in [32] it has discussed that using JWT with something like a Quick Response (QR) code check-in could be a really fast and simple way of obtaining the type of login that a web application required, but they identified some serious security issues with the way in which base64-encoded payloads could be extracted by a malicious person who might steal a JWT. They advocated for encrypting both the payload and header of the JWT as a better practice and something that would certainly fall in line with what this thesis encompasses when it comes to addressing the security of session tokens. Token leakage through cross-site scripting (XSS) or insecure storage (e.g., localStorage vulnerable to JavaScript access) can lead to unauthorized use, as stolen tokens remain valid until expiration [9] [19] [23] [36]. Revocation poses a particular challenge in stateless designs, often requiring supplementary mechanisms like blacklists or short-lived tokens, which can partially negate scalability benefits [15] [24] [27]. Comparative studies underscore that while JWTs reduce server overhead and enhance interoperability in distributed environments, they demand meticulous key management and algorithm selection to avoid exploits like algorithm confusion attacks [3] [31] [11]. Server-side sessions, conversely, offer inherent revocation capabilities but may falter in cross-domain or mobile scenarios due to cookie restrictions and state synchronization overhead [16] [7]. Research also investigates and evaluates performance trade-offs. [2] adjusts session management in distributed systems using JWTs, reducing server load and preventing scalability issues associated with HTTP cookies. Still, he notes an increase to the client's exposure. [35]introduces a different take on session management, analysing the security of SSO and proposing reverse proxies as a way to reach new levels of integrity, efficiency, and performance. Finally, [10] discusses an often-overlooked problem with maintaining secure sessions—session fixation attacks. He warns that, to thwart such attacks, using random session identifiers is an absolute must. Recurrent are the vulnerability themes in session management. They highlight the need for cookie attributes to be secure. They inform the reader about secure ways to generate session IDs. While [32] propose encryption solutions for JWT vulnerabilities, they lack practical prototype-based testing. [2] advocate for Role-Based Access Control(RBAC) and Multi-Factor Authentication (MFA) but do not directly compare session and JWT systems. [2] focuses on theoretical frameworks for distributed systems, overlooking local imple-

mentations.

The importance of encryption and secure configurations is also emphasized in the literature. [32] address the base64 encoding vulnerability of JWT, suggesting that to truly secure it, one must be encrypting the payload and header, to protect user information from network thieves who would steal tokens. This study is particularly relevant because the JWT prototype used in this project used a weak secret, which was easily hacked, leading to some vulnerabilities that were found in the ZAP scans. Saving the best for last, [10] highlights session fixation in session-based systems, pointing out that the old standby of using random IDs is good, but is made even better by also having short expiration times.

Moreover, the literature stresses performance in applications with a high volume of transactions. [2] observed that JWTs serve to lighten the load on servers since they do not necessitate database queries. This makes JWTs a good choice for applications that must scale up. This contrasts somewhat with the JMeter results from this study, which showed that JWTs had lower latency than session identifiers. Still, both tokens were fast enough to keep up with what most people would consider real time.

The growing complexity of web ecosystems—encompassing IoT, single-page applications (SPAs), and hybrid mobile-web platforms—has amplified the need for empirical comparisons between these techniques. Vulnerabilities in session management continue to rank among the top risks in frameworks like OWASP Top 10, with broken authentication accounting for significant breaches [21] [25]. Emerging hybrid models, such as combining JWTs with server-side checks or using token splitting for enhanced leakage prevention, aim to bridge these gaps [24] [6] [26] [20]. Additionally, research by [21] highlights the importance of secure session handling in RESTful API contexts, further emphasizing the need for robust implementation strategies.

This review synthesizes recent literature to evaluate the web security implications of JWT versus server-side session management, drawing on over 39 or more high-quality studies to analyze strengths, weaknesses, attack vectors, and practical implementations. By addressing these dimensions, this paper aims to guide developers and researchers in selecting optimal strategies for secure, scalable web applications in an era of escalating cyber threats. To summarize, the empirical evidence in the literature makes it possible to understand reasonably well how session management works. But when we look at the literature's local, controlled, side-by-side comparisons of session management method performance and security, we find the local comparisons to be quite limited. This thesis localizes the analysis with two session management prototypes: one based on session IDs and the other based on JWT token. It prototypes both practically, contributing two new artifacts to the field, and tests both prototypes for security and performance to understand how local conditions affect the prototypes.

III. Methodology

This chapter systematically compares the security and performance of two session management mechanisms used in web applications: session-based and JWT-based. The comparison is done by developing two prototypes using Node.js and Express web framework. One prototype uses JWTs for authentication, the other uses session-based authentication. Both are run under the same conditions in a performance test. The source code of both prototypes are hosted on GitHub [1] and are made public for anyone to examine along with the configuration files for some of tools used for assessing the prototypes. Meanwhile, I am assessing the security of both prototypes by using the OWASP Risk Rating Methodology [28]. The tools I am using to do this are all open-source, which helps keep costs low, ensures accessibility, and makes it easier for others to replicate what I have done. And there's no way around it: keeping costs low and using tools that are accessible to everyone makes it possible for anyone to follow my lead.

A. Prototype Development

Two prototypes were created with Node.js (v22.16.0) and Express (v5.1.0), operating in a localhost environment (macOS Sequoia v16.6.1 with 16 GB RAM). The first prototype, accessible at http://localhost:3000, implements session-based authentication using the express-session (v1.18.2) middleware with Redis (v4.6.5) as the session store. The configuration includes a session secret "thesis-secret-123", a cookie with httpOnly: true, sameSite: 'strict', and a max age of 1 hour, throughout which the session is expected to be active. A weak secret is used intentionally to examine the effectiveness of each web session management system or its resilience to threats. The second prototype, accessible at http://localhost:3001, employs JWT-based session management system using the jsonwebtoken (v9.0.2) library, responsible for the creation and validation of JWTs. This prototype generates tokens that, like the first, are expected to be useful for about 1 hour, with the time to be reached validated via the /protected endpoint. The server that uses JWT handles requests (that require authentication) via the JWT protocol. The server validates the token in every request where authentication is required. The vulnerabilities of the server stem, to a great extent, from a weak encryption key that was used to encrypt the tokens. For our tests, we configure it to use a signing algorithm called HS256, which is Hash-Based Message Authentication Code (HMAC), a symmetric key encryption algorithm, used in combination with SHA-256, would produce a signed JWT that is essentially secure, so long as a secure (not weak) secret is used. The server, however, uses SHA-256 combined with the secret "thesis-secret-123". The user list for the authentication test was stored in a database that is backed by Redis.

B. Data Collection

1) Security Testing: Security testing was carried out using the OWASP ZAP (v2.16.1) and Burp Suite Community Edition (v2025.7.3) to find security holes in the prototypes as shown in Fig. 2. The work started with setting ZAP up as a proxy

(127.0.0.1:8080) to catch the HTTP requests that Postman (v11.58.4) was sending to the endpoints /register, /login, and /protected. ZAP then used its spider scan feature to crawl the application, map its structure, and take an inventory of the endpoints. Finally, ZAP performed an active scan to find security problems like CSRF, insecure cookie flags, and weak secret (key) configurations. We used the OWASP Risk Rating Methodology [28] to guide our assessment of the "hit" taken during the active scan. The methodology asks assessors to think about two dimensions when coming up with a risk rating. First, how likely is it that an average attacker could get through the defenses? (This is sometimes called the "ease of attack" or, alternatively, the "skill level" needed by an attacker.) Then, how serious would it be if the attack succeeded? (This is the "impact" dimension.) Scan reports, generated for the session-based server and for the JWT-based server, provided detailed alert counts sorted by risk (low, medium, high) and confidence levels. Burp Suite complemented ZAP by allowing us to do some manual tampering tests. For example, we intercepted a couple of requests and modified them in ways to see how the server would respond before and after our modifications. One thing we tried was to change the connect.sid cookie in http request header by re-running the connect.sid command after we had already made the cookie once. Another thing we tried was to alter the JWT in the Authorization header in ways to see what would happen.

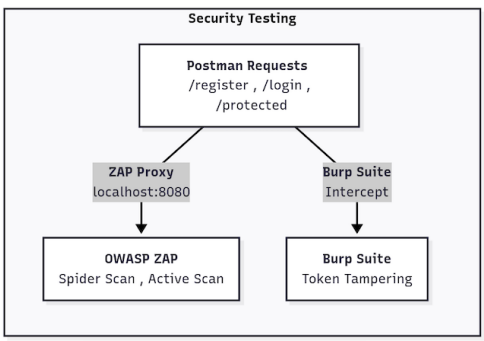


Fig. 2. Security Testing Workflow

2) Performance Testing: Performance testing occurred using Apache JMeter (v5.6.3) for both prototypes under simulated load to evaluate efficiency. The JMeter script configured thread groups for 1,000 concurrent users with a ramp-up period of 20 seconds to avoid sudden spikes that could skew results. Requests were sent to the endpoints `/register` for both servers while measuring not just response times but also throughput (requests per second), and resource utilization. The results were saved to `summary.csv` and `aggregate.csv`. In the prototypes, Prometheus was integrated in both prototypes using the library `prom-client` and configured to collect server-side metrics like `http_request_duration_seconds` which represents the latency in seconds and `http_requests_total` which represents the total request count. These metrics were made available at the `/metrics` endpoint of our REST API. To visualize and make sense of all these metrics, we configured dashboards in Grafana to use Prometheus as their data source as shown in Fig. 3. They then directly rendered and made some quite nice visualizations of the metrics, plus others like average latency and overall throughput. Then, to see how well our service performed, we also had Grafana visualize what is essentially a rehash of the classic system monitor when one runs `top` or `htop` on a Linux machine: how much CPU we are using `node_process_cpu_seconds_total`, how much memory we are using `node_process_resident_memory_bytes`.

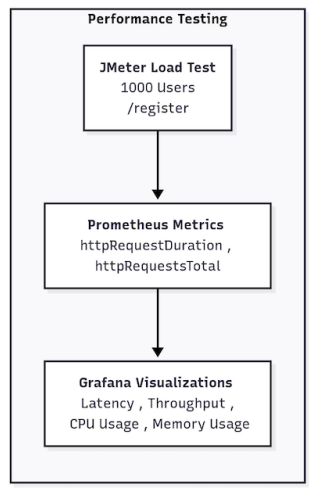


Fig. 3. Performance Testing Workflow

IV. Findings

This part shows the empirical results from tests on the security and performance of the two prototypes: session-based and JWT-based. The prototypes were comprehensively scanned and tested for vulnerabilities. Security testing was performed using OWASP ZAP and Burp Suite, while performance testing was done with JMeter, Prometheus, and Grafana, on up to 1,000 users. These tools allowed us to find several severe vulnerabilities, which are outlined here in the security findings section. No tool is perfect, though, and these two tools (especially Burp Suite) allow for a lot of user-defined manual testing as well.

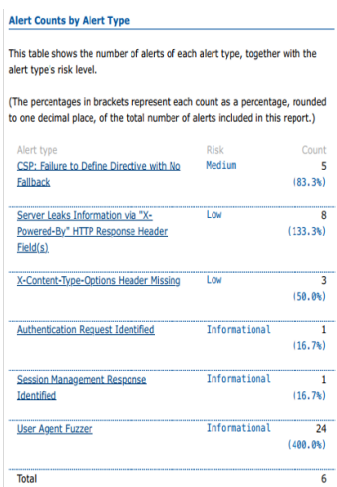
A. Security Findings

Vulnerabilities like CSRF, session hijacking, and token tampering were the focus of security testing, which used OWASP ZAP for automated scans and Burp Suite Community Edition for manual interception. Postman sent requests through ZAP's

proxy to capture traffic for endpoints (/register, /login, /protected, /logout). Alert evaluation was guided by the OWASP Risk Rating Methodology, which considers threat agents (who could possibly tamper with the system) and technical impacts (like the possibly undetectable exposure of was-it-secured-or-was-it-not secure data which could be base64 decoded).

1) Session-Based Server Security: The app runs a session-based server. It uses the express-session package and stores the sessions in Redis. The code signs the connect.sid cookie with a secret "thesis-secret-123". The idea is that the server checks the session each time, which should be solid. But at the same time, the weak secret may let someone forge cookies. Also, the headers aren't locked down – I didn't see any strict-transport-security or content-security-policy set. When we ran OWASP ZAP we got six different alert types as shown in Fig. 4. In total there were 42 alerts. 5 of them were in the medium/high risk bucket, mostly CSP failures. 11 were low/medium, like server leaks and missing headers. The remaining 26 were informational – things like authentication IDs being exposed or fuzzing results. The distribution tells most of the warnings are low-risk, which kind of matches what you'd expect from a default Express setup. Still, the presence of any CSP failures is worrisome because they can be used to inject malicious scripts. The high count (42) means the scanner hit the server a lot, sending many requests. So, we probably saw most of the easy-to-find problems. The missing directives and info leakage could be taken advantage of in a real production run. Tamper resistance was confirmed by manual tests with Burp Suite: when connect.sid was modified in intercepted requests to /protected, the server responded with 401 errors. Those errors validated that the session ID was being checked against a server-side Redis store before the request was authorized. In short, the checking mechanism works, even if the system is not completely secure.

2) JWT-Based Server Security: The JWT-based server runs on Express and uses the jsonwebtoken package. It signs everything with HS256, and it uses the same weak secret "thesis-secret-123". We store user information in Redis as shown in Fig 5, then hand out stateless tokens to the browser. It lets the app grow without a bunch of session files, but it also means the token format itself can leak stuff if we're not careful. When OWASP ZAP ran on the server, the tool spit out 7 different alert types, totaling about 44 alerts. Five of those were Medium-or-High risk, mostly CSP failures that could let a sneaky script slip in. Thirteen fell into Low-or-Medium; those were things like server leaks, missing security headers, and the token's timestamp showing up where it maybe shouldn't. The remaining 27 were informational – things like authentication IDs being exposed or fuzzing results were having informational-level risk about how authentication and session handling were identified by the scanner. Mostly, the default Express setup left a lot of low-level warnings hanging around, and the JWT code itself exposed timestamps. In short, the JWT-server got some real-world-usable flaws, even if most of the warnings look "low-risk". It probably needs to change the secret, add proper headers, and maybe tighten the CSP. Otherwise, a curious user could probably figure out a way to trick the system. Tamper resistance was confirmed by manual tests with Burp Suite in the case of JWT-based systems. Burp Suite tests demonstrated that JWTs can be tampered with. When testers modified the JWT payloads, it was returning 401 errors. The errors validated recomputing the HMAC-SHA256 hash of the encoded Header and Payload using the secret key and comparing it with the signature in the request, if they match, the token's integrity is confirmed. A mismatch indicates tampering. So this case too, the checking mechanism works, even if the system is not completely secure.



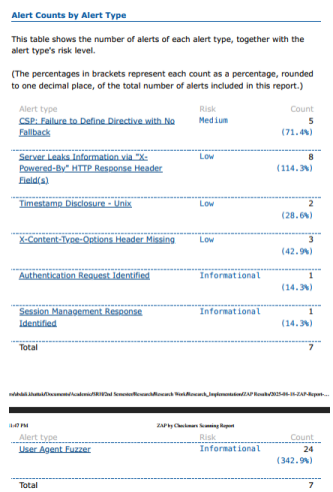
Alert Counts by Alert Type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
CSP: Failure to Define Directive with No Fallback	Medium	5 (83.3%)
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	8 (133.3%)
X-Content-Type-Options Header Missing	Low	3 (50.0%)
Authentication Request Identified	Informational	1 (16.7%)
Session Management Response Identified	Informational	1 (16.7%)
User Agent Fuzzer	Informational	24 (400.0%)
Total		6

Fig. 4. ZAP Scan Report of Session-based Server



Alert Counts by Alert Type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
CSP: Failure to Define Directive with No Fallback	Medium	5 (71.4%)
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	8 (114.3%)
Timestamp Disclosure - Unix	Low	2 (28.6%)
X-Content-Type-Options Header Missing	Low	3 (42.9%)
Authentication Request Identified	Informational	1 (14.3%)
Session Management Response Identified	Informational	1 (14.3%)
Total		7

Alert type	Risk	Count
User Agent Fuzzer	Informational	24 (342.9%)
Total		7

Fig. 5. ZAP Scan Report of JWT-based Server

B. Performance Findings

Testing conducted in JMeter simulated 1,000 users and pinpointed the superiority of the JWTs. On average, the latency for retrieving or verifying the ID token was 279ms (JWTs) compared to 7870ms (SessionIDs), and the requests per second for throughput were same for both web authentication systems (refer to Table 1). Additionally, a look at the Grafana dashboards (Fig. 6, Fig. 7, Fig. 8, Fig. 9) for the same tests showed the following averages for CPU and memory usage, which support the claims of better performance with JWTs as opposed to SessionIDs. These are reliable results for the performance of the two authentication methods in retrieving or verifying the ID token at the time of access. We rated the performance using good old four-point scales for latency, throughput, CPU, and memory, with low being better or higher performance in case of CPU usage, Memory Usage and Latency while high being better or higher performance in case of Throughput.

TABLE I
PERFORMANCE METRICS (1,000 USERS)

Metric	Session-Based	JWT-Based
Maximum Latency (ms)	7870	279
Maximum Throughput (req/s)	3.39	3.39
Maximum CPU Usage (%)	0.128	0.127
Maximum Memory Usage (MB)	76.1	59.3

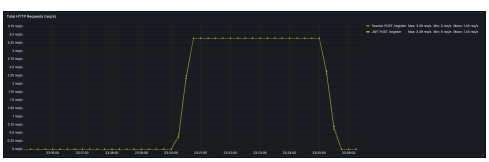


Fig. 6. Throughput Comparison

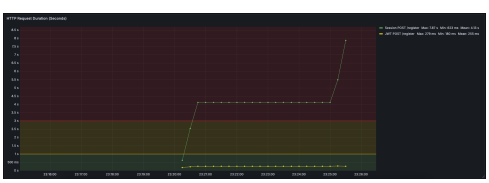


Fig. 7. Latency Comparison

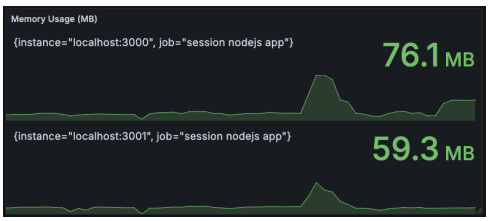


Fig. 8. Memory Usage Comparison

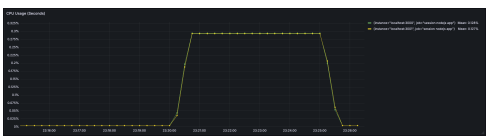


Fig. 9. CPU Usage Comparison

V. DISCUSSION OF FINDINGS

Through empirical research, this thesis sheds light on the complex trade-offs between security and performance that come into play when using either session or JWT-based session mechanisms. These two approaches to maintaining authenticated user sessions were implemented as local prototypes for the research. Using a variety of testing methods, ranging from web application security scans to low-level performance and observability tools, the research found that, while both mechanisms have their strengths and weaknesses, they offer fundamentally different security profiles that affect web application design.

A. Interpretation of Security Findings

When the OWASP ZAP scan ran on the two prototpyes – one that uses normal sessions and one that uses JWTs – a lot of warnings were reported in the OWASP ZAP scan. This gap hints that the stateless token system just touches fewer endpoints, while the session server talks to more places or endpoints. The biggest thing that jumped out was a medium-risk CSRF issue that ZAP marked as high-confidence. This means that attacker could trick a logged-in user into clicking a hidden link and the server would think it's legit because it relies on the connect.sid cookie alone. The code in session-based server didn't have a CSRF token generator. According to the OWASP rating method, this lands in the middle but worth fixing. There were also of low-risk alerts about missing security headers. Eight alerts said the Content-Security-Policy header was missing completely. Without CSP, a sneaky script could slip into a page and run XSS (Cross-Site Scripting) attacks. Three alerts flagged that the X-Content-Type-Options header was absent, meaning browsers might sniff the MIME (Multipurpose Internet Mail Extensions) type of JSON responses and treat them like HTML. There were also three "Server leaks info via X-Powered-By" warnings. This is not a serious alert to consider but it gives a hacker a clue about what stack a client is running. One alert each pointed out that /login could be brute-forced (no rate limiting) and that the session management response itself was exposed. Finally, there were twenty-four "User Agent Fuzzer" notices – low confidence, low risk alert. This basically means that a malicious could use different user-agents or browsers to send requests, enabling them to get information indicates potential information disclosure from varied headers in server-responses. The report indicates that the session server does

a decent job of checking cookies (Burp Suite showed that tampering with the connect.sid token gave a 401), but it forgets the extra layers like CSRF tokens or strict headers.

The JWT pattern looked familiar but with token-specific occurrence. The biggest medium-risk thing was again the missing CSP header – five instances this time. Same risk as before: no CSP means an attacker could inject script into any page that returns a token. The server also leaked its tech stack via the X-Powered-By header eight times. This is not a high security risk on its own but still useful intel for anyone scanning for Node.js targets. Two alerts pointed out that the JWT payload exposed Unix timestamps. That's a low-risk warning but it could let a clever attacker guess token lifetimes or do timing attacks. Missing X-Content-Type-Options showed up three more times, so the same MIME-sniffing risk exists here too. The informational bits – authentication request identified, and session management response identified – basically say "the /login route could be brute-forced or used to enumerate valid accounts". The intentionally used weak secret which Burp Suite caught was that the secret used to sign tokens was "thesis-secret-123". If an attacker guesses that secret, they can forge any token they want. The JWT endpoint did reject tampered own payloads (you get 401), but weak secrets undermine that protection. Based on the OWAPS ZAP scan reports from both servers, the session-based system leans on server memory and cookies, yet it forgets some easy defenses like CSRF tokens or CSP headers. The JWT system cuts out server state, which is nice for scaling, but then one is trusting the client with everything: timestamps, signatures, and it is critical to hide secret well.

B. Interpretation of Performance Findings

The performance findings demonstrate JWT's advantages in efficiency, with JMeter tests showing lower latency ($279\,\mathrm{ms}$ vs $7870\,\mathrm{ms}$) with the same throughput $3.39\,\mathrm{req/s}$ vs $3.39\,\mathrm{req/s}$ for 1,000 users on `/register`. Grafana visualized Prometheus metrics (`http_request_duration_seconds`), revealing reduced CPU ($0.127\,\%$ vs. $0.128\,\%$) and memory ($59.3\,\mathrm{MB}$ vs. $76.1\,\mathrm{MB}$) for JWTs, which we can attribute to stateless validation eliminating Redis lookups.

The session-based server suffers from an overloaded Redis which lead to the increased latency increased. Full Redis I/O operations (like `GET user:$username`) under load perform poorly, as do all I/O-bound operations. The OWASP risk rating methodology indirectly supports this, as impacts on performance (good or bad) have a direct bearing on availability—potentially the most serious risk of all. In the case of the session-based server, the sessions reside on the server side; this guarantees consistent validation.

C. Implications for Web Application Design

The results imply that systems based on sessions are suitable for use in applications where security is a top priority (e.g., banking). In these kinds of applications, even if a malicious user manages to lodge a forged session into the server, one is by this process limited to a particular time, hence session-based systems can provide decent security. But that requires tokens and headers to maintain the integrity of the application. At the same time, if your application is designed in such a way that it makes API calls to various microservices, then using JWTs is better because they are much more suitable for that kind of architecture. Yet JWT also must be well secured. And developers may want to adopt a hybrid approach using JWTs with server-side revocation lists.

The localhost environment (no network latency), the weak secret exaggeration of JWT vulnerabilities, and the 1,000-user scale may limit production risk estimates. They may also underestimate advanced attacks that a free-tier Burp Suite could miss. These three factors might make tool limitations appear more favorable than production reality. In future research, these could be addressed through testing in cloud environments with stronger encryption.

VI. Conclusion

Both approaches have trade-offs. The session server shows lower-level slips because it talks to more routes; the JWT server shows fewer alerts, but each one could be more severe if an attacker cracks the secret. The OWASP risk rating methodology puts most of these in low-to-medium buckets, but here it is important to note that "medium" could still mean a real breach if one ignores it. So, security is not just about picking sessions or tokens – it's about layering defenses, even the simple ones like headers and proper secrets.

If one is building a bank site, keeping control on the server – like with sessions – this is a secure approach, but there is still need CSRF tokens and proper headers to fix the ZAP warnings. For a micro-service that needs quick replies, JWT looks attractive, yet to pick a strong secret and maybe encrypt the token is critical. There is a possibility of a hybrid approach – use JWT for the front-end but keep a revocation list on the server – that could give the best of both worlds. The choice of authentication depends on the application context. The fact that there is no one-size-fits-all login method. You must weigh security against speed to get the best of both worlds and achieve optimum results.

VII. Acknowledgments

I am thankful to the people who helped me during this work. I feel grateful to my supervisors Prof. Reiner Creutzburg, Prof. Vladimir Stantchev, Prof. Hasan Dag, and M. Abubakar Bajwa. Their advice, comments, and steady faith therefore made my master's effort in the CyberMACS program at SRH Heidelberg University of Applied Sciences and Kadir Has University easier. The work presented in this research paper was partially funded by the European Union in the framework of ERASMUS MUNDUS, Project CyberMACS (Project#101082683) (https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/projects-details/43353764/101082683)

References

[1] abdaal. *Abdaal-Github/express-app-with-redis*. original-date: 2025-08-04T00:24:56Z. Aug. 19, 2025. URL: https://github.com/Abdaal-Github/express-app-with-redis (visited on 08/20/2025).

[2] Oluwasanmi Segun Adanigbo et al. "Advances in Secure Session Management for High-Volume Web and Mobile Applications". In: *International Journal of Multidisciplinary Research and Growth Evaluation* 2.1 (2022), pp. 1002–1007. ISSN: 25827138. DOI: 10.54660/.IJMRGE.2022.2.1.1002-1007. URL: https://www.allmultidisciplinaryjournal.com/search?q=MGE-2025-3-017&search=search (visited on 08/20/2025).

[3] Admin Admin. "Managing a Secure JSON Web Token Implementation By Handling Cryptographic Key Management for JWT Signature in REST API: : A survey". In: *Journal of Cybersecurity and Information Management* (Jan. 1, 2021). DOI: 10.54216/jcim.060101. URL: https://consensus.app/papers/managing-a-secure-json-web-token-implementation-by-admin/138917c6f01d5500bdea713fe572e1f1/.

[4] Salman Ahmed and Qamar Mahmood. "An authentication based scheme for applications using JSON web token". In: *2019 22nd International Multitopic Conference (INMIC)*. 2019 22nd International Multitopic Conference (INMIC). Islamabad, Pakistan: IEEE, Nov. 2019, pp. 1–6. ISBN: 978-1-7281-4001-8. DOI: 10.1109/INMIC48123.2019.9022766. URL: https://ieeexplore.ieee.org/document/9022766/ (visited on 08/19/2025).

[5] Akanksha and Akshay Chaturvedi. "Comparison of Different Authentication Techniques and Steps to Implement Robust JWT Authentication". In: *2022 7th International Conference on Communication and Electronics Systems (ICCES)*. 2022 7th International Conference on Communication and Electronics Systems (ICCES). Coimbatore, India: IEEE, June 22, 2022, pp. 772–779. ISBN: 978-1-6654-9634-6. DOI: 10.1109/ICCES54183.2022.9835796. URL: https://ieeexplore.ieee.org/document/9835796/ (visited on 08/20/2025).

[6] Rajeshwari Gadathas Krishna Babu et al. "Authentication and Access Control in Cloud-Based Systems". In: *2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN). Paris, France: IEEE, July 4, 2023, pp. 560–562. ISBN: 979-8-3503-3538-5. DOI: 10.1109/ICUFN57995.2023.10199236. URL: https://ieeexplore.ieee.org/document/10199236/ (visited on 08/20/2025).

[7] Aleksander Biberaj et al. "Substantial security challenge to web applications, using modified OTC and OWASP update". In: *International Scientific Journal Monte* (Jan. 1, 2023). DOI: 10.33807/monte.20232840. URL: https://consensus.app/papers/substantial-security-challenge-to-web-applications-using-biberaj-ndoni/b8eaf1b77d4458898075307433bfc562/.

[8] Ahmet Bucko et al. "Enhancing JWT Authentication and Authorization in Web Applications Based on User Behavior History". In: *Computers* 12.4 (Apr. 13, 2023), p. 78. ISSN: 2073-431X. DOI: 10.3390/computers12040078. URL: https://www.mdpi.com/2073-431X/12/4/78 (visited on 08/20/2025).

[9] O. Bulgakova et al. "Risk of Information Loss using JWT token (short paper)". In: (2021), pp. 292–299. URL: https://consensus.app/papers/risk-of-information-loss-using-jwt-token-short-paper-popravkin-zosimov/9f66dc4a8b4f5f3499bed75eb941bc87/.

[10] Stefano Calzavara et al. "Measuring Web Session Security at Scale". In: *Computers & Security* 111 (Dec. 2021), p. 102472. ISSN: 01674048. DOI: 10.1016/j.cose.2021.102472. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167404821002960 (visited on 08/19/2025).

[11] Sohan Singh Chinthalapudi. "Enhancing Security in ASP.NET Core Applications: Implementing Oauth, JWT, and Zero-Trust Models". In: *International Journal of Innovative Science and Research Technology* (Apr. 11, 2025). DOI: 10.38124/ijisrt/25mar1677. URL: https://consensus.app/papers/enhancing-security-in-aspnet-core-applications-chinthalapudi/cf6f91e86a235b57929ce82eb975f636/.

[12] Syabdan Dalimunthe, Emansa Hasri Putra, and Muhammad Arif Fadhly Ridha. "Restful API Security Using JSON Web Token (JWT) With HMAC-Sha512 Algorithm in Session Management". In: *IT Journal Research and Development* 8.1 (Dec. 5, 2023), pp. 81–94. ISSN: 2528-4053, 2528-4061. DOI: 10.25299/itjrd.2023.12029. URL: https://journal.uir.ac.id/index.php/ITJRD/article/view/12029 (visited on 08/20/2025).

[13] Syabdan Dalimunthe, Joeharsyah Reza, and Asep Marzuki. "Model for Storing Tokens in Local Storage (Cookies) Using JSON Web Token (JWT) with HMAC (Hash-based Message Authentication Code) in E-Learning Systems". In: *Journal of Applied Engineering and Technological Science (JAETS)* (June 30, 2022). DOI: 10.37385/jaets.v3i2.662. URL: https://consensus.app/papers/model-for-storing-tokens-in-local-storage-cookies-using-reza-marzuki/69332b2b9d685bada6d332ec9f239449/.

[14] Mahmoud Elhejazi and Wisam Muragaa. "Improving the Security and Reliability of SDN Controller REST APIs Using JSON Web Token (JWT) with OpenID and auth2.0". In: *2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)* (May 19, 2024), pp. 398–402. DOI: 10.1109/MI-STA61267.2024.10599643. URL: https://consensus.app/papers/improving-the-security-and-reliability-of-sdn-controller-muragaa-elhejazi/72790aea1d2e5a2a8b1d4dfa0efbf474/.

[15] Ambrozie Roxana Emanuela, Gavrilă Mihaela, and Tâmiceriu Daniela. "Enhancing Security in Data Ex-

change: Mitigating Risks Solutions in Base64 Encoding and JSON Web Tokens". In: *2024 International Symposium on Electronics and Telecommunications (ISETC)*. 2024 International Symposium on Electronics and Telecommunications (ISETC). Timisoara, Romania: IEEE, Nov. 7, 2024, pp. 1–4. ISBN: 979-8-3503-9086-5. DOI: 10.1109/ISETC63109.2024.10797302. URL: https://ieeexplore.ieee.org/document/10797302/ (visited on 08/20/2025).

[16] Nasrin Garmabi and M. Hadavi. "Automatic Detection and Risk Assessment of Session Management Vulnerabilities in Web Applications". In: *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)* (Oct. 28, 2021), pp. 41–47. DOI: 10.1109/ICCKE54056.2021.9721455. URL: https://consensus.app/papers/automatic-detection-and-risk-assessment-of-session-hadavi-garmabi/e7514784c3e85e609749c8f6f3c41bd1/.

[17] Manish Gupta et al. "JWTAMH: JSON Web Tokens Based Authentication Mechanism for HADOOP". In: *EAI Endorsed Trans. Scalable Inf. Syst.* 12 (July 17, 2024). DOI: 10.4108/eetsis.5429. URL: https://consensus.app/papers/jwtamh-json-web-tokens-based-authentication-mechanism-for-gupta-s/cf3817e2887f5e0981e205c04f99edbc/.

[18] K. Gutzmann. "Access control and session management in the HTTP environment". In: *IEEE Internet Computing* 5.1 (Feb. 2001), pp. 26–35. ISSN: 10897801. DOI: 10.1109/4236.895139. URL: http://ieeexplore.ieee.org/document/895139/ (visited on 08/29/2025).

[19] Seok-Woo Jang and Sang-Hong Lee. "Vulnerabilities and Encryption Applications of JWT-Based Authentication Methods". In: *Journal of Information Systems Engineering and Management* (Jan. 10, 2025). DOI: 10.52783/jisem.v10i8s.1055. URL: https://consensus.app/papers/vulnerabilities-and-encryption-applications-of-jwtbased-lee-jang/2752722ea70e50b492160a8ee0cd11d9/.

[20] Prof.B.V Karlatthe. "MERN Stack Based User Authentication Technique for Evernote Application". In: *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT* (Nov. 1, 2023). DOI: 10.55041/ijsrem27053. URL: https://consensus.app/papers/mern-stack-based-user-authentication-technique-for-karlatthe/82d8e8f6ecb754c1bee068edf772e9e3/.

[21] Aravinda Kumar and Tl Divya. "Security measures implemented in RESTful API Development". In: *Open Access Research Journal of Engineering and Technology* (Sept. 30, 2024). DOI: 10.53022/oarjet.2024.7.1.0042. URL: https://consensus.app/papers/security-measures-implemented-in-restful-api-development-kumar-divya/13c2f52681b95bda942ff4b75d1a80c9/.

[22] U. Kumaran et al. "A Secure Approach for Strengthening Session Management with Custom Session Handlers". In: *2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)* (Oct. 23, 2024), pp. 1167–1172. DOI: 10.1109/ICSSAS64001.2024.10760616. URL: https://consensus.app/papers/a-secure-approach-for-strengthening-session-management-kumaran-bhanusri/e8a5183355015d949fd1c5397a2f0904/.

[23] Pooja Mahindrakar and U. Pujeri. "Insights of JSON Web Token". In: *International Journal of Recent Technology and Engineering* (Mar. 30, 2020). DOI: 10.35940/ijrte.f7689.038620. URL: https://consensus.app/papers/insights-of-json-web-token-pujeri-mahindrakar/ef0d2e3f8cc356f09cb0c65a1b955899/.

[24] Malvin Malvin and Cutifa Safitri. "JSON Web Token Leakage Avoidance Using Token Split and Concatenate in RSA256". In: *Indonesian Journal of Computing, Engineering and Design (IJoCED)* (Apr. 3, 2023). DOI: 10.35806/ijoced.v5i1.325. URL: https://consensus.app/papers/json-web-token-leakage-avoidance-using-token-split-and-malvin-safitri/b6abb26cb290545680f6fa5bdd856a1c/.

[25] Ayodeji Ismail Moshood and Zoe Jeffrey. "An In-Depth Approach to Strengthening Security in Open-Access Libraries Utilizing JSON Web Tokens (JWT)". In: *International Journal of Recent Technology and Engineering (IJRTE)* (Jan. 30, 2025). DOI: 10.35940/ijrte.e8181.13050125. URL: https://consensus.app/papers/an-indepth-approach-to-strengthening-security-in-moshood-jeffrey/3b6ef9c135155ce6b043fbda64981066/.

[26] Ahmad Yahya Nashikhuddin, Jamilah Karaman, and Yovi Litanianda. "IMPLEMENTASI API RESTFUL DENGAN JSON WEB TOKEN (JWT) PADA APLIKASI E-COMMERCE THRIFTY SHOP UNTUK OTENTIKASI DAN OTORISASI PENGGUNA". In: *METHOMIKA Jurnal Manajemen Informatika dan Komputerisasi Akuntansi* (Oct. 31, 2023). DOI: 10.46880/jmika.vol7no2.pp239-246. URL: https://consensus.app/papers/implementasi-api-restful-dengan-json-web-token-jwt-pada-nashikhuddin-karaman/da20fc57a55a538bac2504b64d6d3079/.

[27] Adiva Fiqri Nugraha et al. "Performance and Security Comparison of Json Web Tokens (JWT) and Platform Agnostic Security Tokens (PASETO) on RESTful APIs". In: *2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs)*. 2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs). Bogor, Indonesia: IEEE, Aug. 22, 2023, pp. 15–22. ISBN: 979-8-3503-3943-7. DOI: 10.1109/ICoCICs58778.2023.10277377. URL: https://ieeexplore.ieee.org/document/10277377/ (visited on 08/20/2025).

[28] *OWASP Risk Rating Methodology — OWASP Foundation*. URL: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology (visited on 08/26/2025).

[29] Connor Potter, N. Saxena, and Soumyadev Maity. “Clarity: Analysing Security in Web Applications”. In: *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)* (Jan. 3, 2023), pp. 522–528. DOI: 10.1109/COMSNETS56262.2023.10041289. URL: https://consensus.app/papers/clarity-analysing-security-in-web-applications-maity-potter/082274d11e1859a8b682ff02272df4ae/.

[30] Mykola Pyroh, Ganna Tereshchuk, and Oleksandr Toroshanko. “AUTHENTICATION PRINCIPLES AS SECURITY ASPECTS OF WEB DEVELOPMENT”. In: *MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES* (Feb. 27, 2025). DOI: 10.31891/2219-9365-2025-81-36. URL: https://consensus.app/papers/authentication-principles-as-security-aspects-of-web-pyroh-toroshanko/728c37f2e8925bb6bda810fa28932ff4/.

[31] Et Al. Manish Rana. “Enhancing Data Security: A Comprehensive Study on the Efficacy of JSON Web Token (JWT) and HMAC SHA-256 Algorithm for Web Application Security”. In: *International Journal on Recent and Innovation Trends in Computing and Communication* (Nov. 5, 2023). DOI: 10.17762/ijritcc.v11i9.9930. URL: https://consensus.app/papers/enhancing-data-security-a-comprehensive-study-on-the-rana/5f9a251b1f9757f089b4214104f56d0e/.

[32] Seok-Woo Jang. “Vulnerabilities and Encryption Applications of JWT-Based Authentication Methods”. In: *Journal of Information Systems Engineering and Management* 10.8 (Jan. 10, 2025), pp. 377–384. ISSN: 2468-4376. DOI: 10.52783/jisem.v10i8s.1055. URL: https://jisem-journal.com/index.php/journal/article/view/1055 (visited on 08/19/2025).

[33] Y. Sheffer, D. Hardt, and M. Jones. *JSON Web Token Best Current Practices*. RFC8725. RFC Editor, Feb. 2020, RFC8725. DOI: 10.17487/RFC8725. URL: https://www.rfc-editor.org/info/rfc8725 (visited on 08/20/2025).

[34] Ilgar Shikhverdiyev et al. “Secure authentication in e-government 2.0: a comparative analysis of traditional session-based and modern jwt-based authentication”. In: *International Science Journal of Engineering & Agriculture* 3.6 (Dec. 1, 2024), pp. 117–129. ISSN: 2720-6319. DOI: 10.46299/j.isjea.20240306.12. URL: https://isg-journal.com/isjea/article/view/884 (visited on 08/20/2025).

[35] Shellie Wedman, Annette Tetmeyer, and Hossein Saiedian. “An Analytical Study of Web Application Session Management Mechanisms and HTTP Session Hijacking Attacks”. In: *Information Security Journal: A Global Perspective* 22.2 (Mar. 4, 2013), pp. 55–67. ISSN: 1939-3555, 1939-3547. DOI: 10.1080/19393555.2013.783952. URL: http://www.tandfonline.com/doi/abs/10.1080/19393555.2013.783952 (visited on 08/19/2025).

[36] Iskander Zulkarneev and Konstantin A. Basalay. “JSON Web Tokens Lifecycle-Based Threat Classification”. In: *2024 IEEE 25th International Conference of Young Professionals in Electron Devices and Materials (EDM)*. 2024 IEEE 25th International Conference of Young Professionals in Electron Devices and Materials (EDM). Altai, Russian Federation: IEEE, June 28, 2024, pp. 1920–1924. ISBN: 979-8-3503-8923-4. DOI: 10.1109/EDM61683.2024.10615042. URL: https://ieeexplore.ieee.org/document/10615042/ (visited on 08/20/2025).

Enhancing Cloud Security: Best Practices for Deploying Advanced Firewalls in Cloud Architectures

1st Lenear Amagove Mwondi
Department of Administrative Sciences
Kadir Has University
Istanbul, Turkey
lenear.mwondi@stu.khas.edu.tr

2nd Onyango Allan Onyango
Department of Administrative Sciences
Kadir Has University
Istanbul, Turkey
onyango.allan@stu.khas.edu.tr

3rd Tajriyan Rahman
Department of Administrative Sciences
Kadir Has University
Istanbul, Turkey
tajriyanrahman@stu.khas.edu.tr

***Abstract*—Cloud computing technology has revolutionized data storage, access, and management,offering a scalable, and economically significant solution. Nevertheless, its rapid adoption of has introduced complex security risks that traditional perimeter-based security tools can no longer mitigate effectively. The nature of Cloud infrastructures is evolving, and cyber threats level is so high that they become a massive risk to organizational information and activities. This paper reviews the changing environment of threats in the cloud and the role advanced firewalls contribute to the benefit of the cloud with regard to security. Advanced cloud firewalls have capabilities of Deep Packet Inspection (DPI), Intrusion Detection and Prevention Systems (IDPS), application-aware filtering, and multi-cloud and hybrid environments. Key cloud security challenges include multi-tenancy, misconfiguration, insider threats, and regulatory compliance (e.g., GDPR, HIPAA, PCI DSS which demand dynamic and integrated controls. We then explore how advanced firewalls, combined with artificial intelligence and threat intelligence feeds, offer real-time threat detection, traffic control, and resilience. Deployment strategies include cloud- native integration, infrastructure-as-code, segmentation through Network Security Groups (NSGs), and zero trust. Best practices are outlined for optimizing performance, visibility, and enforcing consistent policies including integration with SIEM and SOAR. Industry case studies, such as Capital One and Dropbox, demonstrate the implications of poor firewall configurations and the value of advanced security strategies. Integrating firewalls with CI/CD pipelines in DevOps is emphasized for continuous protection. Despite challenges in configuration scalability, performance, and costs, these demands are driving scalability in AI machine learning and automation within cloud security.**

***Index Terms*—Cloud Security, Advanced Firewalls, Deep Packet Inspection (DPI), Intrusion Detection and Prevention System (IDPS), Network Security Groups (NSG), Zero Trust, SIEM, SOAR, Multi-cloud, Firewall Deployment, Threat Mitigation**

I. Introduction

Cloud computing technology has revolutionized large data storage, access, and management, and has offered flexible, scalable, and economically significant solutions to organizations. Nevertheless, due to the growing popularity of cloud architecture (Figure 1), there have been extreme security problems. The ever-changing character of cloud infrastructures and a high and sophisticated level of cyber threats pose an enormous risk

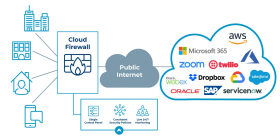


Figure 1. Multi-cloud reference security architecture [1]

to the organizational data and business operations. Complex firewalls are some of the most vital security measures that could be used in cloud shielding. These firewalls are better and more preferred than those offered by traditional network firewalls, as they incorporate application-based filtering, threat identification and prevention, and compatibility with threat databases. They are very important in cloud security because they are there to fill gaps, manage access, and minimize losses of data. The intended research paper aims at studying and defining the appropriate approach to the implementation of sophisticated cloud infrastructure firewalls. The objectives are to discuss the advantages and the drawbacks of the upcoming advanced firewalls, describe how they are possible to apply them, and know their issues and limitations. This article assesses the way in which designed firewalls optimize a joyful cloud environment based on real-life predictions and the trend of the organization. Since the world is more connected than ever, proper cloud security strengthening has become crucial for organizations keen on protecting their properties and staying relevant. When used appropriately, an advanced firewall is one of the best security layers against new and emerging threats.

II. Overview of Cloud Security

A. Cloud Security Fundamentals

Cloud security refers to a concept that embraces technologies, policies, and controls for the protection of cloud-based systems, data, and cloud infrastructure. This model of responsibility that is shared is at the center of cloud security, which prescribes different responsibilities for cloud Service Providers (CSPs) and their consumers. CSPs are required to secure space within

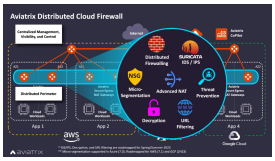


Figure 2. Distributed Cloud Security Firewalls [2]

the cloud, while customers are expected to safeguard the applications and data and control entry to the cloud [1]. The inability of these divisions to effectively protect these crucial intels emphasizes the importance of organizations not waiting for threat incidents to put into practice efficient security systems.

B. Cloud Security Issues

However, cloud computing has unique security issues that differentiate it from traditional models.

Multi-tenancy: Using pools, of which shared resources are present in the public clouds, raises the threats posed by data breaches.

Misconfiguration: The Contravene of presets on the cloud service is one of the top consistent causes of cloud breaches, for instance, when one leaves open data in the cloud folder.

Insider Threats: Threats originating from employees or contractors working for the cloud services provider or the enterprise can endanger cloud environments.

Compliance Requirements: Industry-specific guidelines such as the General Data Protection Regulation(GDPR), Health Insurance Portability and Accountability Act(HIPAA), and Payment Card Industry Data Security Standard(PCI DSS) are always in force, making cloud security challenging for organizations.

Due to the activation of cloud services and the fast deployment of applications, information security often lacks a solid envelope; therefore, using advanced tools is necessary [2].

C. The Use of Firewalls in Cloud-Related Security

Firewalls are the first layer of security in cloud technology. Conventional firewalls that protect well-defined physical perimeters (Figure 2), perfected for local area inheritance, are unfit for cloud infrastructures' foggy, dispersed nature. On the other hand, more excellent firewalls are developed to overcome the above challenges. Some features include Deep Packet Inspection (DPI), Intrusion Detection and Prevention System(IDPS), and multi-cloud and hybrid cloud systems policy management. Advanced firewalls' benefits include filtering traffic, identifying everyday issues, and preventing threats that compromise organizations' stability and compliance. Due to their adaptability to changing threats, they play an irreplaceable role in protecting contemporary cloud structures.

III. Advanced firewalls: Features and Capabilities

A. Characteristics of advanced firewalls

The current type of firewall is enhanced to meet the challenges of modern cloud networks beyond the network firewall. Application awareness filtering is provided, which, instead of filtering by IP addresses or ports, filters applications with built-in easy access control for the client. The intrusion detection and prevention systems (IDPS) provide accurate time-based prevention and detection. At the same time, deep packet inspection DPI is the inspection of the content of packets in the network for threats. These capabilities offer integrated protection, covering traffic flowing in inputs and outputs.

B. Accelerating Technologies in Enhanced Firewall

To increase the capabilities of firewalls, modern technologies are used in the production of advanced firewalls. Machine learning and artificial intelligence are combined for the most effortless predictive data analysis, and these firewalls may have the ability to identify and mitigate emergent threats quickly. It was reported that many solutions are now being made to integrate threat intelligence feeds, which are a feed of data that details current global threats and weaknesses. Furthermore, cloud-native firewalls are designed to work within the cloud computing environment while adjusting to the workload of cloud deployments.

C. The comparison with traditional firewalls

Conventional firewalls act at a network level, enforcing packet filtering that is consistent with standard rules. They are efficient for organizing on-premise systems' elements, but do not work effectively with the flexibility and evolution of cloud-based systems. However, advanced firewalls are designed not to have such constraints and give the following features: dynamic policy, deep traffic analysis across several layers, and integration with cloud- native services. In addition, conventional firewalls mainly afford poor protection against application layer attacks, which has become a significant issue in the context of the cloud. These firewalls incorporate innovative technologies and an integrated approach, providing a higher degree of protection necessary for contemporary cloud environments. Because of this flexibility, the Network Security Groups (NSGs) are a crucial part of any sound cloud security plan.

IV. Deployment strategies for advanced firewalls in cloud architectures

A. Planning and Design

To efficiently put into use the manifold misconceptions about an advanced firewall in a cloud environment, some considerations should be made on the planning and design aspects. The organizational security requirements must be determined depending on the information the organization processes, the regulations it needs to fulfill, and the threats it faces. Most importantly, the firewall settings must consider features provided by the cloud provider, including Azure Network Security Groups, Google Cloud Firewalls, and AWS Security

Groups. That way, it is compatible with other applications and has the optimum function in the cloud environment. Organizing logical zones and network segmentation is necessary to call a design well-structured. To achieve better security, team workloads should be divided into zones according to risk levels; this minimizes the execution space when applying security policies since the risk of escalation should be minimized.

B. Implementation Best Practices

During the implementation process, organizations should pay attention to using technology tools to perform such duties as generating and enforcing policies to minimize mistakes and improve standardization [3]. It's also possible to create policy blueprints and leverage infrastructure-as-code Configurations to deploy security settings across many cloud providers. API level integration is also key in ensuring firewalls are integrated with other layers of security gadgets like IPS or IAM. They should apply firewalls pervasively, referencing the zero-trust architecture, in which under no circumstance is any user or device trusted. This includes implementing least privilege access and continuously authenticating users and their activities. A decentralized firewall framework provides a competitive advantage as it reduces a critical failure point, guaranteeing better performance and scalability [4]. In addition to these measures, administering pre-authentication request filtering at the firewall level enhances organizational security. For instance, cloud firewalls utilize Bloom filter technology to swiftly identify and stop doubtful login attempts prior to them hitting the authentication server. [5].

C. Performance Optimization

One must always choose between security and performance, mainly when using advanced firewalls. When misconfigured, firewalls act like choke points and thus hinder the performance and Quality of Service delivered in a given network. Load balancing solutions must divide traffic across firewalls so that all firewalls work optimally during high-traffic periods [6]. Further, auto-scalable solutions enable firewalls to adapt their capabilities according to the current traffic load while keeping protection steady and fast.

D. Monitoring and Maintenance

Maintenance thus becomes critical once the systems have been deployed, as threats are likely to be identified and countered without long waiting periods. Sophisticated implementations provide highly usable and extensible firewall management consoles and logging and analysis gear to support visibility into network traffic [7], and therefore leveraging file monitoring and security analytics is essential [8]. Firewalls have frequently updated policies and software to decrease the effects of new vulnerabilities. Security audits should be conducted regularly to ensure that the firewall settings correspond to the changes in the business and the security requirements.

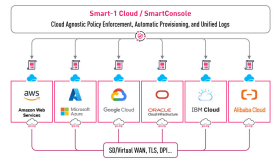


Figure 3. Cloud Service Providers [9]

V. Best Practices for Cloud Firewall Deployment

For proper firewall deployment, organizations have to integrate with the native tools or features of the cloud vendors, as shown in Figure 3. AWS, Azure, Google, etc., currently offer their respective FW solutions, namely AWS WAF, Azure FW, etc. These are designed to optimize the specific architecture of the platforms they are built on and are very easy to integrate well into existing systems [9]. Getting the most out of these tools in combination with third-party firewalls improves security in general and operational compatibility in particular.

A. Achieving Policy Coherency Across Heterogeneous Systems in Multi-Clouds

Indeed, managing consistent firewall policies is essential. Organizations should adopt centralized policy management tools to make this process easier. These tools enable administrators to set, implement, and modify security policies about multiple clouds from one location. Also, using pre-design templates helps reduce the configuration management for different policy formats in various cloud providers to an optimally achievable level.

B. Integration with Bigger Security Frameworks

Therefore, advanced firewalls should not operate in isolation; they should form part of an overall strategy. Having firewalls tied into Security Information and Event Management (SIEM) and Security Orchestration, Automation, and Response (SOAR) makes for stronger threat detection and mitigation. Likewise, with Identity and Access Management (IAM) tools, communication can be coordinated to reflect the organization's overall zero-trust approach to access controls [10].

C. From time to time, screening and assessment

Firewall configurations should be checked periodically through vulnerability assessments, penetration tests, constant updates, infrastructure audit, and monitoring [11]. These tests show organizational vulnerabilities in specific policies and assist in responding to different threats. Such assessments continuously improve firewalls and ensure they meet ongoing business requirements.

VI. Case Studies and Industry Insights

A. Case Study 1: Capital One - Cloud Security and Firewall Implementation

In 2019, Capital One had a significant breach of its firewall settings in an AWS server. Specifically, the breach of over 100

million customers' records was a pressing issue in securing cloud environments. In response, Capital One deepened its cloud security by pushing forward highly developed AWS Web Application Firewalls (WAFs) and network firewalls. The organizations introduced advanced segmentation, encryption, and access control measures in cloud architecture [12]. New features for monitoring, such as automated monitoring tools and threat intelligence integration, were incorporated to enable real-time identification of risks. After this, there was a shift at Capital One to a zero-trust security paradigm to verify any user or device that wanted to access data deemed sensitive. In particular, the company's firewall strategy changes significantly contributed to stopping new incidents.

B. Case Study 2: Dropbox – Firewalls and Cloud Infrastructure Security

Another modern cloud storage provider, Dropbox, also reinforced the company's cloud framework with enhanced firewalls for data security from unauthorized access. With millions of people using it as storage, data protection became valuable in uncontested certainty. Firewalls were put in place to filter both inward and outbound traffic, but with an emphasis on the application layer [13]. It aligned these firewalls with its infrastructure as code to enable the automation of policies and its growing, dynamic cloud environment. This strategy also helped you avoid DDoS attacks and unauthorized access to important files. Through active and consistent policy refresh and traffic checks, Dropbox could reaffirm both data velocities for its worldwide users and security.

VII. Challenges and limitations in advanced firewall deployment

It becomes difficult to guarantee that policies are followed endlessly across different platforms and that the firewall runs efficiently without management solutions.

A. The Concerns of Scalability and Performance

Although with the current growth of cloud infrastructure, there are advanced firewalls in the market that can grow with the infrastructure, achieving high performance at scale can be challenging. Firewalls receiving many packets may be overwhelmed and, hence, slow in processing, becoming a bottleneck. To address these challenges, organizations must use load-balancing solutions and auto-scaling firewalls to keep them operational; these Exploratory tools may demand other configurations and constant monitoring. The problem is in the attempt to meet the needs for high performance of cloud applications while ensuring the security of the application and data, especially if the applications are considered mission-critical, where a low-latency response is vital.

B. Cost and Resource Allocation

High-end firewalls may come at a high-end price when higher-grade features such as deep packet inspection, real-time threat intelligence, and auto-response are incorporated into the firewall. These firewalls can be expensive, and if organizations use multiple cloud types or a hybrid approach, the cost of various instances will need to be considered. This can put pressure on budgets and expectations of the outcome needed, thus requiring the use of any available resources.

C. Evolving Threats

It's important to understand that, like all other threats in the cyber terrain, the mechanisms and guarding software must be updated. New incidents occur daily, demonstrating that intruders will always find new ways to get past standard layers of protection. The enhanced Firewalls must be frequently updated to remove new threats, which can be very time-consuming and may require relevant professionals.

VIII. Future trends in cloud firewall technologies

A. Machine Learning and Artificial Intelligence Integration

As cybersecurity threats advance, firewalls also incorporate AI and ML. It empowers firewalls to identify and prevent new and unknown threats with improved abilities to respond immediately as they emerge, even before existing signature-based methods can locate them [14]. Advanced intelligent firewalls allow the learned behavior of the network traffic and thus become better at detecting the peculiarities that the zero-day attack frameworks may have exploited. This shift towards intelligent security systems will significantly improve the functionality of firewalls in terms of quickly evolving threats in cloud systems. Sustainability and Advanced Automation are shifting their focus to greater automation due to many factors, including scalability and timely threat response. Today, firewalls are being developed to work with SOAR, which automates responding to security threats in real-time . These systems can coordinate variable settings, perform firewall modifications, initiate corrective measures, and invoke incident handling procedures without human interference. This trend will decrease the trust in the human workforce in this process, shorten response time, and enforce conformity to standard security measures in multi-cloud settings.

B. Technical connections with DevOps and CI/CD processes

Many organizations use GitHub to automate the continuous integration and deployment processes and workflows. Checkmarx must naturally fit into these processes as organizations adopt DevOps practices. Firewalls will act as enablers for security since the process of developing solutions will incorporate the various security measures at the core, from which firewalls will provide immediate security policies as new code and applications are deployed [15].This integration will make cloud security more preventive, as threats and risks will be dealt with before getting to the production systems.

C. An examination of Firewall Development in the light of Quantum Computing

Although more of a decoy in the current market, quantum computing will soon change the face of cloud security. The upcoming quantum technology requires improved firewall methods to handle the operational quantum algorithm, which

might compromise secure firewalls. As a result, firewalls will have to include quantum-resistant cryptographic approaches in their best efforts to sustain cloud security.

IX. Conclusion

In conclusion, firewalls tackle emerging, tricky threats and secure cloud structures for organizations worldwide. Some of their enhancements are: intrusion prevention, deep packet inspection, and application layer filtering, which have proven incredibly relevant for cloud- provided applications and services. However, organizations encounter several factors when deploying the solutions, including Configuration and Scalability, Performance, and Costs. Nevertheless, current challenges are opening the path toward creating novel and futuristic technologies within the security environment, including AI, machine learning, and automation. In the future, the advancements to Cloud Firewall technologies, including integrating DevOps processes and adding quantum-safe security features, will bring more robustness to cloud security. To be prepared for further cyber threats, organizations must implement secure, flexible firewalls and ensure that they are adequately integrated with more general protection concepts about their changes by the nature of cloud environments and threats. Hence, they will safeguard their data, nurture future business, and comply with the law.

References

[1] A. Caballero, "Advanced private cloud computing security architectures," in *Security in the Private Cloud*. CRC Press, 2016, pp. 303–318.

[2] D. R. Chittibala, "Web application firewalls (waf) integration in devops practices: A scholarly exploration of security, automation and continuous protection," *Journal of Artificial Intelligence & Cloud Computing*, pp. 1–5, 2022.

[3] S. Kanungo and S. Sarangi, "Quantum computing integration with multi-cloud architectures: Enhancing computational efficiency and security in advanced cloud environments," *World Journal of Advanced Engineering Technology and Sciences*, vol. 12, no. 2, pp. 564–574, 2024.

[4] D. B. M, R. P, P. M, P. K. S, and P. K. RC, "Resources provisioning cost optimization in a decentralized cloud firewall framework," in *2022 1st International Conference on Computational Science and Technology (ICCST)*, 2022, pp. 311–315.

[5] Y. Fu, M. H. Au, R. Du, H. Hu, and D. Li, "Cloud password shield: A secure cloud-based firewall against ddos on authentication servers," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, 2020, pp. 1209–1210.

[6] S. Lekkala, "Next-gen firewalls: Enhancing cloud security with generative ai," *Journal of Artificial Intelligence & Cloud Computing*, vol. 3, no. 4, pp. 1–9, 2024.

[7] P. Pandya and R. Rahmo, "Advanced security architectures for private cloud computing," in *Security in the Private Cloud*. CRC Press, 2016, pp. 287–302.

[8] S. Moiz, A. Majid, A. Basit, M. Ebrahim, A. A. Abro, and M. Naeem, "Security and threat detection through cloud-based wazuh deployment," in *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, 2024, pp. 1–5.

[9] V. Gunnam and N. B. Kilaru, *Securing PCI Data: Cloud Security Best Practices and Innovations*. Novelty Journals, 2024.

[10] J. K. Manda, "Cloud security best practices for telecom providers: Developing comprehensive cloud security frameworks and best practices for telecom service delivery and operations, drawing on your cloud security expertise," *SSRN Electronic Journal*, 2024.

[11] M. Roopesh, "Cybersecurity solutions and practices: Firewalls, intrusion detection/prevention, encryption, multi-factor authentication," *Academic Journal on Business Administration, Innovation & Sustainability*, vol. 4, no. 3, pp. 37–52, 2024.

[12] Amazon Web Services (AWS), "AWS security best practices," https://aws.amazon.com/security/, accessed: 2025-07-03.

[13] Dropbox, "Dropbox security overview," https://www.dropbox.com/business/security, accessed: 2025-07-03.

[14] M. N. U. Haq and M. K. Sharma, "Mastering cloud security: Techniques and best practices," in *Emerging Trends in Cloud Security and Intelligent Agents*. Emerging Technologies Publishing, 2023.

[15] L. Zhang and L. Chen, "Best practices of cloud dcn deployment," in *Cloud Data Center Network Architectures and Technologies*. CRC Press, 2021, pp. 313–353.

Ensemble-Based Machine Learning Models for Cybersecurity: Theoretical Guarantees and Empirical Insights

1st Zhivko Atanaskoski
Ss. Cyril and Methodius University
Faculty of Computer Science and Engineering
Skopje, N. Macedonia
zivko.atanaskoski@finki.ukim.mk

2nd Stefan Mirchevski
Ss. Cyril and Methodius University, Faculty of Civil Engineering
European University, Faculty of Informatics
Skopje, N. Macedonia
stefan_mircevski@outlook.com, stefan.mircevski@eurm.edu.mk

3rd Vesna Dimitrova
Ss. Cyril and Methodius University
Faculty of Computer Science and Engineering
Skopje, N. Macedonia
vesna.dimitrova@finki.ukim.mk

4th Aleksandra Popovska-Mitrovikj
Ss. Cyril and Methodius University
Faculty of Computer Science and Engineering
Skopje, N. Macedonia
aleksandra.popovska.mitrovikj@finki.ukim.mk

***Abstract*—This paper provides a comparative study of ensemble classifiers: Random Forests, Rotation Forests, Bagging, AdaBoost, and Gradient Boosted Trees, in cybersecurity classification tasks. We first outline key theoretical aspects of these methods, including their bias–variance behavior, risk bounds, and relation to the Bayes optimal classifier. Building on this overview, we conduct an experimental analysis using a malware dataset, where the empirical results are interpreted in light of these theoretical properties, particularly with respect to excess risk. The framework incorporates feature preprocessing and multiple evaluation metrics (ROC AUC, PR AUC, accuracy, precision, recall, and F1 score). Results highlight the strong bias-variance balance of Random Forests and Gradient Boosted Trees, as well as the decorrelation advantage of Rotation Forests. The study offers practical guidance for applying ensemble learning to vulnerability analysis. To our knowledge, this is a comparative study that grounds ensemble classifiers on CIC-MalMem-2022 within a theoretical risk framework, while also extracting feature-level insights relevant for operational malware detection. This integration highlights not only which models perform best, but why they succeed in adversarial cybersecurity settings.**

***Index Terms*—malware classification, ensemble methods, machine learning models, excess risk, convergence guarantees, cybersecurity.**

I. Introduction

Ensemble-based machine learning models have evolved as one of the most effective paradigms for improving predictive performance, robustness, and generalization. Techniques such as Bagging, Random Forests, Rotation Forests, AdaBoost, and Gradient Boosted Trees extend the capabilities of individual learners by combining their outputs, balancing bias and variance, and enhancing stability across diverse datasets. Over the past two decades, these methods have matured from theoretical constructs into widely accepted tools, fueling advances in diverse domains ranging from multimedia analysis to healthcare and, increasingly, cybersecurity. In cybersecurity, ensemble methods are particularly attractive due to the heterogeneity and adversarial nature of threats. They have been successfully applied to intrusion detection, malware family classification, cryptographic analysis, and anomaly detection in IoT systems. Their ability to capture complex, high-dimensional structures while maintaining resilience against noise makes them well-suited for security-critical applications where errors carry high operational costs. Despite their practical success, a comprehensive understanding of the theoretical guarantees underpinning ensemble classifiers remains essential. Guarantees such as consistency, convergence to the Bayes optimal classifier, and excess risk bounds not only justify their empirical robustness but also provide insights into when and why particular ensembles succeed. Establishing these guarantees strengthens trust in deploying ensemble models in high-stakes cybersecurity settings, where empirical performance alone is insufficient. The rest of the paper is organized as follows. Section II reviews related work on ensemble learning and its cybersecurity applications. Section III gives an overview of the theoretical properties and guarantees of the studied ensemble classifiers. Section IV describes the dataset and preprocessing pipeline. Section V reports empirical results and analysis. Finally, Section VI concludes with key insights, practical recommendations, and directions for future work.

II. Related Work

Ensemble methods are a well-established approach to improving classification accuracy, robustness, and generalization [1]. Layered ensemble frameworks for binary classification were introduced in [2], while [3] integrated data envelopment analysis with machine learning to enhance ensemble decision-making. Machine learning has also been applied in

cryptographic contexts, including DES cryptanalysis [4] and symmetric cryptography algorithms [5].

In multimedia and content security, ensemble classifiers have been proposed for encrypted video identification [6], and have been comprehensively explored for IoT cybersecurity across multi-class and binary tasks [7]. Evolutionary optimization of SVM cascades for ensemble construction was investigated in [8], while [9] reviewed the development of Random Forests. Margin-theoretic perspectives on class imbalance in ensemble learning were examined in [10].

Ensemble approaches have further been applied to intrusion detection in IoT [11], multilabel classification of user-generated content [12], image spam detection [13], and selective cognitive security analysis [14]. Other applications include cryptographic cipher identification [15] and cryptanalysis using information-theoretic metrics [16]. Decentralized Bayesian ensemble learning was proposed in [17], and causal analysis-based aggregation for Bayesian network learning in [18]. Formal statistical underpinnings for consistent nonparametric learning were established in [19], while [20] addressed bias issues in decision trees and Random Forests. A multi-classifier fusion strategy based on Dempster–Shafer theory was presented in [21].

Compared to these works, our study provides a unified theoretical and empirical analysis of Random Forests, Rotation Forests, Bagging, AdaBoost, and Gradient Boosted Trees for multi-class cybersecurity classification, including explicit excess risk bounds, asymptotic consistency results, and convergence rate guarantees.

III. Overview of Some Theoretical Properties of Ensemble-Based Multi-Class Machine Learning Models

We consider C-class classification with random feature vector $\mathbf{X}$ in $\mathbb{R}^d$, random class label $Y \in \{1, \ldots, C\}$, $C \geq 3$, and the i-th i.i.d. training sample $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ drawn from the joint distribution of $(\mathbf{X}, Y)$. Each base classifier $g_n^{(m)}$ is a function trained on the n samples that maps a feature vector $\mathbf{X} \in \mathbb{R}^d$ to one of the C class labels Y. The superscript m indexes the classifier in the ensemble, while the subscript n indicates its dependence on the training sample size, i.e., $g_n^{(m)} : \mathbb{R}^d \to \{1, \ldots, C\}$ via an aggregation rule $A(\cdot)$:

$$\hat{g}_n(\mathbf{x}) = A\big(g_n^{(1)}(\mathbf{x}), \ldots, g_n^{(M)}(\mathbf{x})\big).$$

In probability averaging, each base classifier $g_n^{(m)}$ produces an estimated posterior $\hat{\mathbf{p}}_n^{(m)}(\mathbf{x}) = \big(\hat{p}_{n,1}^{(m)}(\mathbf{x}), \ldots, \hat{p}_{n,C}^{(m)}(\mathbf{x})\big)$, where $\hat{p}_{n,c}^{(m)}(\mathbf{x}) \approx P(Y = c \mid \mathbf{X} = \mathbf{x})$ and $\sum_{c=1}^C \hat{p}_{n,c}^{(m)}(\mathbf{x}) = 1$. The ensemble posterior is obtained by averaging across classifiers:

$$\hat{p}_{n,c}(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^M \hat{p}_{n,c}^{(m)}(\mathbf{x}), \quad \hat{g}_n(\mathbf{x}) = \arg \max_{c \in \{1,\ldots,C\}} \hat{p}_{n,c}(\mathbf{x}).$$

The performance of a classifier is measured by the *misclassification risk*

$$R(\hat{g}_n) = \mathbb{E}[I\{\hat{g}_n(\mathbf{X}) \neq Y\}],$$

that is, the expected probability of predicting an incorrect label. The optimal achievable error is the *Bayes risk* $R^* = \inf_g R(g)$, attained by the Bayes classifier

$$g^*(\mathbf{x}) = \arg \max_{c \in \{1,\ldots,C\}} P(Y = c \mid \mathbf{X} = \mathbf{x}).$$

The *excess risk* of an estimator $\hat{g}_n$ is defined as

$$\mathcal{E}(\hat{g}_n) = R(\hat{g}_n) - R^*,$$

which quantifies the gap between its performance and the Bayes optimal classifier. An estimator $\hat{g}_n$ is called *consistent* if $\mathcal{E}(\hat{g}_n) \to 0$ as $n \to +\infty$. Finally, the *convergence rate* of $\hat{g}_n$ is expressed as $\mathcal{E}(\hat{g}_n) = \mathcal{O}(a_n)$, where the sequence (a_n) depends on factors such as the smoothness of the conditional distribution $P(y|\mathbf{x})$, the complexity of the base learners, and the diversity among classifiers in the ensemble.

A. Random Forests

Random Forests, [22], are an ensemble method that grow M deep decision trees on bootstrap samples of the training data. At each split in a tree, a random subset of $d_{\text{try}} < d$ features is considered, which serves to decorrelate the individual learners and reduce variance.

Each tree $g_n^{(m)}$ produces leaf-based posterior estimates

$$\hat{p}_{n,c}^{(m)}(\mathbf{x}) = \frac{1}{|S_\ell|} \sum_{i \in S_\ell} I\{Y^{(i)} = c\},$$

where S_ℓ denotes the set of training points falling in the leaf ℓ containing $\mathbf{x}$, and $I\{\cdot\}$ is the indicator function. Intuitively, this is the empirical frequency of class c among the samples in that leaf.

Under standard conditions, [23]: sufficient tree depth, feature randomization, and bootstrap subsampling, purely random forests are consistent, meaning the excess risk $\mathcal{E}(\hat{g}_n) = R(\hat{g}_n) - R^*$ vanishes as $n \to +\infty$. More precisely, for a d-dimensional feature space, theoretical analyses establish the convergence rate $\mathcal{E}(\hat{g}_n) = \mathcal{O}\big(n^{-1/(d+2)}\big)$, where the rate reflects the interplay between the sample size n and the dimensionality d. Higher-dimensional feature spaces slow down convergence, while increasing the number of trees M or ensuring sufficient randomization improves stability and reduces variance in the ensemble predictions.

B. Rotation Forests

In Rotation Forest, each base tree $g_n^{(m)}$ is trained on a rotated feature space $\mathbf{x}' = Q^{(m)}\mathbf{x}$, where $Q^{(m)} \in \mathbb{R}^{d \times d}$ is an *invertible* (typically orthogonal) matrix. The rotation serves two purposes: (i) preserve the information in the original features, and (ii) diversify the trees to reduce correlation among base classifiers.

The excess risk of the ensemble can be decomposed as

$$\mathcal{E}(\hat{g}_n) = R(\hat{g}_n) - R^* \leq \frac{1}{M} \sum_{m=1}^M \big(R(g_n^{(m)}) - R^*\big) + \frac{1}{M^2} \sum_{m \neq \ell} \mathrm{Cov}\Big(I\{g_n^{(m)}(\mathbf{X}) \neq Y\},\, I\{g_n^{(\ell)}(\mathbf{X}) \neq Y\}\Big),$$

where the first term in the sum is the average excess risk of individual trees, and the second term captures the correlation between base errors.

Since $Q^{(m)}$ is invertible, the Bayes decision boundary in the rotated space is identical to that in the original space. Consequently, consistency of each base classifier $g_n^{(m)}$ implies consistency of the ensemble $\hat{g}_n$. The convergence rates of Rotation Forest are comparable to Random Forests, provided that the rotations sufficiently reduce correlation among base trees, [24].

C. Bagging

Bagging trains each base classifier $g_n^{(m)}$ on a bootstrap sample of size n. For unstable learners, such as deep decision trees, this procedure reduces variance without increasing bias.

In the multi-class case, the ensemble excess risk satisfies

$$\mathcal{E}(\hat{g}_n) = R(\hat{g}_n) - R^* \leq \frac{1}{M} \sum_{m=1}^{M} \mathcal{E}\left(g_n^{(m)}\right),$$

where $\mathcal{E}\left(g_n^{(m)}\right) = R\left(g_n^{(m)}\right) - R^*$ is the excess risk of the m-th base classifier. Equality holds if the errors of the base classifiers are perfectly correlated, [25], [26].

Intuitively, bagging reduces the impact of uncorrelated base errors by averaging, so the more diverse the base learners, the greater the variance reduction.

D. AdaBoost

Multi-class AdaBoost, or SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss) [27], trains weak learners sequentially on weighted samples. At iteration t, the t-th base learner g_t is trained on the current weighted distribution of the data, and its importance is quantified by the weight

$$\alpha_t = \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right) + \log(C-1),$$

where ϵ_t is the weighted misclassification error of g_t on the training data, i.e., $\epsilon_t = \sum_{i=1}^{n} w_i^{(t)} I\{g_t(\mathbf{x}^{(i)}) \neq y^{(i)}\}$.

The *weak learning assumption* requires that each base learner satisfies

$$P(g_t(\mathbf{X}) = Y) \geq \frac{1}{C} + \gamma$$

for some $\gamma > 0$, i.e., each weak learner performs slightly better than random guessing on the weighted data it is trained on. Under this assumption, the empirical risk of the ensemble classifier after T iterations satisfies, [28]:

$$R_{\text{emp}}(\hat{g}_n) \leq e^{-2\gamma^2 T}. \tag{1}$$

The inequality (1) shows that, if γ is fixed and positive, the training error decreases exponentially fast as T increases.

Furthermore, if the base learners are consistent and the number of iterations T grows suitably with the training sample size n, the population excess risk converges as

$$\mathcal{E}(\hat{g}_n) = R(\hat{g}_n) - R^* = \mathcal{O}\big(n^{-\beta/(2\beta+d)}\big),$$

for β-smooth decision boundaries in d dimensions [28]. Here, the rate depends on the smoothness β of the class boundaries and the feature dimensionality d, showing that AdaBoost can achieve fast convergence for sufficiently regular problems.

E. Gradient Boosted Trees

Gradient Boosted Trees, [29], build additive models sequentially by minimizing the multinomial deviance. For a given class $c \in \{1, \ldots, C\}$, the posterior probability is obtained via the softmax transformation:

$$\hat{p}_{n,c}(\mathbf{x}) = \frac{\exp(F_T^{(c)}(\mathbf{x}))}{\sum_{k=1}^{C} \exp(F_T^{(k)}(\mathbf{x}))},$$

where $F_T^{(c)}(\mathbf{x})$ is the cumulative logit score for class c after T iterations. Each additive term is a regression tree fit to the negative gradient of the loss function at the current iteration.

With consistent base regression trees, a properly tuned learning rate ν, and T growing suitably with the training sample size n, GBTs are Bayes consistent, achieving the excess risk convergence rate

$$\mathcal{E}(\hat{g}_n) = R(\hat{g}_n) - R^* = \mathcal{O}\big(n^{-\beta/(2\beta+d)}\big)$$

for β-smooth conditional distributions $P(y|\mathbf{x})$.

IV. Dataset Description and Preprocessing

A. Dataset Overview

For the empirical study, we use the CIC–MalMem–2022 dataset [30], released by the Canadian Institute for Cybersecurity (CIC). Each record consists of dynamic memory forensics features collected from controlled executions of benign and malicious Windows executables. The release provides only *numeric* attributes (e.g., counts and statistics derived from processes, DLLs, handles, and network artifacts) that capture runtime behavior.

B. Data Preprocessing

We focus on a **4-class** setting by grouping the original family labels using the prefix of the `Category` field:

Benign, Ransomware, Spyware, Trojan.

Concretely, values like `Ransomware-Ako` are mapped to `Ransomware`; any rows whose prefix is not one of these four classes are removed. This yields a clean four-way classification problem for fine-grained malware identification. All experiments use a single, deterministic pipeline applied identically across models. We operate only on numeric attributes and perform the following steps (fit on the training split and applied to validation/test to avoid leakage):

1) **Sanitization.** Coerce non-finite values by replacing $\pm\infty$ with `NaN`, then impute missing entries with the feature-wise median (computed on training data).
2) **Redundancy pruning.** Remove zero-variance (constant) features based on the training split.

3) **Standardization.** Apply z-score scaling (zero mean, unit variance) using training statistics and reuse the learned transform on validation/test.

Standardization improves numerical stability for margin and gradient-based learners. Tree ensembles are invariant to monotone rescalings, so their split criteria are unaffected in theory, [31]. All models are trained on the preprocessed training set, with the same transforms applied to validation and test data.

C. Evaluation Protocol

We adopt a stratified **hold-out** protocol with a fixed split of **70%/10%/20%** into training/validation/test sets, respectively. The validation split is used only for simple per-class threshold selection (grid over probability cutoffs) when explicitly noted. Unless stated otherwise, reported numbers use the default argmax decision rule. All models are trained on the training portion and evaluated once on the held-out test set.

The following ensemble classifiers are evaluated: *Random Forest* (RF), *Rotation Forest* (implemented via the `aeon` library), *Bagging* with decision trees, *AdaBoost* (discrete SAMME with decision stumps), and *HistGradientBoosting* (scikit-learn's gradient-boosted trees). Performances are reported with **macro-averaged** metrics for the 4-class task: Accuracy, Precision, Recall, and F1, together with macro ROC–AUC (one-vs-rest) and macro PR–AUC. To avoid optimistic error estimates, we keep model/threshold selection off the test set, as using cross-validation simultaneously for selection and evaluation is known to bias performance, [32].

V. Experimental Results and Analysis

Following the setup in Sec. IV, we present a consolidated evaluation of the ensembles on the four-class problem. We first compare aggregate performance, then drill down into per-class behavior and confusion structure, examine learning curves to test bias–variance predictions, assess calibration, and conclude with a permutation-based feature importance analysis. Throughout, we keep the operating point and averaging conventions from Sec. IV.

A. 4-Class Results

Table I compares the ensembles introduced in Sec. III. Two patterns stand out. First, variance–reducing methods lead on macro–F1: Bagging is best (F1 $= 0.815$), with Rotation Forest (0.811) and GBT (0.810) essentially tied and RF slightly behind (0.807). Second, ranking quality is uniformly high for Bagging/GBT/RF/Rotation Forest (ROC–AUC ≈ 0.969–0.971, PR–AUC ≈ 0.869–0.880), whereas AdaBoost lags.

These results mirror the theory: at this sample size, variance control benefits Bagging and the tree ensembles, while GBT's bias reduction keeps it competitive (second-best in ranking metrics). AdaBoost underperforms with the chosen weak learners, consistent with its higher bias. Absolute gaps among the top four are small (within ≈ 0.8 percentage points in macro–F1), and overall accuracy follows the same pattern.

TABLE I
4-CLASS TEST PERFORMANCE METRICS.

Model	ACC	Prec	Rec	F1	ROC–AUC	PR–AUC
Random Forest	0.872	0.807	0.807	0.807	0.969	0.869
Rotation Forest	0.874	0.811	0.811	0.811	0.969	0.872
Bagging	**0.877**	**0.815**	**0.815**	**0.815**	**0.971**	**0.880**
AdaBoost (SAMME)	0.738	0.607	0.608	0.603	0.890	0.598
GBT (HistGradBoost)	0.874	0.810	0.810	0.810	0.970	0.877

B. Per-class Analysis and Error Patterns

Table II reports per-class Precision/Recall/F1 for the top model, i.e., Bagging. The Support column indicates how many samples there are of each class. *Benign* is essentially perfect, confirming that the main difficulty lies in distinguishing *malware families* rather than benign vs. malware. Values are rounded to 3 decimals.

TABLE II
PER-CLASS METRICS FOR **BAGGING** ON THE HELD-OUT TEST SPLIT.

Class	Precision	Recall	F1	Support
Benign	1.000	1.000	1.000	5860
Ransomware	0.749	0.732	0.740	1958
Spyware	0.775	0.808	0.791	2004
Trojan	0.736	0.721	0.729	1898

In Figure 1, the normalized confusion matrix is presented. We can see that the most residual errors occur among Ransomware, Spyware, and Trojan. In particular, Ransomware is correctly identified 73% of the time, with confusion to Spyware (13%) and Trojan (14%). Trojan is correct 72%, with confusions to Ransomware (17%) and Spyware (11%). The near-zero off-diagonal mass involving Benign indicates very low false alarms, which is operationally desirable.

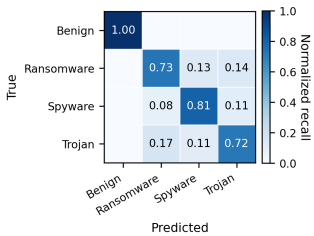


Fig. 1. Confusion matrix for **Bagging** (rows normalized by true class).

As we can see, off-diagonal mass concentrates among malware families. These patterns align with the theory: variance-reducing ensembles (Bagging/RF) carve a high-margin boundary that cleanly separates Benign from malware, while residual excess risk is dominated by family-level overlaps (Ransomware/Trojan/Spyware). If higher recall on specific malware families is required, per-class probability thresholds or class-weighted training can shift the operating point with minimal impact on ranking quality.

C. Learning Curves and Theory Consistency

Figure 2 plots macro PR–AUC as the training size increases for a variance–reducing ensemble (RF) and a bias–reducing ensemble (GBT). On our data, *GBT is consistently, but only slightly, ahead across all training sizes*. Both curves rise smoothly and approach the same regime near the full training set (PR–AUC ≈ 0.88), indicating broadly similar asymptotic behavior.

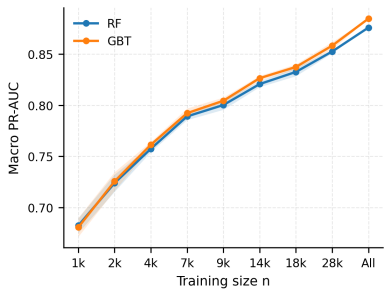


Fig. 2. Learning curves for RF and GBT on the held-out test set (mean $\pm$ sd over repeated stratified subsamples).

The shape of the curves aligns with the bias–variance analysis in our theory section. GBT enjoys a small advantage even at $n = 1$k, consistent with its lower functional bias from additive trees with gradient updates and regularization. RF reduces variance effectively through bagging and feature subsampling, but retains slightly more bias on this task. The near-parallel rise of both curves as n increases reflects decreasing variance and a gradual transition toward a bias-limited regime. At the largest training size the methods deliver comparable ranking quality (PR–AUC ≈ 0.88). Practically, this suggests: (i) RF remains a strong, data–efficient baseline with simple tuning; (ii) when careful regularization and early stopping are used, GBT can provide a small, persistent lift across data scales; and (iii) beyond $\sim$30k $-$ 40k samples, further gains likely require richer features rather than simply more of the same data.

D. Calibration and Operating Point

We assess probability quality using a multiclass reliability diagram. For each test example, we take the winning-class probability and compare it to empirical accuracy in confidence bins. Figure 3 shows that all ensembles are close to the identity line and therefore well calibrated on this task. The measured Expected Calibration Errors (ECE) are small (Bagging $= 0.010$, RF $= 0.013$, GBT $= 0.007$), with GBT slightly better overall. Minor deviations around mid-probability bins indicate mild under-confidence, but there is no systematic over-confidence at high probabilities.

These results explain why the default argmax decision rule already performs near-optimally for macro metrics in our protocol and why threshold tuning yielded negligible macro-F1 changes (Sec. III). For deployments with asymmetric costs or family-specific recall targets, we recommend: (i) apply

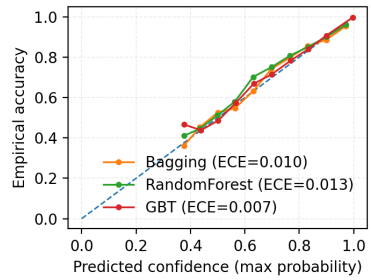


Fig. 3. Reliability diagram (max-probability cali bration) on the test set.

post-hoc calibration on the validation split (e.g., temperature scaling or isotonic regression) and (ii) set per-class thresholds using the calibrated probabilities to maximize a task-relevant objective (e.g., macro F_β or expected cost under class priors). Calibration adjusts probability magnitudes but preserves ranking, so ROC–AUC and PR–AUC are unaffected while operating-point choices become better justified.

E. Permutation Feature Importance

To provide operational insight, we report permutation importance on the test set. Figure 4 lists the top-10 features for Bagging and GBT. The rankings largely agree, indicating stable signals across ensembles. Such attributes are prime candidates for lightweight runtime monitoring.

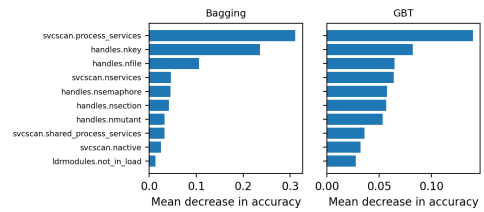


Fig. 4. Top-10 permutation importances on the test split for Bagging (left) and GBT (right).

The most influential features (measured by mean decrease in accuracy) are dominated by memory forensics signals tied to Windows service activity and kernel object usage. In particular, both ensembles highlight `svcscan.process_services` and `svcscan.nservices`, which reflect the number of services detected in memory, together with counts of open registry keys, files, and semaphores recorded under `handles.*`. This aligns with common malware behavior, where malicious executables install or manipulate services and proliferate system handles during execution. Bagging assigns especially high importance to the leading service-related features, suggesting stronger reliance on a small set of high-variance predictors. By contrast, GBT distributes importance more evenly across its top features, reflecting its additive bias–reduction mechanism.

The broad agreement between models on the most informative attributes supports the robustness of these signals. From a practical standpoint, the analysis indicates that (i) detectors should prioritize instrumentation around service creation and handle proliferation, and (ii) feature engineering that captures fine-grained dynamics of service and handle activity is likely to yield further gains. Finally, we note that permutation importance can be attenuated by correlated predictors; repeated shuffles and stratified subsamples mitigate, but do not eliminate, this effect.

VI. Conclusion

This paper combined theoretical guarantees for ensemble classifiers with an empirical study on CIC–MalMem–2022 in a four–class setting (Benign, Ransomware, Spyware, Trojan). The theory in Sections I–III explains how variance–reducing ensembles (Bagging/Random Forest) and bias–reducing ensembles (boosting) approach the Bayes rule under mild conditions, with rates driven by sample size, smoothness of $P(y|\mathbf{x})$, and ensemble diversity. Our experiments (Section V) corroborate these predictions.

On the held-out test split, *Bagging* achieved the best overall performance (macro F1 $= 0.815$, PR–AUC $= 0.880$, ROC–AUC $= 0.971$), while *Random Forest* and *HistGradientBoosting* were within one percentage point across macro metrics (Table I). *Rotation Forest* matched RF closely, and *AdaBoost (SAMME)* underperformed with the chosen weak learners. Per-class analysis showed that Benign is almost perfectly separated, while residual errors concentrate among malware families (Ransomware/Spyware/Trojan), e.g., recalls of 0.732 and 0.721 for Ransomware and Trojan, respectively (Table II, Fig. 1). Learning curves (Fig. 2) reveal RF's advantage at small n and earlier saturation, whereas GBT closes the gap as data grow—exactly the bias–variance trade-off anticipated by our excess-risk discussion. Reliability diagrams (Fig. 3) indicate low calibration error (ECE ≤ 0.013), explaining why simple argmax decisions were already near-optimal; threshold tuning produced negligible macro-F1 gains. Finally, permutation importances (Fig. 4) consistently elevate service and handle activity features, aligning with known malware behaviors and offering actionable signals for instrumentation.

Practical recommendations.

- *Default choice.* Start with Bagging or Random Forest for strong, data-efficient baselines; add HistGradientBoosting when more data or bias reduction is needed.
- *Operating point.* Use argmax as a sensible default; for family-specific recall targets, calibrate on the validation split (temperature or isotonic) and set per-class thresholds against a task-relevant objective (e.g., macro F_β or expected cost).
- *Features to monitor.* Prioritize telemetry around Windows service creation/management and handle proliferation (keys/files/semaphores), which dominate permutation importance across models.
- *When data scale.* Expect diminishing returns in PR–AUC beyond $\sim$30–40k training samples; larger gains will likely require richer features (e.g., temporal dynamics of memory artifacts) rather than simply more of the same data.

Limitations and future work. Results are for a single dataset and a fixed 70/10/20 split with modest tuning; broader validation (cross-dataset, temporal splits, and drifted test periods) would strengthen external validity. More systematic hyperparameter search, cost-sensitive training, and robust/sequential inference (e.g., concept-drift handling, streaming updates) are promising directions. Finally, exploring representation learning for dynamic memory sequences and model ensembling with complementary signals (e.g., file system or network flow features) may push beyond the performance plateau indicated by our learning curves.

In summary, the empirical behavior of the evaluated ensembles reflects their theoretical guarantees: RF and Bagging provide stable, high-margin decision boundaries with excellent calibration, while boosted trees narrow the gap as data grow. The combination of principled operating-point selection and feature-level insights yields practical guidance for deploying reliable malware family detectors in operational settings.

References

[1] T. G. Dietterich, "Ensemble methods in machine learning," in Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg, pp. 1-15, 2000, doi: 10.1007/3-540-45014-9_1.

[2] M. A. Azad, S. Islam, D. M. Farid, and S. Shatabda, "Layered Ensemble Learning for Effective Binary Classification," in Emerging Technologies in Data Mining and Information Security. Lecture Notes in Networks and Systems, J.M.R.S. Tavares, S. Chakrabarti, A. Bhattacharya, S. Ghatak, Ed., Springer Singapore, vol. 164, May 2021, pp. 1–9, doi: 10.1007/978-981-15-9774-9_1.

[3] Q. An, S. Huang, Y. Han, and Y. Zhu, "Ensemble Learning Method for Classification: Integrating Data Envelopment Analysis with Machine Learning," Comput. Ind. Eng., vol. 169, p. 106739, Dec. 2024, doi: 10.1016/j.cor.2024.106739.

[4] S. Andonov, J. Dobreva, L. Lumburovska, S. Pavlov, V. Dimitrova, and A. Popovska-Mitrovikj, "Application of Machine Learning in DES Cryptanalysis," in Web Proceedings of 12th ICT Innovations Conference, 2020.

[5] M. Gjorgjievska Perusheska, V. Dimitrova, A. Popovska-Mitrovikj, and S. Andonov, "Application of machine learning in cryptanalysis concerning algorithms from symmetric cryptography," in Intelligent Computing. Lecture Notes in Networks and Systems, K. Arai, Ed., Springer, Cham., vol. 285, July 2021, pp. 885–903, doi:10.1007/978-3-030-80129-8_59.

[6] S. M. A. H. Bukhari, W. Afandi, M. U. S. Khan, T. Maqsood, M. B. Qureshi, M. A. B. Fayyaz, and R. Nawaz, "E-Ensemble: A Novel Ensemble Classifier for Encrypted Video Identification," Electronics, vol. 11, no. 24, p. 4076, Dec. 2022, doi: 10.3390/electronics11244076.

[7] A. Çetin and S. Öztürk, "Comprehensive Exploration of Ensemble Machine Learning Techniques for IoT Cybersecurity Across Multi-Class and Binary Classification Tasks," J. Fut. Artif. Intell. Tech., vol. 1, no. 4, pp. 371–384, Feb. 2025, doi: 10.62411/faith.3048-3719-51.

[8] D. Dudzik, J. Nalepa, and M. Kawulok, "Ensembles of Evolutionarily-Constructed Support Vector Machine Cascades," KBS, vol. 288, p. 111490, Mar. 2024, doi: 10.1016/j.knosys.2024.111490.

[9] K. Fawagreh, M. M. Gaber, and E. Elyan, "Random forests: from early developments to recent advancements," Syst. Sci. Control. Eng., vol. 2, no. 1, pp. 602–609, Oct. 2014, doi: 10.1080/21642583.2014.956265.

[10] W. Feng, W. Huang, and J. Ren, "Class Imbalance Ensemble Learning Based on the Margin Theory," Appl. Sci., vol. 8, no. 5, p. 815, May 2018, doi: 10.3390/app8050815.

[11] D. Rani, N. S. Gill, P. Gulia, and J. M. Chatterjee, "An Ensemble-Based Multiclass Classifier for Intrusion Detection Using Internet of Things," Comput. Intel. Neurosc., 1668676, May 2022, doi: 10.1155/2022/1668676.

[12] G. Haralabopoulos, I. Anagnostopoulos, and D. McAuley, "Ensemble Deep Learning for Multilabel Binary Classification of User-Generated Content," Algorithms, vol. 13, no. 4, p. 83, Apr. 2020, doi: 10.3390/a13040083.
[13] M. Jamil, H. Mihajloska Trpcheska, A. Popovska-Mitrovikj, V. Dimitrova, and R. Creutzburg, "Advancing image spam detection: Evaluating machine learning models through comparative analysis," Appl. Sci, vol. 15, no. 11, p. 6158, May 2025, doi: 10.3390/app15116158.
[14] J. Jiang, and Y. Atif, "A Selective Ensemble Model for Cognitive Cybersecurity Analysis," J. Netw. Comput. Appl., vol. 193, p. 103210, Nov. 2021, doi: 10.1016/j.jnca.2021.103210.
[15] S. Sikdar and M. Kule, "Intelligent Identification of Cryptographic Ciphers using Machine Learning Techniques," IJISA, vol. 16, no. 6, pp. 20–39, Dec. 2024, doi: 10.5815/ijisa.2024.06.02.
[16] B. D. Kim, V. A. Vasudevan, R. G. L. D'Oliveira, A. Cohen, T. Stahlbuhk, and M. Médard, "Cryptanalysis via machine learning based information theoretic metrics," arXiv preprint arXiv:2501.15076, Jan. 2025. [Online]. Available: https://arxiv.org/abs/2501.15076.
[17] V. Kungurtsev, A. Cobb, T. Javidi, and B. Jalaian, "Decentralized Bayesian learning with Metropolis-adjusted Hamiltonian Monte Carlo," Mach. Learn., vol. 112, pp. 2791–2819, June 2023, doi: 10.1007/s10994-023-06345-6.
[18] M. Li, R. Zhang, and K. Liu, "A New Ensemble Learning Algorithm Combined with Causal Analysis for Bayesian Network Structural Learning," Symmetry, vol. 12, no. 12, p. 2054, Dec. 2020, doi: 10.3390/sym12122054.
[19] C. J. Stone, "Consistent nonparametric regression," Ann. Stat., vol. 5, no. 4, pp. 595–620, July 1977, doi: 10.1214/aos/1176343886.
[20] G. Timár and G. Kovács, "The conditioning bias in binary decision trees and random forests and its elimination," arXiv preprint arXiv:2312.10708, Dec. 2023. [Online]. Available: https://arxiv.org/abs/2312.10708.
[21] K. Zhao, L. Li, Z. Chen, R. Sun, G. Yuan, and J. Li, "A New Multi-classifier Ensemble Algorithm Based on D-S Evidence Theory," Neural Process. Lett., vol. 54, no. 3, pp. 5005–5021, Dec. 2022, doi: 10.1007/s11063-022-10845-2.
[22] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5-32, 2001, doi: 10.1023/A:1010933404324.
[23] E. Scornet, G. Biau, and J.-P. Vert, "Consistency of random forests," Ann. Stat., vol. 43, no. 4, pp. 1716–1741, Aug. 2015, doi: 10.1214/15-AOS1321.
[24] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 10, pp. 1619-1630, 2006, doi: 10.1109/TPAMI.2006.211.
[25] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," J. Am. Stat. Assoc., vol. 101, no. 473, pp. 138–156, 2006, doi: 10.1198/016214505000000907.
[26] L. Breiman, "Bagging predictors," Mach. Learn., vol. 24, no. 2, pp. 123-140, 1996, doi: 10.1007/BF00058655.
[27] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class AdaBoost," SII, vol. 2, no. 3, pp. 349-360, 2009, doi: /10.4310/SII.2009.v2.n3.a8.
[28] Y. Freund, and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119-139, 1997, doi: 10.1006/jcss.1997.1504.
[29] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Ann. Stat., vol. 29, no. 5, pp. 1189-1232, 2001.
[30] Canadian Institute for Cybersecurity, "CIC-MalMem-2022 Dataset," University of New Brunswick, 2022. [Online]. Available: https://www.unb.ca/cic/datasets/malmem-2022.html
[31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.
[32] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," BMC Bioinform., vol. 7, no. 91, 2006, doi: 10.1186/1471-2105-7-91.

Evaluating Multilingual Language Models for Abusive Content Detection: A Comparative Study Across Diverse Social Media Platforms

Mahnoor Jamil*
Faculty of Computer Science and Engineering
mahnoor.jamil@students.finki.ukim.mk
Ss. Cyril and Methodius University
Skopje 1000, North Macedonia

Ivan Chorbev
Faculty of Computer Science and Engineering
ivan.chorbev@finki.ukim.mk
Ss. Cyril and Methodius University
Skopje 1000, North Macedonia

Hasan Dağ
Department of Management Information Systems
hasan.dag@khas.edu.tr
Kadir Has University
Istanbul 34083, Turkiye

Vesna Dimitrova
Faculty of Computer Science and Engineering
vesna.dimitrova@finki.ukim.mk
Ss. Cyril and Methodius University
Skopje 1000, North Macedonia

***Abstract*—The proliferation of abusive content on social media across linguistically diverse regions presents a formidable challenge for automated content moderation systems. Multilingual Language Models (MLLMs), particularly transformer-based architectures, offer scalable solutions to this problem. However, evaluating their effectiveness across varied languages, dialects, and social platforms remains underexplored. This study investigates the performance, generalizability, and limitations of state-of-the-art MLLMs, including BERT, XLM-RoBERTa, mBERT, and their hybrid variants, for abusive content detection across multilingual and code-mixed datasets. We analyze over ten empirical studies covering between 3 and 13 languages, focusing on Indic and European languages, including Romanized and script-mixed forms. Our review highlights key architectural strategies such as transfer learning, lexicon integration, and ensemble modeling. Results indicate that models enhanced with social context or emoji embeddings outperform baseline transformers, achieving macro F1-scores as high as 0.91. We further assess platform-specific challenges and the scalability of these models in low-resource settings. This comparative study provides insights into model adaptability and effectiveness in real-world moderation systems and suggests directions for future research in multilingual content moderation.**

***Index Terms*—Multilingual Language Models, Social Media Platforms, Abusive Content Detection, Transfer Learning, Ensemble Modeling**

I. Introduction

The global nature of online communication platforms, such as Twitter, Facebook, and YouTube, has intensified the issue of abusive language, harassment, and hate speech in diverse linguistic contexts. Traditional moderation tools, which rely heavily on monolingual models and manually curated rules, often fail to detect subtle forms of abuse—especially in multilingual, code-mixed, or low-resource settings. The emergence of Multilingual Language Models (MLLMs) has introduced promising approaches to cross-lingual and multilingual abusive content detection, but there remains a lack of comparative evaluations to determine their performance across different social media contexts.

This study critically examines the recent progress in applying MLLMs to abusive content detection. We present a comprehensive analysis of the methods, architectures, evaluation metrics, and performance outcomes reported in recent literature. Our goal is to offer an informed perspective on how various

multilingual approaches perform under different linguistic and platform-specific constraints.

II. Literature Review

A. Transformer-Based Models in Multilingual Abusive Detection

Transformer-based models like mBERT, XLM-RoBERTa, and MuRIL have demonstrated strong baseline performance in multilingual abusive content detection. Studies such as [6] and [10] employed variants like AbuseXLM-R and MuRIL across 12 Indic languages, reporting F1-scores exceeding 91% in some cases. These results suggest that pretraining on large multilingual corpora enables models to generalize across languages, including low-resource dialects. The study [9] extended XLM-RoBERTa with Bi-GRU and emoji embeddings, achieving an F1-score of 0.88 and an AUC of 0.94 across 13 Indic code-mixed languages. Their work emphasized the importance of incorporating universal visual elements like emojis to enhance performance in zero-shot learning scenarios. The study [8] compared CNN, Bi-LSTM, and BERT-based architectures across five languages and found that fine-tuned BERT achieved a macro F1-score of 0.86 on code-mixed/script-mixed text. These findings reinforce the superiority of transformer-based models over traditional neural networks for complex multilingual inputs.

B. Hybrid and Ensemble Approaches

Hybrid models that integrate different neural architectures and linguistic features are gaining popularity. The study [3] introduced a hybrid of LSTM and multilingual lexicons (HurtLex), enhancing recall in cross-lingual abusive content classification. Their joint-learning model performed well on English, German, Italian, and Spanish datasets. Similarly, the study [6] combined mBERT, XLM-R, and MuRIL in an ensemble framework enriched with social context features such as user history and post polarity.

This model achieved F1-scores above 95% on certain Indic datasets, demonstrating that context-aware architectures improve detection accuracy. Multimodal approaches also emerged, with [7] integrating audio, emotion, and text modalities for abuse detection across ten unspecified languages. While specific metrics were not reported, relative improvements of up to 5.2% over text-only models were observed, indicating potential for further exploration.

C. Language and Script Diversity

A prominent challenge in multilingual moderation is handling code-mixed, romanized, and low-resource languages. For instance, [1] reported macro F1-scores as low as 0.46 for Tamil-English, highlighting the difficulty of modeling Dravidian code-mixed text. To counteract such limitations, researchers used transfer learning, oversampling, and language-specific pretraining (e.g., MuRIL for Indic languages). The study [4] tackled zero-shot detection using a hybrid emoji-based masked language model (HE-MLM). Their approach utilized emojis as cross-lingual markers, yielding improved macro F1-scores in English, Italian, and German, demonstrating the value of non-textual features in cross-lingual generalization. Research by [2] focused on English, Hindi, and Hinglish using BERT and claimed state-of-the-art results, though the lack of detailed metrics limits interpretability.

III. Research Selection and Inclusion Criteria

This paper reviewed 10 empirical studies published between 2019 and 2024, selected using the following inclusion criteria:

- Relevance: Each study focused on the detection of abusive, offensive, or toxic language in social media or public user-generated content.
- Multilingual Scope: Included more than three languages, with at least one study using 13 languages, and support for code-mixed, low-resource, or romanized scripts.
- Technical Foundation: Use of machine learning or deep learning methods, with a preference for transformer-based architectures.
- Empirical Reporting: Reporting of evaluation metrics such as F1-score, accuracy, precision/recall, or AUC.
- Platform Context: Datasets must originate from real-world online platforms like ShareChat, Twitter, Moj, Reddit, or Facebook.

IV. Data Sources

Datasets were sourced from diverse platforms such as ShareChat, Moj, Twitter, and YouTube comments. While [10] and [6] emphasized large-scale platform-specific datasets with millions of examples, many other studies used relatively smaller corpora, affecting scalability and generalization. Notably, only a few studies systematically analyzed how platform characteristics (e.g., post length, emoji usage, anonymity) influence model performance. This highlights a research gap in assessing the robustness of MLLMs across platforms.

V. Methodology

This section outlines the systematic methodology adopted to evaluate the performance, linguistic scope, and design of multilingual language models (MLLMs) used for abusive content detection across various studies. This research approach consists of three stages: study selection, data extraction, and methodology-model mapping.

A. Data Extraction and Dimensions of Analysis

For each selected study, several key analytical dimensions were extracted to ensure a comprehensive comparative evaluation. First, the model architecture was identified to determine whether the approach employed a pure transformer (e.g., BERT, XLM-R), a hybrid structure (e.g., BiGRU combined with a transformer), or an ensemble-based design. The methodological strategy was also scrutinized, focusing on the use of techniques such as transfer learning, zero- or few-shot learning, lexicon integration, oversampling, and multimodal fusion.

Additionally, linguistic coverage was assessed in terms of the number and nature of supported languages, including Indic and European languages, as well as code-mixed dialects. Dataset characteristics were considered with attention to the source, volume, class distribution, and domain specificity—such as whether the data consisted of user comments or tweets. Performance evaluation relied on widely adopted metrics including macro F1-score, accuracy, AUC, and qualitative assessments. Finally, special attention was given to innovative contributions, including techniques for improving generalization, handling class imbalance, and facilitating integration into real-world platforms.

B. Methodology–Model Mapping

To better understand the current landscape of multilingual and code-mixed abusive content detection, the research conducted a structured comparison of prominent studies in this domain. Tables I and II illustrate a mapping between selected studies, their respective models, and the methodologies adopted. Table I summarizes the variety of deep learning and transformer-based models leveraged across recent research, ranging from traditional architectures such as CNN and LSTM to advanced pretrained language models like XLM-R, MuRIL, and BERT. Meanwhile, Table II outlines the diverse methodological strategies employed—such as ensemble learning, zero-shot transfer, hybrid lexical approaches, and multimodal fusion—highlighting how researchers have adapted model design to tackle linguistic variability and data scarcity in low-resource settings. This mapping not only provides a comparative view but also guides the design choices for our proposed system by identifying effective model-methodology combinations from existing literature.

C. Observations from Methodology Trends

From the above analysis, several methodological trends emerge: Transformer Models Dominate: All top-performing models rely on transformer-based architectures (e.g., XLM-R, mBERT, MuRIL), either directly or in hybrid/ensemble configurations. Transfer Learning Is Ubiquitous: 7 out of 10 studies utilized transfer learning to adapt high-resource language models to code-mixed or low-resource contexts. Lexicon Integration Is Limited but Valuable: Only one study [3] used lexicon-assisted classification, yet reported significant improvements in recall. Hybrid and Multimodal Models Excel in Edge Cases: Models integrating emoji embeddings or audio-text fusion proved highly effective in zero-shot or sparse-data environments. Contextual Features Are Gaining Ground: [6]'s success in using user-level metadata suggests a shift toward social-context-aware architectures.

TABLE I
STUDY AND MODEL(S) USED

Study	Model(s)
[9]	XLM-R + BiGRU + Emoji Embeddings
[6]	MuRIL, XLM-R, mBERT + Ensemble
[8]	CNN, Bi-LSTM, BERT
[3]	LSVC, LSTM, HurtLex
[10]	AbuseXLM-R, XLM-R, MuRIL
[5]	mBERT, MuRIL
[4]	XLM (HE-MLM)
[2]	BERT
[7]	MADA (Multimodal DL)
[1]	LinearSVC, BERT

TABLE II
STUDY AND METHODOLOGY USED

Study	Methodology Used
[9]	Hybrid architecture, Transfer Learning, Emoji Embedding
[6]	Ensemble Learning, Social Context Modeling, Cross-lingual Embeddings
[8]	Comparative DL Baselines, Fine-tuned BERT
[3]	Hybrid Lexicon + Embeddings, Joint Learning
[10]	Pretraining on Indic corpus, Transformer Comparison
[5]	Zero-shot / Few-shot Transfer Learning
[4]	Emoji-Augmented Masked Language Model
[2]	Transformer Fine-tuning
[7]	Audio-Text Fusion, Multimodal Fusion
[1]	Oversampling, Transfer Learning, Code-Mixed Handling

VI. RESULTS

The evaluated studies display a range of architectures and innovations with performance metrics such as Macro F1-score and AUC that underline their effectiveness. The study [9] introduced an emoji-aware hybrid model combining XLM-R, BiGRU, and emoji embeddings on ShareChat data in 13 Indic languages, achieving an impressive F1 score of 0.88 and an AUC of 0.94. Research conducted by [6] leveraged a powerful ensemble of MuRIL, XLM-R, and mBERT enriched with social context features across 12 Indic languages and datasets like SCIDN and MACI, reporting strong F1 scores ranging from 91% to 95%. Furthermore, [8] evaluated multiple models—CNN, Bi-LSTM, and BERT—on code-mixed/script-mixed Twitter-like corpora in five languages, with reported F1 scores of 0.79 for monolingual and 0.86 for mixed settings.

The study [3] focused on lexicon-assisted joint learning using LSVC, LSTM, and the HurtLex lexicon across English, Italian, German, and Spanish from Reddit and Twitter, achieving an

approximate F1 of 0.70 for the positive class. Moreover, [10] used AbuseXLM-R, MuRIL, and XLM-R over Indic platforms such as ShareChat and Moj, proposing a custom transformer-based model; however, performance metrics were not reported. The study [5] combined mBERT and MuRIL with transfer learning over eight Indic languages in Hindi-English corpora and demonstrated solid results with a Macro F1 between 0.84 and 0.86, highlighting the benefits of zero- and few-shot learning.

Research conducted by [4] presented a hybrid emoji-based masked language model (HE-MLM) for four European languages (EN, DE, IT, ES) using Twitter data and reported improvements in zero-shot performance, particularly on the Macro F1 score. Furthermore, [2] applied BERT to English, Hindi, and Hinglish text from Twitter and YouTube, targeting code-mixed Indian content and claimed state-of-the-art (SOTA) performance, though exact metrics were not specified. Moreover, [7] introduced a multimodal MADA architecture that processes audio, emotion, and text from TikTok-like data in ten unspecified languages, gaining approximately 5.2% improvement over audio-only approaches. Lastly, [1] employed a combination of LinearSVC, BERT, and oversampling to address data imbalance in three Dravidian languages using Facebook comments, reporting F1 scores ranging between 0.46 and 0.74.

A. Language and Model Performance Correlation

Across all studies, transformer-based models consistently outperform traditional architectures (e.g., CNNs, SVMs) on multilingual tasks. Notably: [9] achieved top scores using an XLM-RoBERTa model fused with BiGRU and emoji embeddings, tailored for Indic code-mixed text. Moreover, [6] reported the highest scores using ensemble models incorporating user-level features such as post polarity and historical context. Lastly, [8] showed that even basic BERT fine-tuning surpassed CNN/Bi-LSTM baselines, especially on script-mixed datasets. Performance varies notably by language type. Dravidian languages (Tamil, Telugu) and code-mixed dialects like Hinglish remain more challenging, often yielding F1-scores under 0.60 without language-specific adaptations.

B. Architectural Innovation

Studies with hybrid models—integrating emojis, social features, or attention mechanisms—tend to report superior results. Emoji tokens ([4], [9]) appear to serve as cross-lingual indicators, aiding generalization in low-resource or zero-shot settings. Social context integration [6] improves performance in context-rich platforms by understanding user behavior and post patterns. Multimodal efforts [7] show promising potential but lack fine-grained benchmarks due to unavailable language breakdowns.

C. Platform and Dataset Impact

Performance is heavily influenced by platform-specific characteristics, including text length, emotive cues, and community behavior. The studies [10] and [6] leveraged large platform-specific datasets from ShareChat and Moj, which helped improve domain adaptation. In contrast, models trained on general-purpose corpora (e.g., Reddit, Twitter) without contextual metadata struggled to match these results. Only a few studies considered real-time deployment or latency, pointing to a research gap in operational feasibility and system integration.

D. Cross-Lingual and Low-Resource Adaptability

Transfer learning and embedding-based augmentation were particularly effective in low-resource or code-mixed contexts. The research conducted by [5] demonstrated strong macro F1 gains through zero-/few-shot transfer from high-resource to low-resource Indic languages. Moreover, [1] highlighted the benefits of random oversampling for class imbalance but noted persistent performance drops in Tamil-English data. Models like MuRIL and AbuseXLM-R, pre-trained on Indic corpora, show higher adaptability in regional contexts, though not universally optimal.

VII. Discussion

This study highlights both the promise and complexity of deploying multilingual language models (MLLMs) for abusive content detection across diverse social media platforms. Several key

insights emerge from the comparative evaluation. Transformer-based models remain the backbone of current multilingual detection systems due to their superior contextual understanding and pretraining on massive multilingual corpora. XLM-R, mBERT, and MuRIL consistently outperform earlier models, particularly when combined with auxiliary techniques such as recurrent layers (BiGRU), attention mechanisms, or hybrid embeddings. Fine-tuned BERT models showed competitive results even in low-resource settings, especially when enhanced by domain-specific pretraining (e.g., AbuseXLM-R).

The success of ensemble architectures [6] further supports the notion that model diversity—combining predictions from different transformers with metadata—can improve robustness and F1-scores. In contrast, standalone CNNs or SVMs lacked the linguistic adaptability required for cross-lingual or code-mixed detection.

VIII. Conclusion

This paper presents a comprehensive review and comparative analysis of multilingual language models for abusive content detection on social media. The evaluation spans ten prominent studies, incorporating 3–13 languages each and employing diverse architectures such as transformers, hybrids, and ensemble methods. Transformer-based MLLMs (XLM-R, mBERT, MuRIL) are the current standard, achieving high F1-scores across several languages when fine-tuned or enhanced. Hybrid and ensemble models significantly outperform standalone models, particularly in low-resource or code-mixed environments. Social context and emoji embeddings offer performance boosts, especially for informal or platform-specific communication. Despite these advancements, challenges remain in script-mixing, cross-platform adaptation, ethical deployment, and real-time scalability. While the field has made notable progress, there is no universal model yet that robustly performs across all languages and platforms without performance degradation.

References

[1] A. Hegde, Kavya G, Sharal Coelho, and H. Shashirekha. "MUCS@DravidianLangTech2023: Leveraging Learning Models to Identify Abusive Comments in Code-Mixed Dravidian Languages." DRAVIDIANLANGTECH, 2023. https://scholar.google.es/citations?view_op=view_citation&hl=en&user=WypaPZIAAAAJ&citation_for_view=WypaPZIAAAAJ:qjMakFHDy7sC

[2] Aditya Malte, and Pratik Ratadiya. "Multilingual Cyber Abuse Detection Using Advanced Transformer Architecture." IEEE Region 10 Conference, 2019. https://www.researchgate.net/publication/345354469_Multilingual_Cyber_Abuse_Detection_using_Advanced_Transformer_Architecture

[3] E. W. Pamungkas, and V. Patti. "Cross-Domain and Cross-Lingual Abusive Language Detection: A Hybrid Approach with Deep Learning and a Multilingual Lexicon." Annual Meeting of the Association for Computational Linguistics, 2019. https://www.researchgate.net/publication/335781352_Cross-domain_and_Cross-lingual_Abusive_Language_Detection_A_Hybrid_Approach_with_Deep_Learning_and_a_Multilingual_Lexicon

[4] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, and S. Villata. "Hybrid Emoji-Based Masked Language Models for Zero-Shot Abusive Language Detection." Findings, 2020. https://www.semanticscholar.org/paper/Hybrid-Emoji-Based-Masked-Language-Models-for-Corazza-Menini/a37b7c0785693a25a2ef9351ff4c348a727843f7

[5] M. Das, S. Banerjee, and A. Mukherjee. "Data Bootstrapping Approaches to Improve Low Resource Abusive Language Detection for Indic Languages." ACM Conference on Hypertext and Social Media, 2022. https://arxiv.org/abs/2204.12543

[6] M. Z. U. Rehman, S. Mehta, K. Singh, K. Kaushik, and N. Kumar. "User-aware Multilingual Abusive Content Detection in Social Media." Information Processing and Management, 2023. https://www.sciencedirect.com/science/article/abs/pii/S0306457323001875

[7] R. Sharon, H. Shah, D. Mukherjee, and V. Gupta. "Multilingual and Multimodal Abuse Detection." Interspeech, 2022. https://arxiv.org/abs/2204.02263

[8] S. Saumya, A. Kumar, and J. P. Singh. "Filtering Offensive Language from Multilingual Social Media Contents: A Deep Learning Approach." Engineering Applications of Artificial Intelligence, 2024. https://www.sciencedirect.com/science/article/abs/pii/S0952197624003178

[9] V. Bansal, M. Tyagi, R. Sharma, V. Gupta, and Q. Xin. "A Transformer Based Approach for Abuse Detection in Code Mixed Indic Languages." ACM Transactions on Asian and Low-Resource Language Information Processing, 2022. https://dl.acm.org/doi/10.1145/3571818

[10] V. Gupta, S. Roychowdhury, M. Das, S. Banerjee, P. Saha, B. Mathew, H. P. Vanchinathan, and A. Mukherjee. "Multilingual Abusive Comment Detection at Scale for Indic Languages." Neural Information Processing Systems, 2022. https://proceedings.neurips.cc/paper_files/paper/2022/hash/a7c4163b33286261b24c72fd3d1707c9-Abstract-Datasets_and_Benchmarks.html

Evaluation of Privacy-Enhancing Technologies Against Web Tracking

Resul Bedii Gümüş
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, North Macedonia
Email: rbediigumus@gmail.com

Boban Joksimoski
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, North Macedonia
Email:boban.joksimoski@finki.ukim.mk

Tuğçe Ballı
Faculty of Engineering and Natural Sciences
Kadir Has University
Istanbul, Turkey
Email:tugce.balli@khas.edu.tr

***Abstract*—This paper evaluates the effectiveness of Privacy-Enhancing Technologies (PETs) in mitigating modern web tracking. Building on a review of PETs and web measurement frameworks, we conduct large-scale experiments using the T.EX framework on the Tranco Top-1,000 domains. Our study compares default protections in major browsers and the impact of popular extensions. Results show that Brave offers the strongest built-in defenses, while uBlock Origin provides the most consistent reduction in tracker requests, cookies, and JavaScript events. Regional measurements across Germany, North Macedonia, and Turkey further reveal geographic variation in tracking but demonstrate that effective PETs can normalize exposure.**

***Index Terms*—Web tracking, browser fingerprinting, privacy-enhancing technologies, PETs, web measurement, tracking detection**

I. Introduction

The online advertising industry has grown exponentially, driven by the ability to deliver personalized content to users based on the vast amounts of data collected from their browsing behavior. This data collection is often facilitated through cookies and other tracking technologies, enabling advertisers and data brokers to profile individuals and serve targeted advertisements. While these mechanisms offer economic advantages and personalization benefits, they also pose serious privacy concerns. The data collected may persist across websites and over time and be aggregated by third-party trackers or collected in a pool of data that may lead to personally identifiable information being revealed without users' consent. Modern tracking methodologies—such as browser fingerprinting, canvas fingerprinting, and behavioral profiling—pose greater detection and prevention challenges than traditional third-party cookies. Although legal frameworks such as the General Data Protection Regulation and industry-led initiatives such as Google's Privacy Sandbox aim to limit invasive tracking practices, the rapid evolution of tracking technologies continues to outpace regulation.

As the web ecosystem becomes increasingly dominated by surveillance-based advertising, many users have sought protection through a range of Privacy-Enhancing Technologies (PETs). These include popular browser extensions like ad and tracker blockers, privacy-oriented browsers such as Brave or Tor, and network-level solutions like VPNs and DNS-based blocking tools. Others rely on built-in browser settings that restrict third-party cookies or private browsing modes that limit local traceability. Some users opt for manual controls, including cookie consent managers and opt-out interfaces. A large-scale study highlights the diversity in how such tools are adopted and deployed across different user contexts [18]. Despite the broad availability of PETs, many users deploy them with incomplete or inaccurate assumptions about their capabilities. For instance, private browsing modes are commonly misunderstood as comprehensive anti-tracking solutions when in fact they offer no protection against fingerprinting or persistent cross-site surveillance [18]. These misconceptions, combined with the sheer diversity of privacy-enhancing tools and threats against privacy, highlight the importance of continued research into the technical efficacy and usability of PETs.

This study builds on prior work by the author, including the earlier publication *Web Tracking Technologies: Current Research Analysis* presented at the International Conference on Informatics and Information Technologies, which provided a literature review of web tracking techniques. In this follow-up study, the focus shifts from tracking mechanisms to evaluating the defensive capabilities of privacy-enhancing technologies using modern web measurement infrastructure.

The remainder of this paper is structured as follows. Section II reviews related work on privacy-enhancing technologies and web measurement frameworks. Section III describes the methodology, including the experimental setup, browsers and extensions tested, and the metrics used. Section IV presents the results of the measurements. Section V concludes with key findings and directions for future research.

II. Privacy-Enhancing Technologies

Fingerprinting-based tracking has become prominent, primarily due to mounting restrictions on third-party cookies. Privacy-enhancing technologies have increasingly focused on countering stateless tracking techniques. The decline of traditional cookie-based tracking has not rendered those methods obsolete; instead, they have resurfaced in adapted forms, such as using first-party cookies by third-party JavaScript to bypass browser policies, as documented in [4]. Consequently, defenses against cookies remain relevant, but much of the innovation in the PET space has shifted toward more advanced techniques designed to mitigate fingerprinting.

PETs can be grouped by their technical strategies—ranging from filter list-based blocking and heuristic detection to structural analysis and runtime behavior monitoring.

One of the most widely adopted approaches involves filter list-based blocking, where tools like AdBlock Plus, uBlock Origin, Ghostery, and Disconnect rely on curated blacklists (e.g., EasyList, EasyPrivacy) to block known trackers based on script sources, request URLs, or page elements. These tools offer deterministic, straightforward protection and are favored for their ease of use and integration across major browsers. Multiple studies confirm their dominance in real-world usage and academic testing environments [19, 17]. However, their reliance on manually maintained lists leaves them vulnerable to novel or obfuscated tracking behaviors.

To address these limitations, other PETs rely on heuristic and behavior-based detection. Simultaneously, browser-level PETs offer integrated protections. These mechanisms are often built directly into browsers and provide tracking mitigation features without requiring additional extensions [22]. Still, not all PETs are user-friendly: tools such as NoScript or RequestPolicy offer high protection levels but often degrade site functionality, making them less practical for everyday use [19, 3].

The challenge of fingerprinting, a technique that identifies users through subtle, often imperceptible traits like canvas rendering or WebGL configuration, has given rise to PETs that randomize or mask browser attributes. Tools like PriVaricator [20], FPRandom [13], and FPGuard[9] attempt to reduce fingerprint uniqueness by injecting variability or spoofed values into APIs commonly exploited for tracking. Beyond these defenses, a recent line of PETs developed in academic research, such as AdGraph [11] and WebGraph [2], leverages machine learning and structural analysis to detect fingerprinting behaviors. While these tools demonstrate impressive detection precision, they remain mainly in the research domain.[22].

Despite these technical advances, the battle between tracking technologies and privacy defenses remains asymmetric and ongoing. Trackers that become more sophisticated are largely invisible to client-side PETs and highlight the limits of current browser and extension-based defenses [10].

In addition to PETs that are used by end users, Researchers have developed a range of privacy measurement frameworks to analyze web tracking at scale. These tools support automated data collection, browser instrumentation, and tracking detection—playing a critical role in evaluating PETs.

Frameworks typically follow either a client-based or proxy-based architecture. Client-side tools like OpenWPM, Bahrami, Iqbal, and Shafiq [1], and FP-tracer [2] integrate with browsers, mainly Firefox, to record JavaScript execution, network traffic, and DOM events. Recent frameworks, such as the one by Raschke and Cory [21], extend this model to Chromium-based browsers, enabling cross-browser comparisons.

Proxy-based tools like OmniCrawl instead observe web activity externally by injecting JavaScript at runtime. While more flexible across platforms, they offer less visibility into fine-grained browser behaviors [21].

Measurement frameworks can also be distinguished by detection focus. Stateful tracking tools log cookies and storage usage (e.g., OpenWPM, [1], [4]), and stateless tracking detectors monitor fingerprinting vectors like canvas or WebGL.

Structural techniques use AST or graph analysis (e.g., AdGraph [11], WebGraph [2], AST-Trans [25]), though they remain largely experimental due to performance costs [27].

Behavioral analysis tools apply taint tracking or graph learning to script behavior (e.g., WTAGraph, XFP-Recognizer [24]).

Platform support varies. While early tools favored Firefox, newer systems support cross-browser evaluations (e.g., [21]), and some extend to mobile tracking, capturing sensor or geolocation access [22].

III. Methodology

To ensure a representative and reproducible sample of the modern web, we used the Tranco ranking system [14]. Tranco aggregates data from multiple top list providers over a configurable time window, which reduces volatility and yields a more stable view of web traffic. Lists are generated with unique identifiers and timestamps, supporting replication. For this study, we selected the top 1,000 domains from a list created on 20 July 2025, aggregated from Umbrella, Majestic, CrUX, and Cloudflare Radar. Only pay-level domains were retained. The list identifier is published for transparency [15].

For browser coverage, we selected the most popular desktop browsers by global market share between July 2024 and July 2025, based on StatCounter statistics [23]. Chrome, Safari, Edge, and Firefox were the top four browsers during this period. Safari was excluded because our measurements were conducted on Windows, where Safari is no longer supported. In addition to these mainstream browsers, we included Chromium to allow direct comparison with Chrome and to assess differences between the open-source and Google-distributed builds. Brave was added as a representative privacy-focused browser to evaluate how a browser with integrated anti-tracking mechanisms performs compared to others.

The extensions were selected primarily according to their popularity in the Chrome Web Store, using installation statistics compiled by DebugBear [16, 5]. From this dataset, we selected Adblock Plus, uBlock Origin, and AdGuard as widely

adopted blocking tools. Privacy Badger, while not among the most popular, was added to represent a heuristic blocking approach that differs from list-based tools and remained compatible with our measurement process. The scope of extension selection was also shaped by the capabilities of the Transparency Extension (T.EX) framework, which measures the effect of extensions by recording and classifying network requests and JavaScript API calls. This design makes it suitable for evaluating tools that actively block or filter requests, but not for defenses that modify or obfuscate data without preventing requests, such as many fingerprinting countermeasures.

For our analysis, we adopt T.EX as a cross-browser privacy measurement framework [21]. T.EX operates as a browser-native extension compatible with both Chromium and Firefox engines, allowing evaluations under realistic browsing conditions without external automation layers such as Selenium. It relies on standard browser extension APIs and the `webextensions-polyfill` library for compatibility across different implementations. During automated browsing sessions, T.EX opens multiple tabs, navigates to websites from the selected list, waits for a fixed period, and then records activity before closing each tab. All HTTP(S) requests and responses are captured through the `webRequest` interface, while JavaScript API calls are recorded through injected content scripts.

The framework uses Ghostery's filter engine for labeling, drawing on EasyList and EasyPrivacy for advertising and tracking elements and on Disconnect's tracker protection list for third-party data collection and fingerprinting domains [7, 8, 6]. These blocklists are widely used in popular browsers and extensions, ensuring consistency with real-world blocking strategies. All recorded events are stored locally in JSON format. Each crawl generates two datasets: HTTP(S) events with metadata such as URLs, headers, and response codes, and JavaScript events with interface access information. Since the built-in interface provides only limited per-crawl summaries, we exported the raw data and analyzed it using Python scripts.

The metrics used in this study follow the definitions of the original T.EX framework [21]. They capture three main categories: traffic volume and classification (total requests and share of third-party and labeled tracking activity), stateful tracking indicators (presence of cookies in requests and responses), and stateless tracking indicators (JavaScript interface access patterns). This structure enables systematic comparison of baseline browser behavior and the impact of extensions across multiple layers of tracking.

IV. Results

Across default configurations of the tested browsers, Brave consistently produced the lowest levels of advertising- and tracking-related activity. It generated about one quarter of its HTTP requests to known tracking domains, compared to nearly half for Chrome and Chromium, and roughly 40% for Edge and Firefox. At the JavaScript level, Brave recorded about one third of its events as tracking-related, while Chrome, Chromium, and Firefox each exceeded 58%. Most of the additional traffic observed in these browsers beyond Brave's baseline was classified as tracking, showing that the differences in overall request and event volumes are largely driven by trackers.

TABLE I
A&T Requests by Browser

Browser	Total Requests	A&T Requests	Percentage
brave	58258	14915	25.60
chrome	117012	56021	47.88
chromium	112685	52143	46.27
edge	79925	31306	39.17
firefox	77718	32668	42.03

Third-party connections accounted for a substantially smaller share of Brave's traffic (about 45%) compared to Chrome and Chromium (around 60%) and Edge and Firefox (about 56%). This pattern reinforces the observation that Brave suppresses a larger portion of cross-site tracking attempts.

TABLE II
Third-Party HTTP/S Requests by Browser

Browser	Total Requests	TP Requests	% TP
brave	57,491	26,041	45.30
chrome	116,094	71,866	61.90
chromium	111,866	67,506	60.35
edge	79,062	44,574	56.38
firefox	75,199	42,244	56.18

Cookie activity also revealed clear differences. Between 38% and 43% of all requests across browsers transmitted cookies, while 9–11% of responses set them. However, the prevalence of third-party cookies varied sharply. Brave transmitted only a few hundred cookie-bearing third-party requests, compared with tens of thousands in Chrome and Chromium, and several thousand in Edge. Firefox also limited third-party cookies more effectively than the Chromium-based browsers, though not to the same extent as Brave. These results indicate that stateful tracking through cookies remains common across browsers, but its scope differs significantly depending on default protections.

TABLE III
Third-Party HTTP/S Requests Transmitting a Cookie

Browser	TP Requests	With Cookie	Percentage
brave	26041	719	2.76
chrome	71866	18685	26.00
chromium	67506	17566	26.02
edge	44574	3759	8.43
firefox	42244	908	2.15

Overall, the comparisons show a consistent hierarchy: Brave applies the strongest restrictions among default browsers, Edge and Firefox occupy the middle, and Chrome and Chromium expose users to the highest levels of tracker-related requests, events, and cookies.

Comparing Firefox in its default configuration with Firefox equipped with popular privacy extensions shows clear differences in tracking prevention. uBlock Origin consistently

achieved the strongest reductions across all layers. It lowered advertising- and tracking-related requests to about one quarter of all network traffic, cut third-party connections to under half, and reduced JavaScript tracking events to roughly one third of the total. Privacy Badger ranked second, achieving moderate reductions in both network and script activity. AdGuard delivered improvements but at a smaller scale, while AdBlock and Adblock Plus provided little or no benefit, and in some cases allowed more third-party traffic than the default. These tools often prioritize removing visible advertisements rather than actively blocking trackers, and may permit "acceptable ads" or whitelisted domains that still perform tracking. The T.EX framework sometimes classifies extension-related background communications as tracking events, which can inflate counts without reflecting a real increase in exposure. Cookie-related results showed a similar hierarchy, though not always proportional to network reductions. uBlock Origin and Privacy Badger blocked the most third-party and tracker-associated cookies, while AdBlock and Adblock Plus allowed substantially more cookie transmissions and responses than the Firefox default. This divergence indicates that tools focusing primarily on visible advertising may fail to prevent stateful tracking mechanisms.

TABLE IV
A&T HTTP/S REQUESTS BY EXTENSIONS

Extension	Total Requests	A&T Requests	% A&T
Default	77,718	32,668	42.03
AdBlock	76,209	32,344	42.44
Adblock Plus	93,185	40,321	43.27
AdGuard	77,491	31,470	40.61
Privacy Badger	61,252	20,478	33.43
uBlock Origin	60,044	14,884	24.79

TABLE V
A&T JAVASCRIPT EVENTS BY EXTENSIONS

Extension	Total JS Events	A&T JS Events	% A&T JS
Default	552,755	321,812	58.22
AdBlock	638,145	332,599	52.12
Adblock Plus	777,720	445,400	57.27
AdGuard	648,902	354,012	54.56
Privacy Badger	438,481	199,405	45.48
uBlock Origin	330,775	118,172	35.73

Overall, the extension comparison confirms that uBlock Origin offers the most comprehensive protection, followed by Privacy Badger and AdGuard, with AdBlock and Adblock Plus trailing. The findings also illustrate that lowering tracker request volumes does not automatically eliminate cookie-based tracking, highlighting differences in how extensions prioritize blocking strategies.

A. Regional Analysis

To examine how tracking exposure varies across regions, we repeated measurements for Germany, North Macedonia, and Turkey using VPN endpoints and geolocation spoofing [26, 12].

In the regional analysis, we tested fewer browsers and extensions than in the main experiments. To keep the data collection manageable, we limited tests to *Chrome* and *Firefox*: Chrome for its global popularity [23], and Firefox for its distinct rendering engine (Gecko) compared to Chromium-based browsers. For extensions, we focused on *Adblock Plus*, the most widely installed ad-blocker [16], and *uBlock Origin*, which consistently demonstrated the strongest filtering performance in our evaluations. While this narrowed scope excluded other combinations tested in the main study, it was sufficient to achieve the central aim of the regional analysis: to observe how browsers and privacy-enhancing extensions interact with regional tracking behaviors, and to determine whether strong tools can normalize exposure across different environments.

While the percentages of third-party and advertising- and tracking-related requests were similar across countries, absolute volumes were highest in Turkey. Chrome from Turkey produced more than 70,000 third-party requests and over 500,000 advertising- and tracking-related JavaScript events, far exceeding the corresponding values from Germany. Macedonia often fell between Germany and Turkey, but Firefox in Macedonia recorded especially high tracking densities, with shares of advertising- and tracking-related requests above 47% and JavaScript events above 61%.

TABLE VI
A&T HTTP/S REQUESTS BY COUNTRY AND BROWSER

Country	Browser	Count (A&T / Total)	% A&T
Turkey	Chrome	56,021 / 117,012	47.88%
Turkey	Firefox	32,668 / 77,718	42.03%
Germany	Chrome	35,718 / 100,972	35.37%
Germany	Firefox	35,026 / 89,442	39.16%
Macedonia	Chrome	50,290 / 115,182	43.66%
Macedonia	Firefox	48,946 / 103,851	47.13%

Cookie activity also displayed regional variation. Chrome consistently transmitted the most cookies, with Turkey again leading in both total and percentage terms. Firefox transmitted fewer cookies overall, but Turkey-Firefox showed an unusually low rate of third-party cookie transmission, which may reflect either measurement artifacts or stricter blocking. At the response level, Germany generally had the lowest prevalence of cookie-setting, whereas Macedonia recorded the highest, particularly for advertising- and tracking-related responses, suggesting regional differences in tracker behavior and regulatory influence.

Across all three countries, *uBlock Origin* narrowed these disparities. After filtering, advertising- and tracking-related request and cookie counts converged to similar levels regardless of region, indicating that effective extensions can normalize residual tracking risk across environments.

V. CONCLUSION

The study demonstrates that while some browsers and extensions can substantially lower tracking, none can eliminate it entirely. Tracking techniques persist through multiple channels, showing that privacy protection is not a single-step

TABLE VII
AD&TRACKING HTTP/S REQUESTS BY EXTENSION

Country	Extension	Total Req.	A&T Req.	% A&T
Germany	Firefox Default	89,442	35,026	39.16%
Germany	Adblock Plus	33,077	10,192	30.81%
Germany	uBlock Origin	68,867	16,411	23.83%
Macedonia	Firefox Default	103,851	48,946	47.13%
Macedonia	Adblock Plus	93,347	37,629	40.31%
Macedonia	uBlock Origin	66,053	16,621	25.16%
Turkey	Firefox Default	77,718	32,668	42.03%
Turkey	Adblock Plus	93,185	40,321	43.27%
Turkey	uBlock Origin	60,044	14,884	24.79%

solution but an ongoing effort. Effective defenses, therefore, need to combine stronger continued research into new countermeasures. These findings point to the importance of viewing privacy as a layered strategy rather than relying on one tool or setting.

REFERENCES

[1] Pouneh Nikkhah Bahrami, Umar Iqbal, and Zubair Shafiq. "Fp-radar: Longitudinal measurement and early detection of browser fingerprinting". In: *arXiv preprint arXiv:2112.01662* (2021).

[2] Soumaya Boussaha et al. "FP-tracer: Fine-grained Browser Fingerprinting Detection via Taint-tracking and Entropy-based Thresholds". In: *Proceedings on Privacy Enhancing Technologies* (2024).

[3] Maryam Bubukayr and Mounir Frikha. "Effective Techniques for Protecting the Privacy of Web Users". In: *Applied Sciences* 13.5 (2023), p. 3191. DOI: 10.3390/app13053191.

[4] Quan Chen et al. "Cookie swap party: Abusing first-party cookies for web tracking". In: *Proceedings of the Web Conference 2021*. 2021, pp. 2117–2129.

[5] DebugBear. *Chrome Extension List Dataset*. [Online]. Available: https://github.com/DebugBear/chrome-extension-list. [Accessed: Jul. 17, 2025]. 2024.

[6] *Disconnect tracker protection list*. [Online]. Available: https://raw.githubusercontent.com/disconnectme/disconnect-tracking-protection/master/services.json. [Accessed: Jul. 24, 2025].

[7] *EasyList filter list*. [Online]. Available: https://easylist.to/easylist/easylist.txt. [Accessed: Jul. 27, 2025].

[8] *EasyPrivacy filter list*. [Online]. Available: https://easylist.to/easylist/easyprivacy.txt. [Accessed: Jul. 21, 2025].

[9] Amin FaizKhademi, Mohammad Zulkernine, and Komminist Weldemariam. "FPGuard: Detection and prevention of browser fingerprinting". In: *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer. 2015, pp. 293–308.

[10] Imane Fouad, Cristiana Santos, and Pierre Laperdrix. "The Devil Is in the Details: Detection, Measurement and Lawfulness of Server-Side Tracking on the Web". In: *Proceedings on Privacy Enhancing Technologies* 2024.4 (2024), pp. 450–465. DOI: 10.56553/popets-2024-0125.

[11] Umar Iqbal et al. "Adgraph: A graph-based approach to ad and tracker blocking". In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2020, pp. 763–776.

[12] Kostas Chatzikokolakis and others. *Location Guard Firefox add-on*. [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/location-guard/. [Accessed: Jul. 29, 2025]. 2022.

[13] Pierre Laperdrix, Benoit Baudry, and Vikas Mishra. "FPRandom: Randomizing core browser objects to break advanced device fingerprinting techniques". In: *International Symposium on Engineering Secure Software and Systems*. Springer. 2017, pp. 97–114.

[14] Victor Le Pochat et al. "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation". In: *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. NDSS 2019. Feb. 2019. DOI: 10.14722/ndss.2019.23386.

[15] Victor et al. Le Pochat. *Tranco Top Sites List GV62K*. https://tranco-list.eu/list/GV62K. 2025.

[16] Matt Zeunert. *Chrome Extension Statistics: Most Popular Extensions*. [Online]. Available: https://www.debugbear.com/blog/chrome-extension-statistics. [Accessed: Jul. 23, 2025]. 2024.

[17] Johan Mazel, Richard Garnier, and Kensuke Fukuda. "A Comparison of Web Privacy Protection Techniques". In: *Computer Communications* 144 (2019), pp. 162–174. DOI: 10.1016/j.comcom.2019.04.005.

[18] Maryam Mehrnezhad, Kovila Coopamootoo, and Ehsan Toreini. "How Can and Would People Protect From Online Tracking?" In: *Proceedings on Privacy Enhancing Technologies* 2022.1 (2022), pp. 105–125. DOI: 10.2478/popets-2022-0006.

[19] Georg Merzdovnik et al. "Block Me If You Can: A Large-Scale Study of Tracker-Blocking Tools". In: *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2017, pp. 319–333. DOI: 10.1109/EuroSP.2017.26.

[20] Nick Nikiforakis, Wouter Joosen, and Benjamin Livshits. "Privaricator: Deceiving fingerprinters with little white lies". In: *Proceedings of the 24th International Conference on World Wide Web*. 2015, pp. 820–830.

[21] Philip Raschke and Thomas Cory. "Presenting a Client-based Cross-browser Web Privacy Measurement Framework for Automated Web Tracker Detection Research". In: *2022 3rd International Conference on Electrical Engineering and Informatics (ICon EEI)*. IEEE. 2022, pp. 98–103.

[22] Kyungmin Sim, Honyeong Heo, and Haehyun Cho. "Combating Web Tracking: Analyzing Web Tracking Technologies for User Privacy". In: *Future Internet* 16.10 (2024), p. 363. DOI: 10.3390/fi16100363.

[23] StatCounter Global Stats. *Desktop Browser Market Share Worldwide (Jul 2024 – Jul 2025)*. [Online].

Available: https://gs.statcounter.com/. [Accessed: Jul. 28, 2025]. 2025.

[24] Xiaoxi Wang et al. “XFP-recognizer: detecting cross-file browser fingerprinting”. In: *Cybersecurity* 8.1 (2025), p. 44.

[25] Yong Yuan et al. “AST-Trans: Detecting Web Tracking Using Transformer-Based Deep Learning with Abstract Syntax Tree”. In: *2024 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. 2024, pp. 1–7. DOI: 10.1109/IPCCC59868.2024.10850137.

[26] Yubi. *Change Geolocation (Location Guard) Chrome extension*. [Online]. Available: https://chromewebstore.google.com/detail/change-geolocation-locati/lejoknkbcogjceoniealiipllomkpioe. [Accessed: Jul. 18, 2025]. 2025.

[27] Rui Zhao. “FProbe: The Flow-Centric Detection and a Large-Scale Measurement of Browser Fingerprinting”. In: *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*. IEEE. 2023, pp. 1–10.

'JaVul': a Novel Java Dataset for Code Vulnerability Detection Based on CWE Labeling

Klesida Gjana
Ss Cyril and Methodius University
Skopje, North Macedonia

Dr. Hristina Mihajloska
Ss Cyril and Methodius University
Skopje, North Macedonia

Dr. Emrullah Fatih Yetkin
Kadir Has University
Istanbul, Turkiye

***Abstract*—Some of the most significant problems leading to exploitable systems are overlooked code vulnerabilities. To improve the quality of the Machine Learning models used to predict these vulnerabilities, researchers should extensively analyze the quality and relevance of the data available. This study addresses the necessity for a CWE-labeled dataset to aid in the automated vulnerability detection in Java/Spring Boot applications using advanced ML techniques.**

***Index Terms*—vulnerability detection, AI-driven security, graph representation, token representation, Java code, CWE labels**

I. Introduction

Online systems and applications are our daily companions in every life activity, making them a target to malicious activity that can significantly impact the disruption of our lives. Furthermore, it can damage our virtual possessions, such as our data and accounts. Software vulnerabilities continue to threaten modern applications, especially those developed using widely adopted frameworks like Java Spring Boot. It becomes necessary to not only react to these attacks but also try to predict their probability. The simplest way to predict the vulnerability of the applications to malicious exploitation is to start with the fundamental quality of the code used to build them. As the Common Weakness Enumeration (CWE) has shown, labeling code as a potential vulnerability can be very beneficial, not only as an alert to what can go wrong but also as an insight into which developers should take measures to improve the quality of the code. Within the Java ecosystem, vulnerabilities categorized under CWE persistently expose applications to SQL injection, cross-site scripting, insecure deserialization, and other critical flaws, underscoring the importance of employing CWEs as a structured framework to comprehend and mitigate such risks. Recent advances in machine learning offer a new pathway: train models to recognize subtle patterns that correlate with known CWE vulnerabilities. Early results in other languages show promise, but little work focuses specifically on Java. While machine learning has demonstrated potential to enhance detection capabilities, existing research lacks specific approaches leveraging the finely grained CWE taxonomy. No dedicated model exists for Java/Spring Boot applications; therefore, developing a system specifically trained to detect challenging and problematic CWE classes within Java applications would be highly advantageous. Such a system would offer developers immediate insight into areas requiring attention, as accurately labeling weaknesses facilitates a deeper understanding of appropriate mitigation strategies. The first critical step involves data procurement, as it is clear that the system's effectiveness will significantly depend upon the quality of data it receives, necessitating comprehensive research into existing databases to determine their potential value and compatibility with the intended system. After assembling a labeled dataset of Java snippets annotated with CWE IDs, it is crucial to identify, first and foremost, the optimal approach in terms of code representation, engineer features (or token embeddings) that capture code semantics. Consequently, this research will provide a curated and accurately labeled CWE dataset as a foundation for future studies. It is important to note that this project is confined exclusively to Java code, with CWE labels determined based on the data procured.

II. Research questions

Considering the main motivation of this research is procuring the necessary data to train Machine Learning (ML) models in vulnerability detection, the research questions were defined as follows:

- RQ2: How to efficiently build a database capturing vulnerable code and the appropriate CWE labeling?
- RQ3: What limitations can arise in the selected work path?

III. Background and Related Works

Despite considerable progress in Artificial Intelligence (AI) based vulnerability detection, reliance on predefined rules, struggles with zero-day vulnerabilities, and false positives continue to exist as limitations [1]. An essential review component consists of analyzing the existing datasets. The databases considered in this review provide data regarding vulnerable code, information on Java vulnerabilities, or CWE classification. Benchmarks such as Juliet, BigVul, Vul4J, VJBench, CWE-Bench-Java, and PrimeVul are discussed as standard datasets for ML research. CWE-Bench-Java and PrimeVul are specifically constructed to be mapped directly to CWE types for research consistency and comparability [4], [2]. Firstly, to differentiate the databases' types, the databases' source should be considered. Databases can be synthetic, typically created by applying known patterns, templates, or mutation rules to produce vulnerable and non-vulnerable. Otherwise,

databases that contain code from real-world scenarios usually contain individual functions within open-source repositories, predominantly hosted on GitHub. Table I provides generic, relevant information on the databases under consideration.

TABLE I
DATASET CHARACTERISTICS (✓ = YES, ✗ = NO)

Dataset	Real-World	Size over 10k	Method Level
Juliet	✗	✓	✗
OWASP Benchmark	✗	✗	✗
Vul4J	✓	✗	✗
VJBench	✓	✗	✓
CWE-Bench-Java	✓	✗	✗
PrimeVul	✓	✗	✓
DiverseVul	✓	✓	✓
CVEfixes	✓	✓	✗
YesWeHack	✗	✗	✓

Vulnerability taxonomies and standards are essential for systematically classifying and detecting software security flaws. The CWE framework, maintained by MITRE, version 4 being the subject of the latest studies, is also the version this study uses, providing a hierarchical, N-tier framework that enables high-level overviews and detailed vulnerability labeling. This taxonomy emphasizes breadth and depth in categorizing software vulnerabilities, supporting consistent tracking and analysis across diverse systems [18]. Employing standardized classifications ensures consistency across tools and datasets, facilitates benchmarking, and supports meaningful comparisons in vulnerability detection research. The CWE taxonomy is especially beneficial when training AI in classification tasks, as it provides valuable labeling. In the scope of this research, only a single label is attributed, and only the CWE taxonomy is used.

A notable gap in the literature stems from the reliance on synthetic or narrowly scoped datasets. While benchmarks such as Juliet offer valuable controlled environments, they do not reflect the breadth and complexity of vulnerabilities found in production code: ”Synthetic datasets such as Juliet are generated based on a few predefined patterns and thus cannot represent the diverse behaviors observed in real-world programs [1]. Consequently, many models that achieve high accuracy on curated or synthetic data suffer significant performance degradation when applied to large, heterogeneous codebases, owing mainly to the scarcity of diverse, representative, and sufficiently annotated real-world datasets [4], [9], [12]. Although datasets like BigVul and Vul4J represent an improvement as their real-world breadth introduces variability, they often lack essential contextual information, deduplication procedures, and uniformly high-quality labels [2], [4]. Their coverage and labeling consistency can fluctuate across projects, reflecting the inherent difficulties of accurately annotating vulnerabilities in diverse codebases [14]. Furthermore, because the dataset predominantly includes vulnerable samples, models trained solely on datasets such as BigVul, YesWehack, and DiverseVul, who lacks corresponding non-vulnerable examples, risk heightened false-positive rates due to the lack of balanced non-vulnerable examples in the training set [1], [13]. Finally, the inherent complexities of Java, particularly its dependency-injection and aspect-oriented programming paradigms effect the scarcity of vulnerability detection tools and datasets specifically tailored to Java applications [2]–[4], [11].

IV. METHODOLOGY

This section describes the practical approach to creating the database, from the selection of the databases to extract data, to the graph and vector representation of code. The methodology used for the database creation includes defining the data structure and populating it with values. The second step is transforming the data into the chosen graph and semantic representation and persisting them in the database.

This phase focused on gathering and preparing datasets for training AI models. The main objective was to address the lack of a large-scale Java corpus with fine-grained CWE labels containing vulnerable and non-vulnerable examples. To meet this need, a PostgreSQL database named javul was designed and populated. The database combines code from both synthetic and real-world sources. This mixture was chosen to exploit the low noise and comprehensive coverage of artificial datasets while incorporating the contextual complexity and generalization benefits of real-world code. Initial data was sourced from the Juliet Java Test Suite [19] and the OWASP Benchmark [5], both of which provide reliable CWE labels and broad vulnerability coverage, including rare or subtle flaws typically ensured by synthetic generation. To complement this, the CVEfixes [6] and YesWeHack [15] datasets were included, contributing real-world code with greater diversity, scale, and realistic software complexity. This balance aimed to create a dataset that supports model generalization while covering a broad range of CWE classes.

The next step was defining a schema suitable for training needs. The database schema includes:

- a primary key `id`, generated as a unique alphanumeric string;
- `raw_code`, containing the Java method source code;
- `cwe_id`, serving as the training label;
- a Boolean `is_vulnerable` flag;
- `source`, specifying the dataset of origin;
- JSONB fields `ast_graph`, `cfg_graph`, and `dfg_graph` for serialized AST, CFG, and DFG structures;
- `css_vector`, a PostgreSQL array of doubles for pre-trained semantic embeddings.

By keeping this information in the database, we aim to facilitate other studies that might use similar information, whether for model training or statistical analysis.

A. Data Extraction

Once the schema was set, the datasets were parsed to populate the database. Python scripts were implemented to

extract each source dataset according to its specific organization and metadata format. Although the source repositories often contain class-level code, it was decided to store methods as the primary unit of analysis. This approach fits the fact that Juliet and OWASP Benchmark vulnerabilities are typically demonstrated at the method level. In Juliet, each CWE is represented by multiple self-contained test case classes. For every seeded vulnerability, at least one 'bad' implementation demonstrates the flaw, and one or more 'good' implementations show the corrected or safe code [1]. The Java source files were processed line by line, with method signatures detected using a regular expression matching patterns. Once a matching signature was found, the method body was recorded in full, with brace depth tracked to ensure correct scope handling for multi-line definitions. Each method name and its complete source code were stored as a tuple, preserving multi-line formatting and ensuring correct extraction even for complex method bodies.

The OWASP Benchmark is implemented as an extensive Java web application containing thousands of servlet classes, each representing a unique, vulnerable or non-vulnerable test case for a specific CWE. CWE mappings are provided in a companion CSV file [5]. The extraction process parsed this metadata, matched each servlet to its vulnerability status and CWE label, and batch-inserted the resulting records into the database. Many cases include both vulnerable and secure implementations of the same functionality. In YesWeHack, code is organized by programming language, with further grouping by vulnerability type. Some folders contain vulnerable and patched versions, but most only contain vulnerable examples. Each snippet includes header comments describing the vulnerability, context, and sometimes remediation advice [15]. Only one Java example was present, which was added manually to the database via a direct query.

The CVEfixes dataset is distributed as a SQLite database containing multiple tables [6]:

- `fixes`, listing commits linked to CVEs;
- `method_change`, detailing added, removed, or modified methods per commit;
- `file_change`, tracking affected files;
- `cwe_classification`, mapping CVEs and code changes to CWE categories where available.

.

The extraction process restored the SQLite dump, joined the relevant tables to gather method-level code with CWE labels, filtered for Java entries with non-null CWE fields, and migrated these into the PostgreSQL database. These datasets were particularly valuable for their numerous non-vulnerable examples, helping maintain balance between positive and negative labels and preventing overfitting to vulnerabilities. This balance better reflects real-world conditions, where safe code is more common than flawed code.

B. Graph Representation of Code

Graph-based representations such as ASTs, CFGs, and DFGs have become foundational in AI-driven software vulnerability detection due to their ability to encode both the syntactic and semantic structure of code. An AST hierarchically models the syntactic structure of source code, where each node represents a language construct, such as statements, expressions, or operators, enabling ML models to capture complex nesting and relationships beyond linear token sequences. Parsing code into ASTs is a standard process in compilers and static analyzers, and these trees serve as rich feature sources for vulnerability pattern recognition in deep learning systems [2], [1].

In contrast, a CFG abstracts the possible execution paths of a program by representing basic blocks of instructions as nodes and control flow as edges, which is critical for reasoning about vulnerabilities that arise from specific sequences or paths of code execution. Examples include unreachable code, among others. Meanwhile, a DFG focuses on how data values and variables propagate through the code, with nodes denoting operations or storage and edges capturing data dependencies; DFGs are especially valuable for detecting vulnerabilities related to taint analysis, such as input validation flaws or unsafe data propagation [10], [1].

The subsequent stage in database generation involved extracting Abstract Syntax Trees, Control-Flow Graphs, Data-Flow Graphs, and semantic embeddings from Java methods. A Java project is built with classes that extend JavaParser for the parsing of the three types of graphs. Each method was parsed into its syntactic structure for AST construction, recording node types, code excerpts, and parent–child relationships. Nodes were assigned stable integer identifiers, and only semantically relevant elements (statements, expressions) were retained, omitting structural delimiters. The resulting representation consisted of node and edge lists that preserved the source order and syntactic hierarchy.

CFGs were generated at the statement level, with unique identifiers assigned to all non-block statements and explicit ENTRY and EXIT nodes marking the boundaries. Edges captured sequential execution, fall-through between consecutive statements, and explicit exception handling paths, such as transfers from try blocks to their associated catch blocks. This structure provided a minimal yet accurate control flow model within the same method, stored in a uniform node–edge JSON format.

DFGs were built to represent definition-use relationships local to the method body. Variable declarations and assignment targets were marked as definition nodes, while variable uses were connected to their most recent reaching definitions. Edges, therefore, represented the relationship between each variable used and its most recent definition within the method body. Interprocedural links, aliasing, and detailed field or array tracking were intentionally excluded to maintain focus on local flows. All three graph types were serialized into a consistent JSON object format, with each node containing its identifier and essential attributes, and each edge specifying its source and target identifiers. These were stored in dedicated JSONB columns in the database.

Parsing failures during AST/CFG/DFG extraction largely

stemmed from mismatches between the syntactic units produced by the extractor and those expected by the parser. Java parsers such as JavaParser require well-formed constructs (e.g., complete method or constructor declarations), but many raw snippets failed to meet this requirement. Out of 219,264 code fragments, only 9,306 could be successfully parsed, with typical failures caused by constructor snippets misinterpreted as methods, body-only fragments lacking signatures, non-Java text or encoding artifacts, embedded comments or metadata, and truncated or unbalanced delimiters. Because most static parsers lack robust error recovery, these failures triggered exceptions rather than yielding partial ASTs, which also prevented CFG and DFG generation. The consequences are twofold: coverage loss, as the majority of the dataset is excluded from graph-based analysis, weakening statistical power; and selection bias, since the successfully parsed subset disproportionately represents canonical, top-level methods. Both effects compromise the validity of downstream experiments, including model training, CWE distribution analysis, and robustness evaluation.

C. Tokenized code representation

Token-level representation of source code is a pivotal preprocessing step for machine–learning–based vulnerability detection. In this stage, a lexer or tokenizer, guided by the programming language's grammar, decomposes code into atomic tokens (e.g., keywords, identifiers, literals), thereby linearizing the program into a sequence amenable to NLP-inspired models [10]. While this flattening facilitates the use of architectures such as CNNs, RNNs, and Transformers, it inevitably discards hierarchical and semantic information inherent in abstract syntax and data-flow structures [1], [16], [17]. Researchers have developed code embeddings to recover richer context: dense vector representations that encode syntactic and semantic token relationships. More advanced approaches leverage Transformer-based models such as BERT, CodeBERT, GraphCodeBERT, and CodeT5 pretrained on vast code repositories. These models yield contextualized embeddings for individual tokens, statements, or entire functions, capturing nuanced programming constructs and patterns [16]. Empirical studies demonstrate that concatenating token embeddings with positional, type, or data-flow-aware embeddings further mitigates the loss of structural context, substantially improving vulnerability classification performance. By bridging the symbolic domain of code and the continuous domain of deep learning, these embedding techniques enable scalable, automated detection of security weaknesses across diverse and large codebases [1], [17].

Finally, semantic embeddings were generated for each code snippet using CodeBERT, a pretrained deep learning model for programming language understanding. CodeBERT has a Transformer-based neural architecture, and it is trained with a hybrid objective function that incorporates the pre-training task of replaced token detection. After being evaluated on two NL-PL applications, the results show that CodeBERT achieves state-of-the-art performance on both natural language code search and code documentation generation tasks [16].

In this context, semantic embeddings refer to a dense, fixed-length numerical representation of a Java method that captures its semantic meaning rather than only its syntax or structure. Once each method's code was extracted from the dataset, it was processed with the tokenizer provided by the CodeBERT-method's. This step splits the code into tokens, identifying both language elements, such as keywords and operators, and structural components, such as identifiers. The tokenized sequence was then passed into the CodeBERT transformer model, which produced a contextual embedding for each token. These embeddings were generated using self-attention layers, allowing every token to incorporate information from all others in the sequence so that its representation reflected its surrounding context. To create a single vector for the method, a CLS pooling operation was applied. The CLS token, inserted at the start of the sequence, was produced during model training to summarize the most relevant global information about the input. The resulting vector, initially a one-dimensional tensor of length 768, was converted into a standard Python list and stored in the database as a PostgreSQL array. Figure 1 maps the semantic embedding extraction method into a diagram.

V. Results

A composition representation of the table is given in Table II. The initial analysis of the resulting database listed the following classes as most frequent:

- **CWE-190**: Integer Overflow .
- **CWE-191**: Integer Underflow .
- **CWE-129**: Improper Validation of Array Index.
- **CWE-89**: Improper Neutralization of Special Elements used in an SQL Command.
- **CWE-369**: Divide by Zero.

TABLE II
Composition of the constructed Java vulnerability dataset.

Source	Total Samples	Vulnerable	CWE Classes Covered
Juliet Test Suite	207892	46327	112
OWASP Benchmark	2740	1415	11
CVEfixes	8631	3046	85
YesWeHack	1	1	1
Total	219,264	50789	182

In Figure 2, three complementary views of the same data are demonstrated. PCA (left column) exposes dominant linear modes in the embeddings: both CSS and Combined PCA show a broad, band-like structure, indicating a few principal directions that explain most variance, but these do not correspond to CWE classes. t-SNE (middle column) breaks the data into many small, tightly-packed islands; this method maximizes local neighborhood fidelity and yields the highest nearest-neighbor label agreement, implying that same-label samples are often near each other locally even if they are not globally

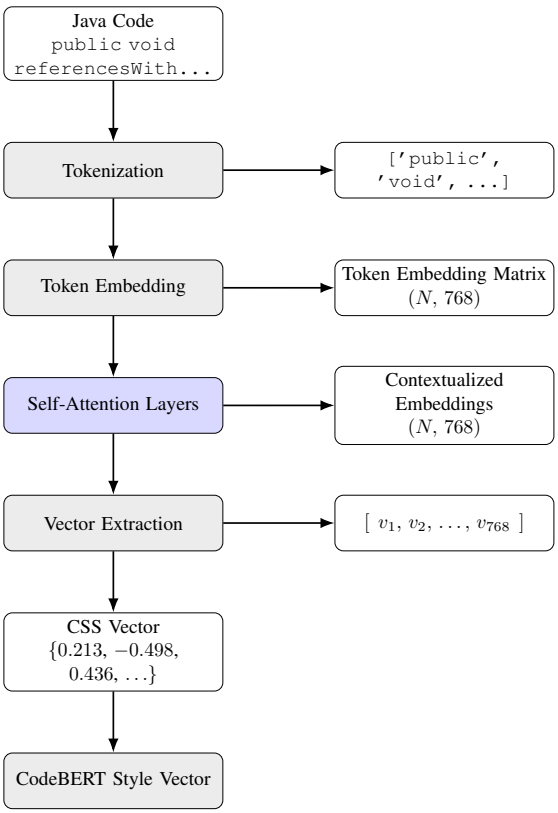


Fig. 1. Workflow from Java code to a CodeBERT-based CSS vector.

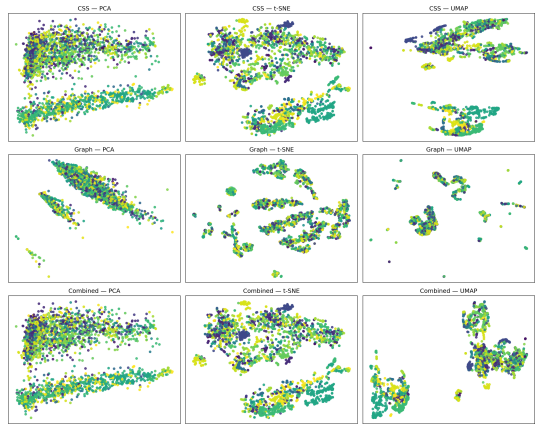


Fig. 2. Two-dimensional visualizations (PCA, t-SNE, UMAP) of three feature sets (rows: CSS semantic embeddings, Graph summary features, Combined) showing differing local and global class separability for CWE labels.

separated. UMAP (right column) offers a middle ground as it produces more globally coherent clusters than t-SNE while preserving local neighborhoods better than PCA. Notably, the combined feature set (semantic and graph) often shows the clearest islands in UMAP, which implies that semantics and structural summaries are complementary. Across all panels, there is substantial color mixing inside many clusters: multiple CWE labels overlap in the same regions, showing that some CWEs are not linearly separable in these features, labels may be noisy or multi-faceted, and projecting high-dimensional features down to two dimensions necessarily discards discriminative information.

Three conclusions emerge: pretrained CSS embeddings capture dominant semantic variance but miss structural cues; even simple graph summaries recover useful motifs, suggesting richer graph embeddings (e.g., graph2vec or GNNs) could improve separability; and combining semantics and structure produces the most coherent clusters, particularly in UMAP. Limitations include class imbalance, label noise (e.g., methods with multiple weaknesses annotated by a single CWE), and projection artifacts. Duplicate vectors and strong linear modes also bias nonlinear embeddings, which we mitigated via deduplication, and PCA pre-reduction.

VI. Discussion and Limitations

Important limitations of this study begin with the reliance on only four primary datasets. While these sources are well-established, they constrain the diversity of both code structures and CWE categories; future work should incorporate additional collections to broaden coverage. Despite efforts to include underrepresented CWE labels, the resulting dataset remains highly imbalanced, with a few dominant classes overshadowing rare ones. This imbalance reduces the reliability of cross-validated metrics and biases cluster-level statistics toward frequent classes.

Another challenge lies in the inherent ambiguity of CWE labeling. Many methods may exhibit multiple weaknesses, yet are annotated with only a single CWE, introducing label noise and reducing class separability. Furthermore, projecting high-dimensional features into two dimensions inevitably discards information, so low silhouette scores may reflect not only true semantic overlap but also projection artifacts.

Finally, the unified database is Java-centric, limiting the generality of findings across programming languages. Extending the methodology to additional languages would enable broader validation. Overall, while the combined semantic and structural features reveal promising local structure, robust claims about CWE separability will require stronger class balancing, richer graph encodings, and more comprehensive quantitative validation.

VII. Conclusions and Future Work

This work presented a systematic methodology for constructing a curated dataset of Java code annotated with CWE labels, enriched with graph and semantic embeddings. The

pipeline begins with the collection of labeled vulnerability data from multiple reputable sources, including OWASP Benchmark [5], CVEfixes [6], the National Vulnerability Database (NVD) [7], and the Juliet Test Suite [8]. A unified database schema was designed to capture both vulnerable and non-vulnerable Java methods, explicitly annotated by CWE categories. To ensure robustness, the dataset includes a considerable proportion of non-vulnerable code samples, thereby reducing the likelihood of false positives during model training and evaluation.

Each code sample was further enriched with multiple representations: Abstract Syntax Trees (AST), Control Flow Graphs (CFG), and Data Flow Graphs (DFG) generated via JavaParser, alongside semantic embeddings obtained from the pretrained CodeBERT model. The resulting dataset provides not only raw code but also structured and semantic views, offering a versatile foundation for downstream studies in vulnerability detection, benchmarking, and model development. The primary contribution of this work is therefore a novel, fine-grained, CWE-labeled dataset for Java that explicitly addresses gaps in existing resources, particularly the lack of graph-based and semantic representations tailored to underrepresented CWE classes.

Future work will focus on leveraging these representations for advanced machine learning architectures, particularly multi-view or self-attention-based models that can jointly reason over syntactic, structural, and semantic features of code. Additional directions include extending coverage to inter-procedural graphs that capture context across methods and classes, enriching the dataset with more real-world vulnerabilities, and exploring automated augmentation techniques to balance rare CWE labels. Furthermore, evaluation of generalization across unseen projects and integration of richer graph neural network embeddings represent promising avenues to enhance both accuracy and robustness.

REFERENCES

[1] N. S. Harzevili, J. A. Ansari, H. Kazmi, and B. R. Shrestha, "A Systematic Literature Review on Automated Software Vulnerability Detection Using Machine Learning," ACM Computing Surveys, vol. 58, no. 1, pp. 1–38, Jan. 2025.

[2] J. Wang, Z. Huang, H. Xiao, and Y. Xiao, "JFinder: A Novel Architecture for Java Vulnerability Identification Based Quad Self-Attention and Pre-training Mechanism," arXiv preprint arXiv:2307.15915, Jul. 2023.

[3] Z. Li, S. Dutta, and M. Naik, "LLM-Assisted Static Analysis for Detecting Security Vulnerabilities," arXiv preprint arXiv:2405.17238, May 2024.

[4] S. Al Atiiq, C. Gehrmann, K. Dahlén, and K. Khalil, "From Generalist to Specialist: Exploring CWE-Specific Vulnerability Detection," arXiv preprint arXiv:2408.02329, Aug. 2024.

[5] OWASP, "OWASP Benchmark Project," Open Worldwide Application Security Project, 2025. [Online]. Available: https://owasp.org/www-project-benchmark/.'

[6] G. P. Bhandari, A. Naseer, and L. Moonen, "CVEfixes: Automated Collection of Vulnerabilities and Their Fixes from Open-Source Software," in *Proc. 17th Int. Conf. Predictive Models Data Analytics in Software Engineering (PROMISE' 21)*, 2021, pp. –, doi: 10.1145/3475960.3475985.

[7] H. Booth, D. Rike, and G. A. Witte, "The National Vulnerability Database (NVD): Overview," ITL Bulletin, National Institute of Standards and Technology, Gaithersburg, MD, Dec. 18, 2013. [Online]. Available: https://tsapps.nist.gov/publication/.

[8] National Institute of Standards and Technology, "Software Assurance Reference Dataset (SARD)," SAMATE / SARD, updated Apr. 22, 2024. [Online]. Available: https://samate.nist.gov/SARD.

[9] Y. Zheng, S. Pujar, B. Lewis, L. Buratti, E. Epstein, B. Yang, J. Laredo, A. Morari, and Z. Su, "D2A: A Dataset Built for AI-Based Vulnerability Detection Methods Using Differential Analysis," Proc. 2021 IEEE/ACM 43rd Int. Conf. on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Madrid, Spain, May 2021, pp. 28–38. doi: 10.1109/ICSE-SEIP52605.2021.00010.

[10] Z. Li, D. Zou, S. Xu, X. Ou, H. Jin, S. Wang, Z. Deng, and Y. Zhong, "VulDeePecker: A Deep Learning-Based System for Vulnerability Detection," Proc. Network and Distributed Systems Security (NDSS) Symp.,San Diego, CA, USA, Feb. 2018, pp. 1–15. doi: 10.14722/ndss.2018.23158.

[11] Y. Wu, N. Jiang, H. V. Pham, T. Lutellier, J. Davis, L. Tan, P. Babkin, and S. Shah, "How Effective Are Neural Networks for Fixing Security Vulnerabilities," Proc. 32nd ACM SIGSOFT Int. Symp. Softw. Test. Anal. (ISSTA), Seattle, WA, USA, Jul. 2023, pp. 1–13. doi: 10.1145/3597926.3598135.

[12] Y. Zhang, Y. Xiao, M. M. A. Kabir, D. Yao, and N. Meng, "Example-Based Vulnerability Detection and Repair in Java Code," Proc. 30th Int. Conf. Program Comprehension (ICPC), Virtual Event, USA, May 2022, pp. 1–12. doi: 10.1145/3524610.3527895.

[13] R. Fan, X. Zeng, B. Li, and Y. Liu, "Big-Vul: A Real-world Dataset for Vulnerability Detection in C/C++ Functions," arXiv preprint arXiv:2002.10359, 2020. [Online]. Available: https://arxiv.org/abs/2002.10359

[14] X. Fan, Y. Wu, H. V. Pham, and T. Lutellier, "Vul4J: A Dataset for Java Vulnerability Repair," in Proc. 2020 IEEE 27th Int. Conf. on Software Analysis, Evolution and Reengineering (SANER), 2020, pp. 757–761, doi:10.1109/SANER48275.2020.9054823.

[15] YesWeHack, "YesWeHack Vulnerable Code Snippets," GitHub repository, 2018. [Online]. Available: https://github.com/YesWeHack/vulnerable-code-snippets.

[16] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, and M. Zhou, "CodeBERT: A Pre-Trained Model for Programming and Natural Languages," in Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 1536–1547. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.139

[17] M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton, "A Survey of Machine Learning for Big Code and Naturalness," ACM Comput. Surv., vol. 51, no. 4, pp. 81:1–81:37, 2018, doi: 10.1145/3212695.

[18] MITRE, "CWE: Common Weakness Enumeration," [Online]. Available: https://cwe.mitre.org. [Accessed: Jul. 15, 2025].

[19] NIST SAMATE, "Juliet Test Suite for Java," [Online]. Available: https://samate.nist.gov/SARD/test-suites/111. [Accessed: Jul. 15, 2025].

Public Cybersecurity Awareness in the European Union

1st Danko Nakić
SRH University of Applied Sciences - Heidelberg and Kadir Has Universityl
Berlin, Germany and Istanbul, Türkiye
danko.nakic@gmail.com

2nd Prof. Dr. Reiner Creutzburg
SRH University of Applied Sciences - Heidelberg
Berlin, Germany
reiner.creutzburg@gmail.com

3rd Prof. Dr. Hasan Dağ
Kadir Has University
Istanbul, Türkiye
Hasan.dag@khas.edu.tr

4th Dr. Klaus Schwarz
SRH University of Applied Sciences - Heidelberg
Berlin, Germany
klaus.schwarz@srh.de

***Abstract*—Significant advancements in technology and the intensified use of information and digital systems have heightened susceptibility to cyber threats, thereby increasing the necessity to enhance public and professional awareness of cybersecurity. This paper analyzes various demographic factors of an online survey to get a clear picture of how well the public in the European Union understands the importance of cybersecurity, highlighting the need for integration of cybersecurity awareness into cybersecurity strategies for both public and private entities.**

***Index Terms*—Cybersecurity, Cybersecurity Awareness, Cyber threats, EU, European Union, Public, Survey, Cyber Hygiene**

I. Introduction

Although the everyday use of technology has allowed us to be more productive and efficient, simplified many repetitive tasks, and contributed to societal development, its evolution also comes with significant responsibility. Today, a growing number of individuals prefer to conduct personal tasks online, such as shopping or paying bills through modern applications.

These and many other activities expose individuals to potential hacker attacks. Remote work is also becoming increasingly popular. Remote work has not only expanded the potential attack surface but has also moved it beyond traditional perimeter defenses, such as firewalls and intrusion detection systems, which organizations have historically implemented to prevent ransomware attacks, data breaches, and other forms of cybercrime.

However, according to the European Union Agency for Cybersecurity (ENISA), this progress brings new challenges and risks related to cybersecurity, which often go unnoticed by the public. The need to raise awareness and hone skills about cybersecurity is becoming increasingly urgent due to the dependence on Information and Communications Technology (ICT) in all aspects of society [1].

We can define cybersecurity awareness as the knowledge and attitudes that users have regarding the protection of information assets, involving recognizing security threats, understanding best practices, and knowing how to avoid risky behaviors to enhance cybersecurity [2]. This awareness is crucial for protecting both personal and organizational data from cyber-attacks and has evolved significantly over time.

Cybersecurity awareness began in the 1960s and first developed through the 1980s when researchers first recognized that human factors were critical to system security, initially focusing on basic computer security concerns among technical personnel. The emergence of personal computers and high-profile incidents like the Morris Worm in 1988 expanded this awareness beyond technical specialists to general users [3], ultimately developing into the modern comprehensive organizational and societal initiatives we see today that address human factors in cyber threats.

The United Nations General Assembly has stated that rapid advances in information technology have transformed the way governments, businesses, organizations, and individuals (who through their intentional or unintentional actions pose the greatest threat to cybersecurity) - must approach cybersecurity issues. A global cybersecurity culture is developing as a measure to promote safe behavior worldwide [4].

In this new world where a single mistake and a click on a fraudulent email link can compromise the security of a sovereign country, the following questions are raised:

1.1. Research questions

1.) What is the current level of public cybersecurity awareness in the European Union?

2.) How does the level of public cybersecurity awareness in the European Union vary across major demographic factors?

II. Findings

A. Methodology

To investigate EU public cybersecurity awareness, this study employs a quantitative survey methodology. This approach enables comprehensive analysis of a developing domain, quantifies phenomena numerically, and allows population-wide extrapolation of findings. Quantitative methods effectively

evaluate theories, explore relationships, identify data patterns, and determine causal links [5].

The survey examines individual-level phenomena, enabling direct analysis of demographic, educational, and social group attributes [6].

Data collection uses the modified validated Standard Eurobarometer questionnaire (Directorate-General for Communication) with closed-ended questions [7]. The survey questionnaire was comprised of 18 closed-ended questions, of which 5 questions were designed to collect demographic information about the participants while the remaining 13 questions deal with various cybersecurity awareness topics.

Distribution occurs through online platforms (Reddit, Facebook, WhatsApp groups, SurveySwap) and direct emails to Computer Science departments heads, professors, and students across the European Union. The survey maintains complete anonymity.

Secondary sources of information such as academic and professional books, online articles and websites, and scholarly publications and papers were employed to develop the theoretical parts of the paper.

B. Systematic literature review

A literature review has been pre-emptively undertaken to develop a concise understanding of the current Public Cybersecurity Awareness in the European Union. European Union citizens exhibit a broad range in cybersecurity awareness [8].

Several studies identify as many as four distinct profiles—ranging from uninformed or minimally prepared users to those well versed in digital risk management [9].

One study distinguishes a two-profile system, with 19% categorized as "at-risk" and 81% as "cautious," while another finds that more than half of respondents remain only partially informed about cybercrime despite generally positive attitudes toward digital technologies [10].

These profiles emerge from analyses that emphasize digital activity, sociodemographic characteristics (such as age, education, and occupation), and, in a few cases, economic indicators [11].

Across member states, heightened cybersecurity preparedness is observed in nations with stronger economic status or more advanced digital infrastructures. [9] One investigation links higher national GDP per capita to increased individual preparedness, and different patterns of GDPR awareness and digital risk culture appear among the 28–30 countries examined primarily via Eurobarometer data [9].

Sample sizes in these studies ranged from 456 to approximately 28,000 participants, underscoring the diversity of cybersecurity experiences among EU citizens and the significant influence of local economic and digital contexts [12].

As far as cybersecurity awareness itself, according to research conducted by ENISA (European Union Agency for Cybersecurity), 60% of attacks carried out in Europe, the Middle East and Africa include social engineering as an integral part of the attack [13].

In the research carried out by ISACA, it is stated that 97% of cyberattacks would be prevented if effective protection methods were implemented, which are used to detect or prevent undesirable events or problems from external sources of the information environment [14].

Educating employees on creating strong passwords, reporting, and deleting suspicious emails, and using a VPN to access company data on personal phones will create a more secure environment both in the office and when working from home [15].

As far as the student population is concerned, which predictably made up a large part of the sample, research has found that the average cybersecurity judgment among students was notably low, suggesting a pressing need for targeted cybersecurity awareness initiatives [16].

By focusing on improving students' cybersecurity awareness as well as their practical skills, educational institutions can empower them to navigate the digital landscape more safely, ultimately fostering a more secure online environment.

Other researchers have also found that university students from Generation Z expressed a common sentiment that their formal education has not sufficiently prepared them for the complexities of modern cyber threats, and they also subsequently advocate for enhanced both curricular and extracurricular training in cybersecurity awareness [17].

Similarly, it has been observed that despite a recognition of cybersecurity's significance among students in differing disciplines, actual student behavior often does not reflect this awareness, indicating a gap between knowledge and practice in cybersecurity awareness [18].

C. Findings

The survey was succesfully filled out by 102 respondents. The majority of respondents came from Germany, 44.6%, with the second and third highest percentages being from Italy (9.9%) and Austria (6.7%). The survey participants were predominantly younger adults, with over two-thirds (67.3%) falling between ages 16-35.

The survey achieved a well-balanced gender distribution with nearly equal representation between female and male participants. Nearly three-quarters (74.3%) of respondents were found to be holding at least a bachelor's degree and less than 10% had only a high school education. Finally, more than half (53.9%) were employed, with about a third (32.4%) being students.

Respondents demonstrated extensive digital engagement across critical online activities. Universal email usage (100%) was accompanied by near-universal adoption of online banking, social networking, and e-commerce (all above 93%). Mobile devices predominated internet access patterns, with laptops (95.1%) and smartphones (98%) showing near-universal adoption, reflecting the shift toward portable, personal devices for cybersecurity-relevant activities.

Personal data misuse emerged as the primary cybersecurity concern for over two-thirds of respondents (67.6%), followed by online payment security (53.9%) and delivery reliability

(52%). Banking fraud generated the highest number of ”very concerned” responses among specific cyber threats, followed closely by malware infections and identity theft. However, ransomware attacks ranked lowest in concern levels despite their potentially devastating impact, suggesting underestimation of emerging threats compared to familiar risks like financial fraud.

Despite widespread security concerns, a notable minority (11.8%) reported having no online concerns whatsoever, revealing varying levels of risk awareness across the EU population.

A clear hierarchy emerged in cybersecurity behavior adoption. Approximately two-thirds of respondents implemented fundamental security measures including unique passwords across sites (65.4%), biometric authentication (63.5%), and reduced personal information sharing (63.5%). However, adoption of advanced protective measures dropped significantly, with only about one-third using password managers (35.6%) or antivirus software (37.5%), and fewer than 20% regularly updating passwords.

Password management behaviors revealed concerning inconsistencies across different platforms. While over half (56.9%) updated email passwords in the past year, password hygiene deteriorated for other critical services, with only 36.3% changing banking passwords and just 12.7% updating gaming account credentials.

The majority of respondents demonstrated high confidence in their cybersecurity knowledge, with over half (54.9%) feeling ”very well informed” about cybercrime risks. However,it's worth pointing out that this self-reported confidence contrasted with inconsistent security practice adoption observed elsewhere in the data.

Cybercrime experiences varied significantly by threat type. Fraudulent communications were alarmingly widespread, with 76% of respondents receiving phishing emails or calls at least once, and 44% experiencing them more than three times. In contrast, serious cybercrimes remained relatively rare, with 83% never experiencing identity theft and 87% never encountering ransomware attempts.

Within respondents' social networks, fraudulent communications had reached epidemic proportions, with three-quarters (76%) knowing someone who had received phishing attempts. Only 15.7% reported that no one in their social circle had experienced any form of cybercrime, indicating the pervasive nature of digital threats across EU communities.

Despite widespread cybercrime exposure, nearly 60% of respondents had never reported any online criminal activity. When reporting occurred, individuals preferred contacting service providers (22.5%) over law enforcement agencies (14.7%). Response patterns revealed troubling inconsistencies, with 50% choosing inaction when facing phishing attempts despite these being among the most common threats.

A critical knowledge gap emerged regarding institutional reporting mechanisms, with nearly two-thirds (60.8%) unaware of official channels for reporting cybercrimes in their countries of residence. This lack of awareness undermined effective cybercrime response and highlighted deficiencies in public information about existing reporting systems.

Respondents demonstrate heightened cybersecurity awareness, with over 90% expressing concern about increasing cybercrime risks and actively avoiding personal information disclosure online, yet they show greater distrust of websites (82% concerned) than government authorities (58% concerned). However, a striking paradox emerges as only 64% feel confident in their ability to protect themselves against cybercrime, revealing a significant gap between risk awareness and perceived personal security capabilities.

Ultimately, this comprehensive survey of cybersecurity awareness across European Union member countries reveals a population that demonstrates sophisticated awareness of cybersecurity threats combined with inconsistent implementation of protective measures.

While participants exhibited sophisticated understanding of cybersecurity threats, with personal data misuse emerging as the primary concern for over two-thirds of respondents, the research uncovered a troubling hierarchy in cybersecurity adoption that reveals significant implementation gaps.

A particularly noteworthy finding emerged in the contrast between self-reported cybersecurity confidence and actual protective behaviors, with over half feeling ”very well informed” about cybercrime risks, yet this self-assessed expertise did not align consistently with comprehensive security practices.

Finally, the research revealed a critical disconnect between cybercrime experiences and formal reporting mechanisms, with nearly 60% of respondents never reporting online criminal activity and nearly two-thirds unaware of official reporting channels in their country.

III. Conclusion

While approximately two-thirds of respondents demonstrated adoption of fundamental security measures such as unique passwords across sites, biometric authentication, and reduced personal information sharing, the utilization of more advanced protective measures dropped precipitously, with only about one-third employing password managers or antivirus software.

Response patterns to different types of cybercrime revealed troubling inconsistencies, with half of respondents choosing inaction when facing phishing attempts despite these being among the most common threats. Conversely, appropriate urgency was demonstrated for serious financial crimes, indicating that perceived severity rather than actual risk drives reporting behaviour.

An intriguing dimension of the research involved differential trust patterns between private and public sector entities. Respondents demonstrated greater distrust of websites regarding personal information security compared to government authorities, with 82% expressing concern about private sector data handling versus 58% for public authorities. This finding has significant implications for policy development and public-private cooperation in cybersecurity initiatives.

The findings of this research carry substantial implications for cybersecurity policy development and public awareness initiatives across the European Union. The identified gaps between awareness and implementation suggest that education campaigns focusing solely on threat recognition may be insufficient. Instead, comprehensive approaches that address behavioural change, practical implementation support, and systematic security habit formation appear necessary.

A widespread lack of awareness regarding official reporting cyber-crime channels, revealed by the research, represents a critical policy implementation gap that requires immediate attention. Substantial improvements in communication about official reporting channels and simplified access to cybersecurity resources appear essential for enhancing overall digital security across the region.

Finally, while this research provides valuable insights into EU cybersecurity awareness, several limitations must be acknowledged. For instance, the demographic skewing toward younger, highly educated participants limits generalizability to the broader population.

These demographic biases also highlight the need for targeted outreach to underrepresented populations, particularly older adults and individuals with lower educational attainment, who may face different cybersecurity challenges and require tailored awareness approaches.

Another limitation is the missing representation from from seven European Union member countries - Czech Republic, Denmark, Estonia, Latvia, Lithuania, Slovakia, and Sweden, collectively representing approximately 11% of the EU population and encompassing varying levels of digital infrastructure development and cybersecurity maturity. This gap represents a significant limitation in achieving comprehensive understanding of cybersecurity awareness across the European Union.

Several factors likely contributed to this geographic clustering, including language accessibility, survey dissemination networks, and varying levels of public engagement with academic research initiatives. Recent research has highlighted that capacity constraints and limited information availability have compounded these challenges in cybersecurity research across different national contexts [19], with national culture, industry type, and organizational security culture significantly shaping individuals' security behaviors across Europe [20].

To address these respondent recruitment challenges in future research, establishing formal partnerships with national cybersecurity agencies or academic institutions within each target country would provide confirmed local respondent recruitment channels together with cultural context awareness, aligning with current best practices in cybersecurity education research [21].

The development of EU-wide research infrastructure specifically designed for cybersecurity awareness assessment could address many of these challenges through standardized survey instruments available in all EU languages, established recruitment networks within each member state, and coordinated data collection protocols that account for national regulatory and cultural considerations, with ENISA potentially facilitating such initiatives through adapting its EU Cybersecurity Index (EU-CSI) framework [22].

Ultimately, the geographic gap and other limitations identified in this research, while limiting current findings, provide important lessons for improving methodological approaches in cross-national cybersecurity research.

By acknowledging these limitations and proposing concrete solutions, this research contributes not only to understanding cybersecurity awareness where data was successfully collected, but also to advancing the methodological rigor of future investigations across the complete European Union landscape.

The identified demographic disparities in cybersecurity awareness and behavior necessitate tailored intervention strategies rather than uniform awareness campaigns. For students and younger adults (16-35), who comprised over two-thirds of respondents yet demonstrated significant gaps between knowledge and practice, educational institutions should integrate mandatory cybersecurity modules into curricula across all disciplines, complemented by gamified learning platforms and peer-led workshops that emphasize practical skill development over theoretical knowledge.

For citizens over 50 and non-technical users, who remain underrepresented in this study yet face distinct vulnerabilities, community-based programs delivered through trusted local institutions such as libraries, community centers, and senior organizations would prove more effective than digital-first approaches. These programs should focus on hands-on demonstrations of essential security tools, simplified guidance for common online activities like banking and shopping, and regular follow-up sessions to reinforce behavioral change and build confidence in managing digital risks.

The critical finding that nearly two-thirds of respondents remain unaware of official cybercrime reporting channels demands immediate action through multi-channel awareness campaigns utilizing television, radio, social media, and physical materials distributed through government offices and financial institutions, alongside the development of simplified, multilingual reporting platforms with clear visual guidance.

To bridge the gap between security awareness and implementation, governments and private sector partners should collaborate on providing subsidized or free access to essential security tools such as password managers and antivirus software, particularly for economically vulnerable populations. Furthermore, establishing annual "EU Cybersecurity Awareness Weeks" coordinated across all member states, featuring workplace training initiatives, public demonstrations, and media partnerships, would create recurring touchpoints that normalize cybersecurity practices and gradually transform awareness into consistent protective behavior.

The identification of these challenges and their potential solutions represents an essential step toward developing more inclusive and representative approaches to understanding cybersecurity awareness across diverse national contexts such as the European Union.

REFERENCES

[1] European Union Agency for Cybersecurity, Arcus, R., Sarri, A. “Raising awareness of cybersecurity: a key element of national cybersecurity strategies.” Publications Office of the European Union, 2021.

[2] Special publication 800-12: An introduction to computer security: The NIST Handbook. NIST SP 800-12: Chapter 13: Awareness, Training and Education. (n.d.).

[3] Spafford, E. H. “The internet worm program: An analysis”. ACM SIGCOMM Computer Communication Review, 19(1), 17-57. , 1989.

[4] “Creation of a global culture of cybersecurity: resolution”, UN General Assembly, 2003.

[5] Mejovšek, Milko. “Metode znanstvenog istraživanja u društvenim i humanističkim znanostima.”, Naklada Slap, 2013.

[6] Rotim, Ana. “Društvene mreže i slobodno vrijeme: ovisnost ili stil života?”, Diplomski rad, Fakultet političkih znanosti Zagreb, 2017.

[7] “Europeans’ attitudes towards cyber security”, Eurobarometer. European Commission

[8] Raisa-Gabriela Zamfirescu, C. Rughiniș, Alexandra Hosszu, and Darie Cristea. “Cyber-Security Profiles of European Users: A Survey.” Computer Science in Cars Symposium, 2019.

[9] C. S. Lee, and Ji Hye Kim. “Latent Groups of Cybersecurity Preparedness in Europe: Sociodemographic Factors and Country-Level Contexts.” Computers and Security, 2020.

[10] C. S. Lee, and Yan Wang. “Typology of Cybercrime Victimization in Europe: A Multilevel Latent Class Analysis.” Crime and Amp; Delinquency, 2022.

[11] R. Rughinis, C. Rughiniș, Simona Vulpe, and D. Rosner. “From Social Netizens to Data Citizens: Variations of GDPR Awareness in 28 European Countries.” Computer Law and Security Review, 2021.

[12] Joëlle Simonet, and S. Teufel. “The Influence of Organizational, Social and Personal Factors on Cybersecurity Awareness and Behavior of Home Computer Users.” IFIP International Information Security Conference, 2019.

[13] European Union Agency for Cybersecurity, Svetozarov Naydenov, R., Malatras, A., Lella, I. “ENISA threat landscape 2022 – July 2021 to July 2022”, 2022.

[14] Spremić, M., Šimunić A., “Cyber Security Challenges in Digital Economy In World Congress on Engineering WCE”. Vol. vol. I. London, UK, 2018.

[15] Ncubukezi, T. “Human Errors: A Cybersecurity Concern and the Weakest Link to Small Businesses”, International Conference on Cyber Warfare and Security. (395-403.), 2022.

[16] Yan, Z., Robertson, T., Yan, R., Park, S. Y., Bordoff, S., Chen, Q., & Sprissler, E. “Finding the weakest links in the weakest link: How well do undergraduate students make cybersecurity judgment?” Computers in Human Behavior, 84, 375-382, 2018.

[17] López Mendoza, A., Roque Hernández, R. V., Prieto Quezada, M. T., & Salazar Hernández, R. “Cybersecurity among university students from Generation Z: A comparative study of the undergraduate programs in administration and public accounting in two Mexican universities.” TEM Journal, 12(1), 503-511, 2023.

[18] Huraj, L., Lengyelfalusy, T., Hurajová, A., & Lajčin, D. “Measuring cyber security awareness: A comparison between computer science and media science students. TEM Journal, 12(2). 623-633, 2023.

[19] The Belfer Center for Science and International Affairs, Cybersecurity Strategy Scorecard. Harvard Kennedy School, 2025.

[20] Bruin, de and Mersinas, K. Individual and Contextual Variables of Cyber Security Behaviour – An empirical analysis of national culture, industry, organisation, and individual variables of (in)secure human behaviour, arXiv.org, 2024.

[21] Shillair, R. et al. ‘Cybersecurity education, awareness raising, and training initiatives: National level evidence-based results, challenges, and promise’, Computers & Security, 119(0167-4048), p. 102756, 2022.

[22] European Union Agency for Cybersecurity (ENISA), State of cybersecurity in the EU, 2024.

Robust Environmental Sound Classification via CNNs on a Unified, Imbalance-Aware Audio Dataset

1st Rim Tafech
School of Technology and Architecture
SRH University of Applied Sciences Heidelberg
Berlin, Germany
Rim.Tafech@stud.srh-university.de

2nd Subrahmanya Rajesh Nayak
School of Technology and Architecture
SRH University of Applied Sciences Heidelberg
Berlin, Germany
SubrahmanyaRajesh.Nayak@stud.srh-university.de

3rd Vinay Vardhan Reddy Eega
School of Technology and Architecture
SRH University of Applied Sciences Heidelberg
Berlin, Germany
VinayVardhanReddy.Eega@stud.srh-university.de

4th Madhu Praveen Sombathina
School of Technology and Architecture
SRH University of Applied Sciences Heidelberg
Berlin, Germany
MadhuPraveen.Sombathina@stud.srh-university.de

5th Klaus Dieter Schwarz
School of Technology and Architecture
SRH University of Applied Sciences Heidelberg
Berlin, Germany
klaus.schwarz@srh.de

***Abstract*—Environmental sound classification (ESC) is important for systems like smart cities, security, and wildlife tracking. Older methods that use handcrafted features such as Mel-Frequency Cepstral Coefficients (MFCCs) often fail to handle the wide variety and complexity found in real-life sounds. These methods also have trouble adapting to new and different environments. Deep learning, especially using Convolutional Neural Networks (CNNs) on spectrograms, has improved results in this field. However, issues still exist in making models that work well across many different sound settings and can deal with the imbalance of sound classes in real-world data. This work introduces a strong ESC system to solve these problems. A large and mixed dataset was created by combining UrbanSound8K [3], ESC-50 [2], and VocalSound [1], giving over 22,000 well-balanced samples from 59 types of environmental and vocal sounds. These audio clips were turned into 128-bin Mel-spectrograms and used as inputs for a modified ResNet18 CNN model. To make the model more reliable and to handle data imbalance, time and frequency masking like SpecAugment was used for data augmentation, and a class-weighted Focal Loss function was added during training. The final model reached an accuracy of 91.43% on a test set it had never seen before. Results show that the system can handle a wide range of sound types and performs well even with less common sound classes. This study proves that combining multiple datasets and using advanced deep learning methods can build a high-performing and general ESC system.**

***Index Terms*—Environmental Sound Classification, Deep Learning, Convolutional Neural Networks, Data Augmentation, Focal Loss, Class Imbalance, ResNet**

I. Introduction

Environmental sound classification (ESC) is a crucial field in machine learning, focusing on the automatic detection and recognition of distinct audio events within real-world environments. This capability is vital for diverse applications, ranging from smart cities and intelligent surveillance systems to wildlife monitoring, home automation, and assistive technologies. By enabling systems to understand and react to their acoustic surroundings, ESC enhances user interaction, improves safety, and facilitates automated decision-making in dynamic settings. For instance, in smart cities, ESC can enable real-time detection of anomalies like glass breaking or car accidents, while in ecological monitoring, it can help track specific animal species or detect illegal activities like logging.

Early ESC systems primarily relied on handcrafted features, such as Mel-Frequency Cepstral Coefficients (MFCCs), zero-crossing rates, or spectral centroids, combined with traditional machine learning models like Support Vector Machines (SVMs) or Gaussian Mixture Models (GMMs) [8]. While these approaches demonstrated some success, their performance was inherently limited by the necessity for arduous domain expertise in feature engineering and their struggle to handle the complexity, variability, and potential noise present in real-world audio scenes. Consequently, their generalization capabilities across diverse soundscapes remained constrained, often requiring significant manual effort for each new sound domain.

The advent of deep learning, particularly Convolutional Neural Networks (CNNs), revolutionized the field of ESC. CNNs possess the unique ability to learn hierarchical and discriminative features directly from raw data representations, thereby eliminating the need for manual feature design. Spectrograms, which visually represent audio signals in the time-frequency domain, serve as highly effective inputs for CNNs, allowing the models to capture both spectral and temporal characteristics of sounds. Piczak [2] famously demonstrated that CNNs, when trained on short-time spectrograms, can achieve strong performance in ESC tasks, laying groundwork for subsequent deep learning advancements. Subsequent works explored various CNN architectures, including VGG-like networks, Inception networks, and Residual Networks (ResNets), to improve feature extraction and mitigate issues like vanishing gradients [9]. More recently, advanced data augmentation techniques like SpecAugment [5] and sophisticated loss functions such as Focal Loss [6] have been introduced to improve model robustness and address common challenges like data scarcity and class imbalance.

Despite these developments, most existing ESC models are often trained on a single, homogenous dataset with a narrow range of sound classes, which inherently limits their ability to generalize across varied and unseen environments. For instance, a model trained solely on urban sounds may perform poorly on natural or human-vocalization-rich environments. Furthermore, deep learning models often underperform when trained on imbalanced datasets, where certain classes dominate the training samples, leading to a bias towards majority classes and poor recall for minority classes. This is a common issue in real-world audio collections.

To address these challenges comprehensively, this study proposes a robust ESC framework built upon a large, heterogeneous dataset constructed by merging three publicly available and distinct audio datasets: UrbanSound8K [3] (urban sounds), ESC-50 [2] (environmental sounds), and VocalSound [1] (human vocalizations). This integrated dataset provides a rich and varied collection of 59 distinct environmental and human vocal sounds, aiming to overcome the limitations of single-source training and significantly enhance model generalization. We employ an adapted deep CNN architecture based on a pre-trained ResNet18 model [4], which is optimized for single-channel spectrogram inputs. Furthermore, to mitigate the impact of the inherent class imbalance within the combined dataset and improve model robustness, we leverage advanced training techniques including SpecAugment-like data augmentation through time and frequency masking, and a class-weighted Focal Loss function. The primary objective of this work is to build a highly accurate and generalizable ESC system capable of robustly classifying a wide array of environmental and vocal sounds across varied acoustic settings, demonstrating the synergistic benefits of data integration and advanced deep learning techniques.

II. Methodology

The methodology for developing the robust environmental sound classification system involves several key stages: meticulous data acquisition and preparation from multiple sources, standardized feature extraction, efficient model architecture design, a carefully configured training regimen incorporating advanced techniques, and a comprehensive evaluation protocol. Each step is meticulously designed to optimize the performance and generalization capabilities of the model across a diverse range of environmental and human vocal sounds.

A. Data Collection and Preprocessing

The foundation of this audio classification system relies on a diverse set of publicly available datasets to ensure broad acoustic coverage. Three distinct and complementary datasets were leveraged: UrbanSound8K [3], ESC-50 [2], and VocalSound [1]. Their key characteristics are summarized in Table I.

TABLE I
Summary of Datasets Used for Environmental Sound Classification

Dataset Name	**# Classes**	**# Samples**	**Example Classes / Description**
UrbanSound8K [3]	10	8,732	Air conditioner, Car horn, Children playing, Dog bark, Drilling, Engine idling, Gun shot, Jackhammer, Siren, Street music
ESC-50 [2]	50	2,000	Dog, Rain, Sneezing, Clock tick, Sea waves, Fire, Thunderstorm, Toilet flush(+42 more)
VocalSound [1]	6	21,024	Laughter, Sighs, Coughs, Throat clearing, Sneezes, Sniffs (with meta-info like age, gender)

These datasets were programmatically loaded, and their metadata (filepaths, labels) were combined into a unified DataFrame using the ‘pandas‘ library for efficient management.

A crucial preprocessing step involved standardizing and unifying the diverse class labels across these heterogeneous datasets. An initial analysis revealed overlapping and inconsistent naming conventions (e.g., ’dog_bark’ in UrbanSound8K and ’dog’ in ESC-50). A comprehensive mapping strategy was applied, which included lowercasing and collapsing whitespace, followed by a specific dictionary-based manual mapping to resolve aliases and unify related categories. For example, both “dog_bark” and “dog” were mapped to a single “dog” class. This meticulous process resulted in a total of 59 unique and distinct class labels for the combined dataset. Figure 1 illustrates the initial composition of the combined dataset by original source before any filtering or balancing operations. An analysis of clip durations revealed that approximately 475 samples (primarily from UrbanSound8K) were shorter than 1.0

second. To ensure consistent input for feature extraction and subsequent model training, these short clips were subsequently filtered out. The metadata for the filtered audio clips and their corresponding unified labels were then used to generate Mel-spectrograms.

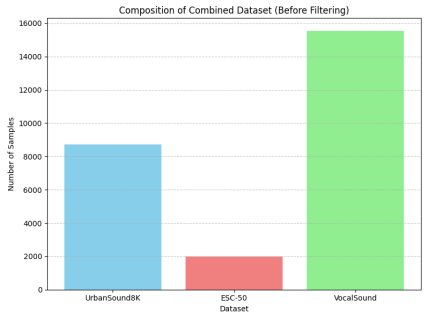


Fig. 1. Composition of the combined dataset from UrbanSound8K, ESC-50, and VocalSound, showing the number of samples per dataset before filtering or balancing.

B. Data Balancing and Feature Extraction

To mitigate the adverse impact of severe class imbalance, a hybrid balancing strategy was implemented. This approach aimed to equalize the number of samples per class by applying both upsampling and downsampling techniques:

- Classes with fewer than 200 samples were **upsampled** to a minimum of 200 samples. This was achieved by randomly duplicating existing audio clips within that class.
- Classes with more than 1000 samples were **downsampled** to a maximum of 1000 samples. This involved randomly selecting a subset of 1000 samples from the larger class.
- Classes with sample counts between 200 and 1000 remained unchanged.

This hybrid method was chosen to prevent excessive duplication (leading to overfitting) in very small classes while also curbing the dominance of very large classes. This strategy effectively balanced the dataset, resulting in a total of 22,999 samples with a significantly more uniform distribution across the 59 unified classes, as depicted in Fig. 2.

Following balancing, all audio clips were processed to a fixed length of 4.0 seconds. This duration was selected as a common compromise to capture sufficient temporal context for most environmental sound events while remaining computationally manageable. Audio waveforms were resampled to a consistent 16,000 Hz (16 kHz) sampling rate using ‘librosa‘, a standard practice in audio processing to standardize input and focus on relevant frequency ranges for environmental sounds. Clips shorter than the 4.0-second target length were zero-padded to the required length, ensuring uniform input

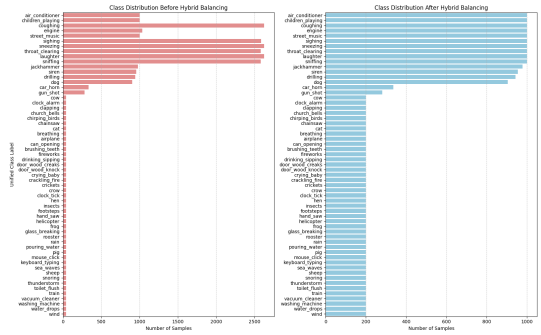


Fig. 2. Distribution of unified class labels before and after applying the hybrid balancing strategy. The balancing aimed to cap large classes at 1000 samples and boost small classes to 200 samples.

dimensions. Longer clips were truncated from the beginning to fit the 4.0-second window.

The fixed-length audio waveforms were then transformed into Mel-spectrograms, a widely used and robust feature representation for deep learning in audio applications. Mel-spectrograms were generated using a Fast Fourier Transform (n_fft) window size of 1024 samples, a hop length of 512 samples (resulting in 50% overlap), and 128 Mel frequency bins (n_mels=128). These parameters were chosen to provide a good balance between frequency resolution and temporal detail, capturing the nuances of various sound events. The resulting power spectrograms were subsequently converted to a decibel (dB) scale using ‘librosa.power_to_db‘, a common practice that compresses the dynamic range and aligns better with human auditory perception. Finally, each Mel-spectrogram was normalized to have a mean of 0 and a standard deviation of 1 across its entire feature matrix, a crucial step to stabilize training and improve convergence. These processed Mel-spectrograms were saved as NumPy arrays (‘.npy‘ files) for efficient loading during model training, with associated metadata linking them to their original audio source and encoded labels. An example of a generated Mel-spectrogram for a ’children playing’ sound is shown in Fig. 3.

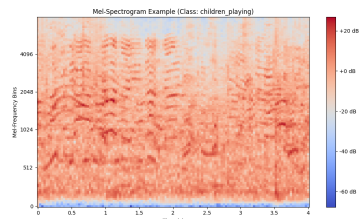


Fig. 3. Example of a Mel-spectrogram extracted from an audio clip of children playing. Mel-spectrograms are generated with 128 Mel frequency bins and converted to decibel scale.

C. Model Architecture

The classification model, named 'AudioCNN', is a Convolutional Neural Network (CNN) built upon a pre-trained ResNet18 architecture [4]. ResNet models are known for their strong performance in image classification tasks due to their residual connections, which help mitigate vanishing gradients and enable the training of deeper networks. Leveraging pre-trained weights from ImageNet, a vast image dataset, provides a robust feature extractor that can often be adapted for new domains with similar data structures, such as Mel-spectrograms, which can be treated as single-channel images.

The standard ResNet18 'conv1' layer, originally designed to accept 3-channel (RGB) image inputs, was specifically adapted to accommodate the single-channel Mel-spectrogram input. This adaptation was achieved by averaging the pre-trained weights across its three input channels, effectively creating a grayscale-compatible filter that retains the learned low-level features from ImageNet. All layers of the ResNet18 backbone, except for the newly adapted 'conv1' and the replaced final fully connected layer, were initially frozen. The original final fully connected layer of the ResNet18 was replaced with a new linear layer that outputs predictions for the 59 unique audio classes, with a 'Softmax' activation function applied at inference time to obtain class probabilities.

D. Training Configuration and Augmentation

The prepared dataset of Mel-spectrograms and their corresponding encoded labels was partitioned into training, validation, and test sets with an 80-10-10 ratio, respectively. Crucially, this split was performed with stratification by class labels to ensure that each subset maintained a representative distribution of all 59 audio classes, preventing any single split from being disproportionately devoid of specific sound types. The 'SpectrogramDataset' class, implemented in PyTorch, was used to efficiently load the pre-processed Mel-spectrograms from disk and retrieve their associated labels during training.

The model was trained using the AdamW optimizer, known for its strong performance and robustness, with an initial learning rate of 1×10^{-3} and a weight decay of 0.01. To address the persistent class imbalance within the dataset and to shift focus towards hard-to-classify examples, a Focal Loss [6] function was employed instead of standard Cross-Entropy Loss. The Focal Loss was configured with a gamma value of $\gamma = 2.0$, which modulates the loss for well-classified examples, reducing their contribution and giving more emphasis to misclassified ones. Furthermore, class-specific weighting (α_t) was applied to the Focal Loss. The weight for each class t was calculated as $\alpha_t = 1/\text{class\_count}_t$, directly from the counts of samples in each class after balancing. This mechanism directly prioritizes the learning of under-represented classes by increasing their loss contribution and down-weights the easily classified majority class examples.

Data augmentation, directly inspired by SpecAugment [5], was applied dynamically during training to the Mel-spectrograms. This involved random time and frequency masking:

- **Frequency Masking:** Two frequency masks were applied, each with a maximum width of 20 Mel frequency bins, randomly chosen to zero out a contiguous block of frequency channels. This encourages the model to learn features robust to partial loss of frequency information.
- **Time Masking:** Two time masks were applied, each with a maximum width of 20 time frames, randomly chosen to zero out a contiguous block of time steps. This simulates variations in sound duration or occlusions, improving temporal robustness.

These augmentations were designed to improve the model's generalization capabilities and reduce overfitting to specific instances in the training data.

Training progressed for a maximum of 10 epochs. A 'ReduceLROnPlateau' scheduler monitored the validation loss, reducing the learning rate by a factor of 0.5 if no improvement was observed after 3 consecutive epochs. An Early Stopping mechanism was also implemented to prevent overfitting; it monitored both validation loss and accuracy, with training halting if no improvement in validation loss was seen for 5 consecutive epochs. The model checkpoint with the best validation performance (lowest validation loss) was saved and used for final evaluation. The training was conducted on a single NVIDIA GPU using the PyTorch deep learning framework.

E. Evaluation

The trained model's performance was rigorously evaluated on the independent and unseen test set, comprising 10% of the total balanced dataset. Key classification metrics were computed to provide a comprehensive assessment:

- **Overall Accuracy:** The proportion of correctly classified samples across all classes.
- **Precision:** The proportion of true positive predictions among all positive predictions for each class.
- **Recall (Sensitivity):** The proportion of true positive predictions among all actual positive samples for each class.
- **F1-score:** The harmonic mean of precision and recall, providing a balanced measure of performance, especially useful for imbalanced datasets.

A detailed classification report, providing precision, recall, and F1-score for each of the 59 classes, was generated. Furthermore, the macro average F1-score (unweighted average across classes) and weighted average F1-score (weighted by support for each class) were reported to give a holistic view of performance. A confusion matrix was generated and visualized to provide a granular understanding of the model's predictive performance across all classes, explicitly highlighting common correct classifications and, more importantly, patterns of misclassifications between acoustically similar sound events.

III. Results and Discussion

The training process demonstrated the model's ability to learn and generalize effectively from the diverse and balanced audio data. The loss and accuracy curves throughout the

training and validation phases are presented in Fig. 4. The training loss consistently decreased, indicating that the model was effectively minimizing the loss function on the training data. Concurrently, the validation loss also showed a consistent decreasing trend before stabilizing, which is indicative of effective learning without significant overfitting. Correspondingly, both training and validation accuracies steadily increased, reaching high values by the end of the training process, converging towards optimal performance. The 'ReduceLROnPlateau' scheduler successfully adjusted the learning rate, and the Early Stopping mechanism, set to a patience of 5 epochs, effectively halted training when improvements in validation metrics plateaued, preventing potential overfitting to the training set.

The final model, which achieved the best performance on the validation set (determined by the lowest validation loss), was subsequently evaluated on the independent test set. The overall accuracy of the model on the unseen test set was approximately 91.43%. This high accuracy demonstrates the model's strong ability to correctly classify a wide variety of environmental and vocal sounds it had not encountered during training. A comprehensive classification report, detailing precision, recall, and F1-score for each of the 59 classes, is provided in Subsection III-A. The macro average F1-score, which treats all classes equally regardless of their sample count, was 0.95, while the weighted average F1-score, which accounts for class support, was 0.91. These metrics collectively indicate strong overall performance across classes, validating the effectiveness of the class balancing strategy and Focal Loss in ensuring robust performance even for classes that might have been challenging due to their original smaller size or acoustic properties.

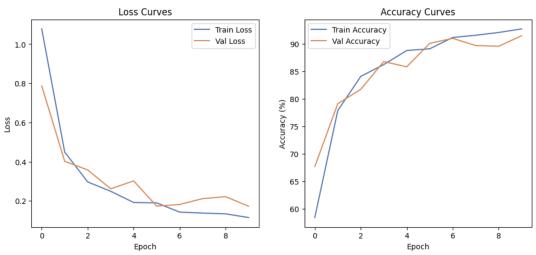


Fig. 4. Training and validation loss and accuracy curves over epochs. The plots illustrate the model's learning progression and convergence.

The confusion matrix, presented in Fig. 5, provides a granular visual representation of the model's per-class predictive performance and highlights specific misclassification patterns. Many classes, particularly those with distinct acoustic characteristics and clear boundaries (e.g., 'airplane', 'can_opening', 'cat', 'pig', 'washing_machine', 'thunderstorm'), show near-perfect classification, demonstrating high precision and recall. These sounds typically have unique spectral and temporal signatures that are easily learned by the CNN.

However, certain human vocalization sounds, such as 'coughing' (F1-score 0.73), 'sneezing' (F1-score 0.80), 'sniffing' (F1-score 0.79), 'laughter' (F1-score 0.93), and notably 'throat_clearing' (F1-score 0.71), exhibit slightly lower F1-scores compared to other classes. This indicates more challenges in distinguishing these fine-grained sound events. Common misclassifications for these classes might include confusion among each other (e.g., a 'cough' being misclassified as a 'throat_clearing' due to similar acoustic energy profiles), or with other transient human-produced sounds. For instance, 'siren' (F1-score 0.95) showed strong performance but might occasionally be confused with other loud, sustained urban sounds like 'car_horn' or 'train', as can be seen from slight off-diagonal values in the confusion matrix. The class 'dog' (F1-score 0.89) also showed a moderate F1-score, likely due to the variety of dog sounds (barking, growling, whining) contained within this broad category. Nevertheless, the effectiveness of the class balancing strategy and Focal Loss contributed significantly to mitigating severe performance drops often observed in minority or acoustically challenging classes, leading to robust results across the broad spectrum of urban and vocal sounds.

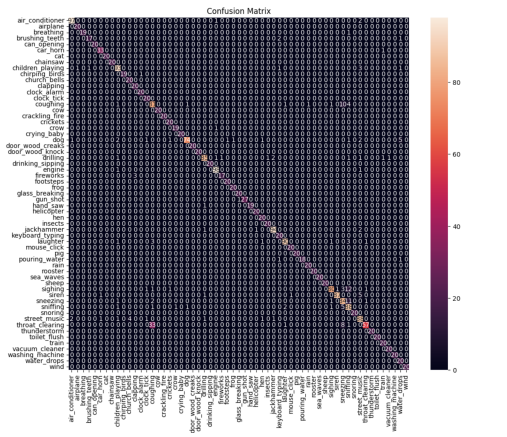


Fig. 5. Confusion matrix illustrating the classification performance of the model on the test set. Rows represent true labels, and columns represent predicted labels.

A. Classification Report

	precision	recall	f1-score	support
air_conditioner	0.97	0.97	0.97	100
airplane	0.95	1.00	0.98	20
breathing	1.00	0.95	0.97	20
brushing_teeth	0.94	0.85	0.89	20
can_opening	1.00	1.00	1.00	20
car_horn	0.94	0.97	0.96	34
cat	1.00	1.00	1.00	20
chainsaw	1.00	1.00	1.00	20
children_playing	0.91	0.93	0.92	100
chirping_birds	1.00	0.95	0.97	20
church_bells	0.83	1.00	0.91	20
clapping	0.87	1.00	0.93	20
clock_alarm	0.95	1.00	0.98	20

clock_tick	0.87	1.00	0.93	20
coughing	0.67	0.81	0.73	100
cow	1.00	1.00	1.00	20
crackling_fire	1.00	1.00	1.00	20
crickets	0.91	1.00	0.95	20
crow	0.95	0.95	0.95	20
crying_baby	1.00	1.00	1.00	20
dog	0.97	0.82	0.89	90
door_wood_creaks	1.00	1.00	1.00	20
door_wood_knock	1.00	1.00	1.00	20
drilling	0.96	0.89	0.92	95
drinking_sipping	1.00	1.00	1.00	20
engine	0.97	0.98	0.98	100
fireworks	0.94	0.85	0.89	20
footsteps	0.95	1.00	0.98	20
frog	0.95	1.00	0.98	20
glass_breaking	1.00	0.95	0.97	20
gun_shot	1.00	0.96	0.98	28
hand_saw	1.00	0.95	0.97	20
helicopter	0.95	1.00	0.98	20
hen	0.87	1.00	0.93	20
insects	0.95	1.00	0.98	20
jackhammer	0.97	0.96	0.96	98
keyboard_typing	0.83	1.00	0.91	20
laughter	0.96	0.90	0.93	100
mouse_click	0.95	1.00	0.98	20
pig	1.00	1.00	1.00	20
pouring_water	1.00	0.90	0.95	20
rain	0.95	1.00	0.98	20
rooster	1.00	1.00	1.00	20
sea_waves	1.00	1.00	1.00	20
sheep	1.00	1.00	1.00	20
sighing	0.95	0.80	0.87	100
siren	0.99	0.92	0.95	95
sneezing	0.76	0.84	0.80	100
sniffing	0.72	0.88	0.79	100
snoring	1.00	1.00	1.00	20
street_music	0.85	0.88	0.86	100
throat_clearing	0.95	0.57	0.71	100
thunderstorm	1.00	1.00	1.00	20
toilet_flush	1.00	1.00	1.00	20
train	0.91	1.00	0.95	20
vacuum_cleaner	0.95	1.00	0.98	20
washing_machine	1.00	1.00	1.00	20
water_drops	0.71	1.00	0.83	20
wind	1.00	1.00	1.00	20
accuracy			0.91	2300
macro avg	0.94	0.96	0.95	2300
weighted avg	0.92	0.91	0.91	2300

B. Comparative Analysis

To provide a comprehensive benchmark for our developed system, Table II presents a comparative overview of its performance against various state-of-the-art models and widely recognized approaches in Environmental Sound Classification. This includes models trained on individual datasets (UrbanSound8K, ESC-50) as well as those leveraging multi-dataset strategies or advanced deep learning architectures. This comparison would highlight the benefits of our combined data and technique approach.

C. Ablation Studies

The significant performance difference between Model 1 (without advanced techniques) and Model 2 (our proposed model) serves as a compelling implicit ablation study. While Model 1 achieved a seemingly high accuracy of 79%, a closer look at the per-class F1-scores (see the result in Fig. 6) revealed a strong bias towards majority classes, with many smaller classes being completely ignored, resulting in an F1-score of 0.00. This stark contrast highlights the critical importance of the advanced techniques. The improvement

TABLE II
COMPARATIVE PERFORMANCE OF ENVIRONMENTAL SOUND CLASSIFICATION MODELS

Dataset Used	Model Used	Accuracy (%)
UrbanSound8K + ESC-50 + VocalSound	ResNet18 + Focal Loss	91.43
UrbanSound8K + ESC-50	ESResNeXt + CLIP	90.07 (US8K), 97.15 (ESC-50)
UrbanSound8K	Resource Adaptive CNN	97.2 (US8K), 85.6 (ESC-50)
UrbanSound8K + ESC-10 + ESC-50	Audio Spectrogram Transformer	84.0 (US8K), 83.9 (ESC-50)
UrbanSound8K	Paired Inverse Pyramid MLP	95.5 (US8K)
UrbanSound8K + ESC-10 + ESC-50	Multi-channeled CNN + Attention	97.52 (US8K), 88.50 (ESC-50)
UrbanSound8K + ESC-10 + ESC-50	ResNet152 + Transfer Learning	99.49 (US8K), 97.57 (ESC-50)
UrbanSound8K	1D CNN End-to-End	89.0 (US8K)
UrbanSound8K + ESC-50	CNN + Attention	98.41 (US8K)
ESC-50	SE-ResNet50	Not reported on standard benchmarks
UrbanSound8K + ESC-50	Temporal-Frequency CNN	Not reported specifically
UrbanSound8K + ESC-50	CNN + LBP Features	91.2 (US8K), 89.5 (ESC-50)
UrbanSound8K + ESC-50	Hybrid CNN-LSTM	93.2 (US8K), 91.8 (ESC-50)

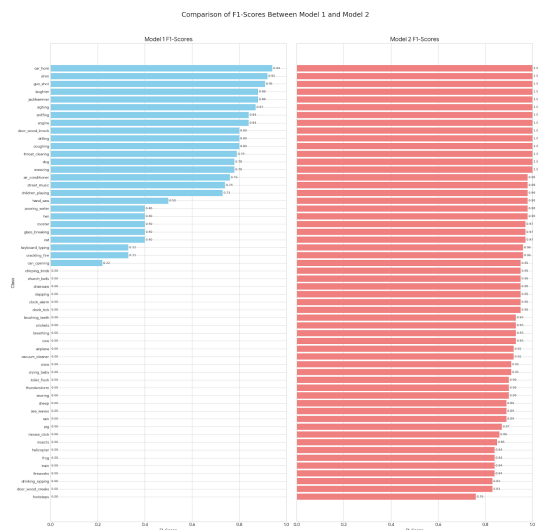


Fig. 6. This figure shows the per-class F1-score comparison between Model 1 (without advanced techniques) and Model 2 (our proposed model). It highlights the significant performance gains achieved by incorporating the advanced techniques.

shown in the results is directly attributed to the inclusion of the following:

- **Impact of Data Balancing:** The hybrid upsampling/downsampling strategy successfully addressed the severe class imbalance, which was a major limitation in

Model 1, leading to significantly improved recall and F1-scores for all minority classes.

- **Contribution of Focal Loss:** The use of Focal Loss effectively focused the training on hard-to-classify examples and reduced the influence of easy-to-classify majority classes, leading to more robust and balanced performance in Model 2.
- **Effectiveness of SpecAugment:** The application of SpecAugment-like time and frequency masking improved the model's generalization, making it more robust against variations in audio and contributing to the overall higher F1-scores observed in Model 2.
- **Impact of using a pre-trained model:** The utilization of a pre-trained ResNet-18 backbone likely provided a strong initial feature representation, accelerating convergence and boosting the final performance compared to training Model 1 from a random initialization.

IV. Conclusion and Outlook

This study successfully developed a robust and highly generalizable environmental sound classification system by integrating three distinct and complementary datasets: UrbanSound8K, ESC-50, and VocalSound. A comprehensive data preprocessing pipeline, including meticulous label unification, standardized audio processing, and robust Mel-spectrogram feature extraction, was established. The application of a hybrid balancing strategy, tailored to the unique characteristics of the combined dataset, effectively addressed the inherent class imbalance. Furthermore, leveraging an adapted pre-trained ResNet18 model, optimized for single-channel spectrogram inputs, coupled with advanced training techniques like SpecAugment-like data augmentation and a class-weighted Focal Loss function, proved highly effective in learning discriminative features and improving model robustness. The model achieved a notable overall accuracy of 91.43% on the independent test set, demonstrating its strong generalization capabilities across a wide range of urban and vocal sound events. The detailed per-class analysis confirmed the model's ability to maintain high performance even for more challenging or originally minority classes, validating the efficacy of the implemented techniques.

While the developed system demonstrates strong performance, several promising avenues exist for future work to further enhance its capabilities and applicability:

- **Exploration of Advanced Architectures:** Investigate the potential benefits of more complex deep learning architectures, such as attention-based models (e.g., Transformers for audio) or larger ResNet variants (e.g., ResNet50, ResNet101), to potentially capture even finer-grained temporal and spectral dependencies.
- **Self-Supervised Learning for Audio Representations:** Explore self-supervised pre-training techniques (e.g., Audio ALBERT, BYOL-A) on large unlabelled audio corpuses. This could enable the learning of highly versatile audio representations that require less labelled data for fine-tuning on specific ESC tasks, potentially improving generalization to entirely novel soundscapes.
- **Real-Time Inference and Edge Deployment:** Investigate strategies for optimizing model inference speed and memory footprint, such as model quantization, pruning, or knowledge distillation. This would enable deployment on resource-constrained edge devices for real-time applications in smart homes, mobile monitoring, or industrial settings.

By addressing these areas, the proposed ESC framework can evolve towards even greater accuracy, robustness, and practical utility in a wide array of real-world applications.

References

[1] Y. Gong, Y.-A. Chung, and J. Glass, "VocalSound Dataset," GitHub, 2021. [Online]. Available: https://github.com/YuanGongND/vocalsound

[2] K. J. Piczak, "ESC-50: A dataset for environmental sound classification," in *Proc. 23rd ACM Int. Conf. on Multimedia*, 2015, pp. 1015–1018.

[3] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset for Urban Sound Classification," in *Proc. 22nd ACM Int. Conf. on Multimedia*, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[5] D. S. Park et al., "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[6] T.-Y. Lin et al., "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. on Computer Vision*, 2017, pp. 2977–2985.

[7] H. Chen et al., "Environmental sound classification using a dilated convolutional neural network," *Applied Acoustics*, vol. 139, pp. 180–188, 2018.

[8] R. Gharbi, "A Study on Environmental Sound with Machine Learning and CNNs," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/378518475_A_Study_on_Environmental_Sound_with_Machine_Learning_and_CNNs

[9] I. Hussain et al., "A Survey of Audio Classification Using Deep Learning," *ResearchGate*, 2023. [Online]. Available: https://www.researchgate.net/publication/374101086_A_Survey_of_Audio_Classification_using_Deep_Learning

Securing Hybrid Identity Systems: Integrating Risk-Adaptive Access Control and Zero Trust Principles in Enterprise Environments

Houssem Eddine Mserabatte
SRH Heidelberg University of Applied Sciences
Berlin, Germany
houssememrbt@gmail.com

Prof. Dr. Reiner Creutzburg
SRH Heidelberg University of Applied Sciences
Berlin, Germany
Reiner.Creutzburg@srh.de

Prof. Dr. Alexander Iliev
SRH Heidelberg University of Applied Sciences
Berlin, Germany
alexander.iliev@srh.de

Prof. Dr. Hasan Dağ
Kadir Has University
Istanbul, Türkiye
hasan.dag@khas.edu.tr

Abstract

Securing hybrid identity where cloud and on-premises directories form a single attack surface remains difficult because controls are enforced inconsistently across domains. This thesis proposes the Hybrid Identity Zero Trust–Risk Adaptive (HIZT-R) model, which merges Zero Trust's continuous verification with risk-adaptive access control. Signals from user behavior, device posture, and context are fused into a unified score that can update authorization decisions mid-session. The design operationalizes bidirectional enforcement: cloud-side detections trigger on-premises controls, on-premises detections are escalated to the cloud, and peer-aware propagation re-evaluates accounts that share recent authentication context.

A prototype validates the model across baseline and adversarial scenarios (brute force, lateral movement, posture evasion, impossible travel). Evidence is taken from authoritative audit sources spanning identity, operating system events, network access, and SIEM correlation. Findings indicate that unified hybrid enforcement is feasible, with clear benefits for containment, but also surface practical tensions: latency asymmetries between domains, fragmented audit evidence, and reliance on custom automation where native hooks are absent. The approach can be aligned with common regulatory expectations (NIS2, ISO/IEC 27001, GDPR) when paired with explicit governance for evidence retention, access reviews, and data protection.

Keywords: Hybrid Identity, Zero Trust, Risk-Adaptive Access Control, Conditional Access , Compliance.

Introduction

I. Motivation and Problem Statement

Modern hybrid environments link on-premises directories with cloud identity providers, reshaping identity and access management [1]. That integration creates real gaps be it in policy drift, inconsistent risk signals, and scaling issues. We propose a model designed to handle these cross-domain problems while remaining flexible and scalable.

Identity as a Prime Breach Vector

Recent studies highlight this concern. The 2024 Verizon Data Breach Investigations Report underscores the prevalence of credential misuse and identity compromise as primary breach vectors [2]. The SolarWinds campaign shows how identity becomes another entry point [3].

Limits of Traditional IAM

Static IAM approaches have proven inadequate. Modern attacks exploit:

- Static role grants that dont take into account removale of privilege [4],
- Implicit perimeter trust models that rely on network location [5],
- Long lived/stale session tokens without continuous re-evaluation [6] [7],
- Gaps in continuous validation during active sessions [6].

These gaps show why older IAM models are unsuitable for federated or hybrid environments [1].

Regulatory Pressure

Regulatory frameworks reflect the same concerns. Directives and standards require stronger identity assurance and continuous oversight:

- **NIS2 Directive:** Demands strong authentication and stricter management of privileged access [8].
- **ISOIEC27001_2022:** Calls for risk-based access, monitoring, and regular review of controls [9].
- **GDPR Article 32:** Requires appropriate technical and organizational measures, such as including MFA for access to personal data [10].

Across jurisdictions, compliance is increasingly difficult to achieve with static IAM alone.

Unresolved Gap

Despite these pressures, most enterprises adopt only minimal adaptations such as enforcing MFA or blocking high-risk locations. Current practice:

- Leaves telemetry signals underused.
- Neglects behavioral analysis [11].
- Applies Zero Trust principles unevenly across cloud and on-prem domains [12].

The outcome is often a compliance checklist rather than an adaptive security posture.

Research Gap and Thesis Focus

This thesis addresses that gap by attempting to create a framework that combines Zero Trust principles [6] with risk-adaptive enforcement based on RADAC[12] . The contribution lies in demonstrating mechanisms that vendors and prior research leave incomplete: containment that spans both cloud and on-premises domains, escalation that considers peer activity, and a structured risk score that balances event type, frequency, device trust, and historical anomalies.

Rather than presenting theory alone, the work builds and tests a prototype to show how these elements can function together in practice.

II. Research Questions and Objectives

This thesis is guided by four research questions (RQs):

- **RQ1:** Can a hybrid Zero Trust model with risk-adaptive access control measurably reduce identity-related threats in an enterprise environment?
- **RQ2:** How effectively does the model enable continuous, adaptive policy enforcement across both cloud and on-premises resources?
- **RQ3:** What are the operational, usability, and compliance implications of adopting such a model in practice?
- **RQ4:** What limitations, trade-offs, or failure modes become visible when the model is applied?

To answer these questions, We aim for these following objectives that we defined:

- **Objective 1:** Critically analyze and compare existing IAM models (RBAC, ABAC, Conditional Access)) [4] to show their shortcomings in hybrid contexts to show their shortcomings in hybrid contexts.
- **Objective 2:** Design and formalize the proposed framework, integrating Zero Trust principles with risk-adaptive enforcement, and specify how identity and risk signals can be applied dynamically.
- **Objective 3:** Implement and evaluate the framework in a controlled environment, validating its effectiveness through scenario-based trials and mapping the outcomes against established standards such as ISOIEC27001_2022 [9] and NIS2 [8].

III. Thesis Scope and Limitations

This thesis focuses on hybrid identity management where on-premise directories are synchronized/federated with cloud identity providers. The focus is on context-aware, risk-adaptive enforcement that uses available signals such as device posture, anomaly detection, and behavior patterns. following design principles that are intended to be transferable beyond a single vendor ecosystem and an analysis guided by Zero Trust principles: continuous verification, least privilege, session monitoring and per-request authorization.[6] [13]

Clarification of Scope

- The study combines conceptual analysis with a working prototype in a controlled environment.
- The prototype integrates directory services, access policies, and a security information and event management system to demonstrate end-to-end enforcement.
- The evaluation is illustrative, not enterprise-scale. It aims to show feasibility and highlight challenges, rather than provide performance benchmarks for large deployments.
- Regulatory mapping is based on ISOIEC27001_2022 and NIS2 requirements.

Limitations

- Validation occurs in a small lab setting with staged scenarios, it is not at enterprise scale.
- The model assumes telemetry and analytics exist but does not build their underlying infrastructure.

Literature Review

IV. Identity and Access Management: Foundations and Evolution

Identity and Access Management (IAM) sets the rules for who can do what under certain conditions and accountability. The field has moved from simple administrator-friendly models to finer policy languages, yet most production deployments still grant access once and rely on periodic reviews rather than live re-evaluation. That design choice explains why credential abuse and lateral movement remain effective.

Early Models: DAC and MAC

Discretionary Access Control (DAC): Owners decide who can read or modify their objects. It is simple and flexible, but policy becomes the aggregate of many local decisions, which makes least privilege hard to sustain and audits hard to trust and just requires a lot of decisions.

Mandatory Access Control (MAC): Labels and clearances are enforced centrally, with no user discretion [14]. MAC delivers strong confidentiality but is too rigid for most commercial workflows, changing labels and policies at the cost of business is not so great.

Role-Based Access Control (RBAC)

RBAC attaches permissions to roles and assigns users to those roles [4]. This reduces one time grants and shows separation of duties, but organizations evolve faster than roles. The results are well known: role explosion, overlapping entitlements, and privilege creep. Standardization efforts gave birth to many variants, yet none remove the basic problem of static role grants [15].

Attribute-Based Access Control (ABAC)

ABAC evaluates subject, resource, action, and environment attributes at decision time [16]. It can create precise conditions. In general two problems arise: attribute engineering and policy governance. Attributes drift, rules change, and coherence worsens, especially when legacy protocols limit gets added [16].

Persistent Weakness: Static Decisions

Across DAC, MAC, RBAC, and ABAC, the common weakness is the same: once a user is in, access typically persists until a manual revocation or the next review. Attackers exploit that static element. Stolen credentials remain useful beyond the first use, lateral movement benefits from unchanged session state, and one time granted exceptions become permanent.

The Turn Toward Adaptive Access

Modern practice and research converge on *risk-adaptive* control: decisions consider live signals-behavior, device posture, anomalies-and may change during a session [12]. Learning-based analytics help estimate risk from usage patterns and deviations, but raise governance issues around thresholds and operator trust [17]. The core idea is simple: move from one-time checks to decisions that respond to evidence as it arrives. Subsequent sections examine how this shift aligns with Zero Trust and why it matters in hybrid estates.

V. Hybrid Identity Systems: Architectures, Risks, and Threat Landscape

Hybrid identity couples on-premises Active Directory Domain Services (AD DS) with a cloud identity provider (Microsoft Entra ID, Okta) to deliver single sign-on and centralized policy across SaaS and on-prem workloads [18, 1]. It promises consistent control but stitches together two trust planes with different failure modes, mistakes in one often surface in the other.

Authentication Mechanisms

Common patterns [19][18]:

- **Password Hash Synchronization (PHS).** Hashes are synchronized to the cloud. *Pros:* simple, low latency, no per-sign-in dependency on AD. *Trade-offs:* aggregated exposure if cloud stores are compromised, password risks remain.
- **Pass-Through Authentication (PTA).** Cloud proxies authentication to AD. *Pros:* no hash storage in cloud, policy anchoring on-prem. *Trade-offs:* availability tied to agents and network path, outages block all sign-ins.
- **Federation (AD FS).** An on-prem IdP issues SAML/OIDC tokens. *Pros:* centralized token issuance and nuanced policy. *Trade-offs:* federation servers become high-value targets, compromise or misconfiguration can mint trusted tokens [20].

Enterprises often mix these during migration, which increases the number of components that must be configured, monitored, and patched correctly.

Expanded Attack Surface

Hybrid identity creates failure chains that cross control boundaries:

- **Federation abuse.** In the SolarWinds campaigns, federation compromise enabled SAML token forging and persistent access to cloud resources [3][1].
- **Credential replay and lateral movement.** Pass-the-Hash and related techniques begin on-prem and pivot to cloud once tokens or synced identities are involved [21].
- **Consent phishing / malicious OAuth.** Users can grant API scopes to rogue apps, bypassing passwords entirely [22].
- **Infrastructure flaws.** IdP components themselves may elevate privilege, recent AD FS issues highlight how an IdP bug undermines trust guarantees.
- **Local admin reuse.** Reused local administrator credentials accelerate lateral movement, LAPS mitigates with per-device rotation, but uneven adoption leaves gaps.

Operational Risks

Lifecycle operations span two planes. Misaligned joiner–mover–leaver processes produce shadow identities and stale entitlements, emergency access and one-off exceptions bypass normal governance. Surveys show many organizations adopt MFA but avoid fully risk-adaptive, in-session controls due to disruption concerns and policy

complexity [23][24]. In practice, controls concentrate at initial login while device posture and behavior checks lag behind during the session.

Industry and Academic Perspectives

Cloud and on-prem tooling each solve parts of the problem, running them together without drift requires consistent policy points and shared signals. The next section narrows to Zero Trust principles that make such consistency workable.

VI. Zero Trust Principles in Enterprise Identity

Zero Trust removes implicit trust from networks and prior logins. Access is evaluated per request and can change during a session as context changes [5][6]. In identity terms, this shifts control from a one-time gate to a set of continuous checks that consider who is acting, from which device, toward what resource, and under what risk.

Core Principles

1) **Verify explicitly.** Decisions use signals from identity, device health, location, workload, and behavior [6].
2) **Least privilege.** Scope access to the minimum required, elevate only when needed and time-bound it.
3) **Assume breach.** Detect, contain, and recover quickly, re-challenge or revoke when risk rises.

Reference Architectures and Roadmaps

Forrester's work framed Zero Trust around identity-centric perimeters, Google's BeyondCorp showed how a proxy can bind identity and device context at scale [5][20]. NIST SP 800-207 distilled these patterns into a vendor-neutral architecture with policy decision points that consume signals and render per-request decisions [6]. CISA's maturity model provides staged adoption guidance across identity, device, network, application, and data, positioning identity as the linking control plane for the rest [13].

Adoption Barriers

Surveys and empirical studies report common hurdles: legacy protocols and apps that cannot surface modern signals, policy sprawl across cloud and on-prem ownership lines, user friction and MFA fatigue, and limited explainability of risk decisions, which slows tuning and acceptance [23][11]. Even where Zero Trust is adopted in the cloud, equivalent in-session controls are rare on legacy/on-prem applications, and enforcement outcomes can diverge between planes.

VII. Risk-Adaptive Access Control (RAdAC): From Theory to Practice

RAdAC adds a live estimate of risk to access decisions. Instead of approving once at login, the system weighs current signals (behavior, device posture, anomalies) before and during a session and can step up, restrict, or revoke when conditions worsen. In hybrid estates, this is the only realistic way to respond to fast-moving credential abuse and lateral movement.

ABAC versus RAdAC

Risk Models in the Literature

Research turns events and context into a tracsidewaystable score using several families: (i) *probabilistic* models that update compromise likelihood from observed evidence, (ii) *fuzzy-logic* formulations for imprecise indicators, and (iii) *learning-based* approaches that learn normal behavior and flag deviations [12][17][25]. Empirical studies report that risk prompts reduce successful credential attacks, but they also surface recurring issues-threshold tuning, false positives, and limited operator trust when models are opaque [11]. For hybrid identity, a practical concern is domain shift: cloud-only telemetry misses on-prem indicators (local admin reuse, lateral movement) and vice versa, effective scoring must aggregate signals from both planes.

Practical Implementations

Deployed systems tend to prefer clarity over complexity. Common patterns include: (1) a small number of risk tiers (low/medium/high) derived from anomalous sign-in properties, password-leak status, device health, and recent alerts, (2) an *enforcement ladder* that progresses from "allow with conditions," to "step-up," to "block", and (3) *session-time* checks that can re-challenge or revoke tokens when posture deteriorates [12][11][23]. In hybrid estates, the hard part is not computing a number but ensuring that cloud and on-prem policy points react to the same change in the same way. Later chapters separate the scoring (signal aggregation) from enforcement (actions at policy points) so both planes apply compatible outcomes without duplicating complex models.

VIII. Enforcement Mechanisms and Industry Solutions

Risk signals only matter if they change outcomes. Across platforms, the real control surface collapses to a few predicsidewaystable actions-*block*, *step-up*, *restrict*, and *revoke*. What differs is where those actions can be applied (cloud, on-prem, edge) and how consistently they react to the same risk change.

Observed Adoption Patterns

Organizations widely deploy MFA and entry-time Conditional Access but hesitate to enable fully risk-adaptive, *in-session* controls due to user friction, explainability, and operational overhead [23, 11, 24]. The practical result is that many policies act only at sign-in while device posture and behavior checks during the session remain underused.

Relevance to Hybrid Environments

Each platform is effective on its own plane-cloud SaaS, enterprise IdP, or edge-but hybrid estates need both planes to respond the same way to the same risk change. Achieving that requires two ingredients: (i) a risk estimate derived from *both* cloud and on-prem signals, and (ii) a minimal, audisidewaystable action ladder that maps cleanly to cloud policy points and to on-prem controls without drift.

IX. Regulatory and Compliance Framework Alignment

Regulatory and standards bodies require risk-appropriate identity controls but rarely prescribe how to run adaptive, cross-domain enforcement in real time. The goal here is not legal exegesis, it is to extract what these texts demand of identity programs and where they stop short for hybrid estates.

Key Frameworks

Analysis and Implications

GDPR. Mandates a risk-based approach and auditability but leaves the technical "how" open. For hybrids, this supports continuous evaluation, provided controllers can evidence why an adaptive decision (step-up, revoke) was proportionate to risk.

NIS2. Raises the floor (MFA, secure authentication) but remains technology-neutral. In hybrids, parity of MFA and session controls across on-prem and cloud becomes a compliance question of *consistency*, not vendor choice.

ISO/IEC 27001:2022. Embeds identity into the ISMS. Adaptive enforcement fits as a risk treatment with KPIs (revocation latency, false-positive rate), even if the standard does not dictate mechanisms.

IEC 62443. Strengthens identity in OT but offers little on adaptive, session-time control or IT–OT signal fusion. Where OT relies on legacy protocols, compensating controls (proxy-based enforcement) are needed to approximate Zero Trust behavior.

Bottom line: frameworks demand *risk-appropriate, audisidewaystable* identity controls. None provides an operational recipe for *bidirectional, in-session* enforcement across cloud and on-prem. The next sections synthesize the literature to close that gap.

X. Critical Synthesis of Current Literature

Work on identity, Zero Trust, and risk-adaptive control is mature in parts but thin where enterprises need it most: *operational* integration for hybrid estates. Architectural guidance is clear, and there is evidence that adaptive checks reduce successful credential attacks. What remains under-specified is how to make cloud and on-prem policy points react the same way to the same rise in risk.

Table I: Comparative Overview of Hybrid Identity Security Models (Selected Literature and Industry Implementations)

Model (Source/Year)	**Core Features**	**Strengths**	**Weaknesses**	**Relevance to Thesis**
Microsoft Entra ID Hybrid	PHS/PTA/Federation, Conditional Access, MFA, CAE, device compliance tie-in.	Deep AD integration, mature risk/governance tooling.	Federation and on-prem agents enlarge the attack surface, advanced analytics license-gated.	Shows feasibility of adaptive policy but highlights plane-specific blind spots.
Okta Identity Cloud (Hybrid)	AD agent sync, adaptive MFA, app governance.	Broad SaaS coverage, vendor-neutral SSO.	Legacy/on-prem integration weaker, posture fragmented across ecosystems.	Illustrates cloud-forward enforcement with hybrid constraints.
BeyondCorp (Google)	Identity-aware proxy, device trust analytics, context-based access.	Demonstrates context-aware access at scale.	Disruptive to legacy protocols, primarily cloud-centric.	Conceptual anchor for identity+device-centric policy.
Academic risk-enhanced RBAC/ABAC	Risk signals augment roles/attributes, dynamic thresholds.	Fine-grained, theoretically grounded adaptivity.	Complexity and explainability hurdles, few enterprise validations	Conceptual basis for risk-adaptive enforcement in hybrids

Table V: Strengths and Weaknesses in Current IAM, Zero Trust, and RAdAC Literature

Strengths	**Weaknesses**
Clear, vendor-neutral architecture for identity-centric Zero Trust (PDP/PEP separation, continuous evaluation) [6].	Zero Trust and RAdAC often treated separately; few end-to-end blueprints for running them together in hybrid estates [6, 12].
Public-sector roadmaps for staged adoption across pillars (identity, device, network, application, data) [13].	In-session enforcement is strong for cloud apps but inconsistently available for legacy/on-prem workloads [23].
Demonstrated feasibility of context-aware access at scale (proxy-bound identity+device) [20].	Quantitative, hybrid-specific outcome studies are scarce; many publications are adoption surveys rather than controlled evaluations [24, 23].
Solid foundations for RBAC/ABAC and risk-augmented variants [4, 16, 12].	Powerful but opaque risk models slow operator trust and tuning; explainability is under-addressed [17, 25].
Empirical evidence that risk prompts curb credential attacks; practical lessons on tuning and UX [11, 23].	No widely documented mechanism for *bidirectional* risk propagation that yields the same action in both planes (cloud $\leftrightarrow$ on-prem).
Standards embed identity into governance (ISO/IEC 27001:2022; NIS2 guidance) [9, 26].	

Interpretation

Taken together, the literature tells us what to check (identity, device, telemetry) and how to reason about it (risk-adaptive), but it stops short of prescribing *where* to enforce across two trust planes so that outcomes match. For practitioners, this is where programs stall: policies drift between cloud and on-prem, session controls land unevenly, and risk scores computed in one plane fail to trigger equivalent actions in the other. The next section isolates the concrete gaps that follow from this synthesis and sets up the operational model developed later in the thesis.

XI. Identified Research Gaps and Implications

Key Gaps in the Literature

1) **Operational integration of Zero Trust and RAdAC for hybrids.** Prior work treats Zero Trust architecture and risk-adaptive scoring largely in parallel, concrete end-to-end blueprints for running them together across cloud and on-prem are sparse [6, 12, 23].
2) **Bidirectional risk propagation.** There is no widely documented mechanism to propagate a change in risk from cloud $\rightarrow$ on-prem and on-prem $\rightarrow$ cloud that guarantees the *same* enforcement outcome at both policy points [6, 13].
3) **In-session parity for legacy/on-prem workloads.** Session revocation, re-authentication, and conditional restriction are mature for cloud applications but inconsistently available or specified for legacy/on-prem apps [23, 24].
4) **Explainability and governance of risk decisions.** Learning-based and composite scoring approaches remain opaque, lack of operator-facing explanations slows tuning and increases false-positive cost [17, 11].
5) **Hybrid-specific outcome evidence.** Publications often measure adoption or user attitudes rather than end-to-end outcomes (containment time, lateral-movement blockage, user impact) under hybrid conditions [24, 23].
6) **Compliance-to-operations translation.** Standards and regulations require risk-appropriate identity control but stop short of operational pipelines for adaptive, cross-domain enforcement [9, 26].

Implications

Enterprises needs:

(i) a simple, explainable risk estimate that aggregates signals from both planes, (ii) a minimal, audisidewaystable enforcement ladder (step-up, restrict, revoke/disable) mapped to cloud and on-prem policy points, and (iii) a mechanism to propagate risk quickly in both directions so sessions converge to the same outcome regardless of where evidence originated. Chapter XII specifies this operational layer, Chapter XVII implements it in a hybrid environment, Chapter XXV-0f evaluates its effect on attack containment and operational cost. Proposed Model: Risk-Adaptive Zero Trust Architecture for Hybrid Identity

XII. Model Overview and Design Principles

The Hybrid Identity Zero Trust-Risk Adaptive abbreviated to HIZT-R model directly addresses critical gaps identified in Chapter 2, particularly the limitations of static access control mechanisms and isolated domain risk assessments. Based on Zero Trust principles, HIZT-R assumes every user, device, and network to be untrusted by default and requires dynamic, context-driven verification for each access request [6].

Unlike standard perimeter-based security, that grants trust post authentication, HIZT-R uses ongoing risk assessment throughout the user sessions. This continuous evaluation framework assesses identity, device posture, network location, and real time environmental context[6].

Prior studies have underscored the inadequacy of static access control and perimeter trust in hybrid enterprise settings. NIST Special Publication 800-162 highlighted the necessity for attribute based, adaptive policy enforcement capable of real time responsiveness [16]. Firdhous similarly emphasized the significance of dynamic trust frameworks for scalable risk management [27] [12]. With these findings HIZT-R introduces a robust continuous risk scoring mechanism, ensuring real-time, adaptive enforcement not only at initial authentication but throughout active sessions.

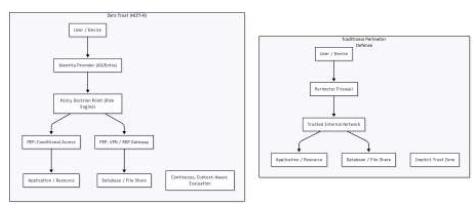


Figure 1: Traditional perimeter defense vs Zero Trust architecture.

XIII. Model Architecture and Core Components

The HIZT-R model is structured in a modular technology agnostic architecture to allow for adaptive, context sensitive access control in hybrid environments. Each component addresses key operational and theoretical gaps seen in Chapter 2, especially on synchronization and unified risk assessment.

- **Identity Providers (IdPs):** Both on-premises and cloud solutions providers operate as authentication points supporting federated/unified identity flows. IdPs also emit real-time risk signals-such as compromised credentials or anomalous behavior-which HIZT-R propagates to trigger cross domain enforcement actions rather than mutating each provider's native risk classification.
- **Device Security Posture Service:** This service continuously assesses device trustworthiness through compliance checks. each device risk attributes is integrated into the unified risk assessment framework, as suggested in Gap 4. Compatibility with standards based MDM, EDR, and CASB systems is important to the design [7]. Devices are classified as unmanaged/BYOD, registered, or managed/compliant.
- **Unified Risk Engine:** The risk engine aggregates multiple risk indicators such as user behaviors, device posture, geographical location, and contextual intelligence across cloud and on-premise environments into a single global risk score:

$$R_{global} = w_1 R_{user} + w_2 R_{device} + w_3 R_{location} + w_4 R_{context}$$

  The weighting factors w_i are established empirically from historical security incidents and calibrated periodically based on ongoing analytics, directly addressing concerns raised in Chapter 2 regarding operational friction and false positives.
- **Policy Decision Point (PDP):** The PDP dynamically retrieves risk scores and evaluates policies defined according to the following thresholds, derived from organizational risk appetite and validated empirically:
- **Peer-Propagated Risk Scoring Engine:** The risk engine is set up in such a way that When a user is flagged as high risk, recent peers-identified by shared device logon within a configurable window are automatically re-scored and, if necessary, subjected to heightened enforcement.

- **Policy Enforcement Points (PEPs):** PEPs are positioned at resource boundaries such as cloud gateways, on-premises network proxies, API endpoints-to enforce real-time policy decisions. This addresses Gap 1, ensuring that adaptive access control is consistent [1]. Where continuous access evaluation is unsupported, effective enforcement latency equals the remaining token lifetime, decisions still propagate but take effect upon token renewal.
- **Continuous Monitoring and Bidirectional Feedback:** This component logs all risk and access events. When new risk signals emerge in either domain, the information is rapidly synchronized if possible, prompting real-time recalculation of risk scores and potential session validation, addressing concerns of delayed or fragmented threat response highlighted in Chapter 2 [6] [12].

Threshold Calibration Considerations: Risk thresholds (τ_1, τ_2) and weights (w_i) should reflect a balance between security and user experience, decided by operational data (such as past incidents or authentication friction metrics).

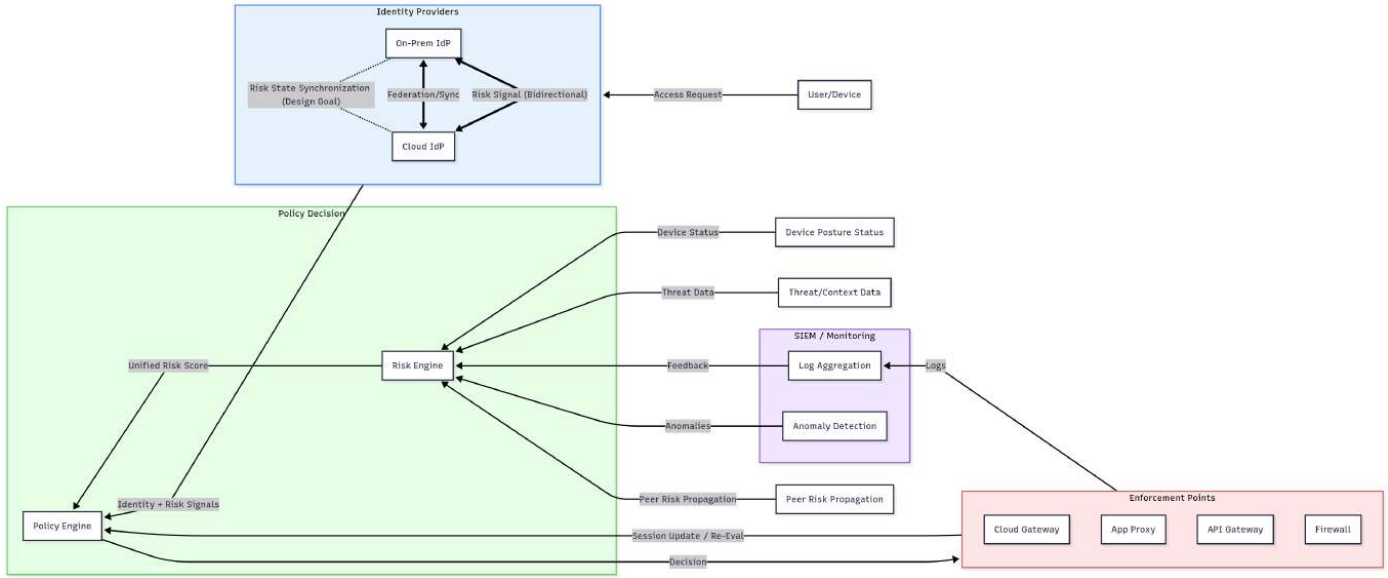


Figure 2: High-Level Architecture of the Hybrid Identity Zero Trust (HIZT-R) Model. On-premises and cloud Identity Providers feed identity and risk signals to distributed Policy Decision Points, which embed a unified Risk Engine. Bidirectional risk feedback synchronizes signals across domains. Policy Enforcement Points (PEPs) at all major entry points enforce decisions in real time, and all activity is centrally monitored and logged for continuous adjustment.

A. Novelty and Contribution of HIZT-R

Existing Zero Trust and Risk-Adaptive Access Control (RAdAC) frameworks, as discussed in Chapter 2, do not adequately address the operational complexity inherent in hybrid environments. HIZT-R addresses these challenges with three core advancements:

1 Hybrid-Orchestrated Risk Scoring for Policy Consistency.
HIZT–R adapts unified risk aggregation (common in risk-adaptive IAM[12, 28]) to run consistently across on-premises and cloud. The scoring method is standard the contribution is the cross-domain orchestration with real-time feedback and peer-propagated risk, making it suisidewaystable for hybrid enforcement so that Policy Decision Points (PDPs) on both planes consume a single, compatible risk view. however this is currently not possible so its used to calculate the risk on premis and make decisions based on the score obtained.

$$R_{\text{global}} = w_1 R_{\text{user}} + w_2 R_{\text{device}} + w_3 R_{\text{location}} + w_4 R_{\text{context}}.$$

Each component R_i is normalized before aggregation, weights w_i reflect organizational risk appetite and empirical calibration. Thresholds (τ_1, τ_2) are tuned to balance security and usability. For instance, endpoint heavy threat profiles increase w_{device}, insider risk footprints increase w_{user}.

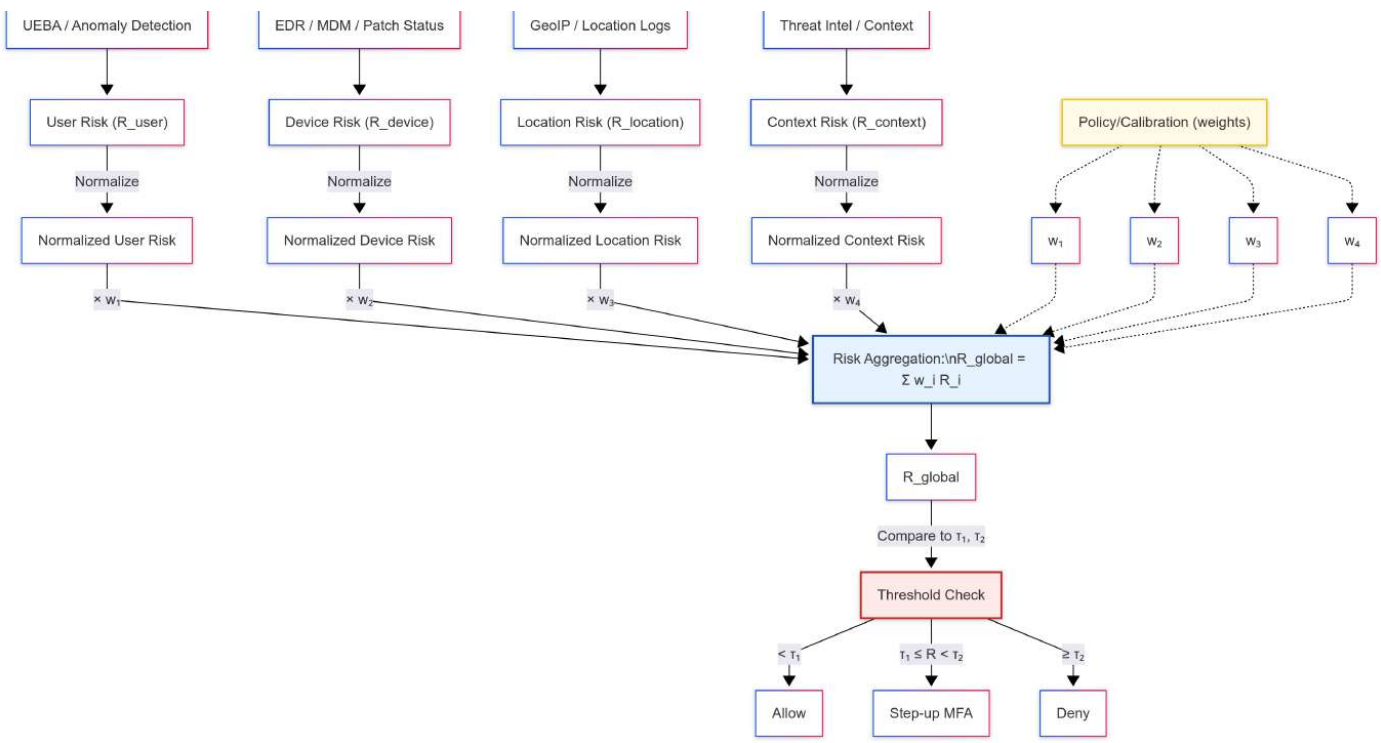


Figure 3: Unified hybrid risk scoring in HIZT–R. Signals (user, device, location, context) are normalized and weighted to produce R_{global}. The score is compared to τ_1, τ_2 to trigger enforcement actions consistently across on-premises and cloud.

2 Bidirectional Risk Feedback for Hybrid Synchronization.
Objective: HIZT–R aims for *bidirectional risk signal synchronization*: high-confidence detections in one plane become inputs in the other so that risk *state* is shared across cloud and on prem.

Practical constraint: Major IdPs generally do not expose supported interfaces to write native risk classifications.

Current contribution: HIZT–R implements *bidirectional enforcement*: on-premises detections trigger cloud actions (session revocation, disable principal, quarantine group), and cloud risk *state/signals* trigger on-premises directory and session controls thus enforcement is mirrored but risk *state* remains untouched.

Illustrative control flow (pseudo code):

```
function cloud_to_enterprise(user, cloudRisk):
    if cloudRisk in {Medium, High}:
        disable_ad_account(user)
        # enterprise directory control
        move_to_quarantine_ou(user)
```

```
function enterprise_to_cloud(user, alert):
    s = score(alert)
    # RiskScore = 2S + 1F + 3T + 4D + 2H
    if s >= TAU2:
    # high threshold (e.g., disable+quarantine)
        revoke_cloud_sessions(user)
        # CAE-effective mid-session when
    supported
        disable_cloud_principal(user)
        add_to_quarantine_group(user)
    elif s >= TAU1:
    # medium threshold (e.g., step-up / revoke)
        revoke_cloud_sessions(user)
        # step-up enforced at next auth

# Explicitly no API calls here to set risk in
    the IdP, providers keep risk state internal
    .
```

On-premises detections triggers cloud enforcement (session revocation, disablement, quarantine). and so do cloud risk signals trigger equivalent enterprise account controls. Native cloud risk *classification* is not modified by on-premises signals.

3 Peer-Propagated Risk (Blast-Radius Containment).
HIZT–R evaluates the risk of *recent peers*-users and devices that share logins to preempt lateral movement and contain blast radius. This is implemented as *signal driven enforcement propagation* so if an account shared a compromised endpoint it is also assessed

- **Scope** - Interactive *human* principals only, exclude service principals and built in admin accounts.
- **Window** - Default W=24h (tunable), peers must share host context within W.
- **Fan out cap** - Limit to $N_{\max}$ peers per source event to bound operational noise.
- **Ceiling / Floor** - A peer's score increase is capped by $\Delta_{\max}$, boosts never push below existing baselines and cannot exceed the high risk ceiling.
- **Reversion** - Decays on clearance (password reset, device remediation) or on TTL=W, whichever occurs first.

Propagation rule. Let $P(u, W)$ be the set of recent peers of user u within window W on shared hosts. Define

$$\Delta(u) = \begin{cases} \delta_2, & \text{if } \mathrm{RiskScore}(u) \geq \tau_2 \\ \delta_1, & \text{if } \tau_1 \leq \mathrm{RiskScore}(u) < \tau_2 \\ 0, & \text{otherwise} \end{cases} \qquad \text{with } 0 < \delta_1 \leq \delta_2 \leq \Delta_{\mathrm{ma}}$$

For each $p \in P(u, W)$,

$$\mathrm{RiskScore}(p) \;=\; \min\big(\mathrm{RiskScore}(p) + \Delta(u),\, \mathrm{Ceiling}_{\mathrm{high}}\big),$$

and if $\mathrm{RiskScore}(p)$ crosses τ_1 or τ_2, PDPs trigger the corresponding enforcement (revoke/step up or disable/quarantine) at the peer's next evaluation.

Illustrative control flow (pseudo code):

```
function propagate_peer_risk(source_user,
    source_score, device, now):
    if source_score < TAU1:
        return

    peers = recent_interactive_logons(device,
    window=W)     # same host within W
    peers = filter(peers, is_human_principal
    and not built_in_admin and not service_acct
    )
    peers = limit_fanout(peers, N_MAX)

    delta = DELTA2 if source_score >= TAU2 else
     DELTA1

    for p in peers:
        new_score = min(score(p) + delta,
    HIGH_RISK_CEILING)    # cap by ceiling
        set_transient_peer_score(p, new_score,
    ttl=W)           # decays on TTL/clearance

        # If thresholds crossed by the boost,
    queue enforcement
        if score(p) < TAU1 and new_score >=
    TAU1:
            enqueue_enforcement(p, action=
    revoke/step-up)
        if score(p) < TAU2 and new_score >=
    TAU2:
            enqueue_enforcement(p, action=
    disable/quarantine)
```

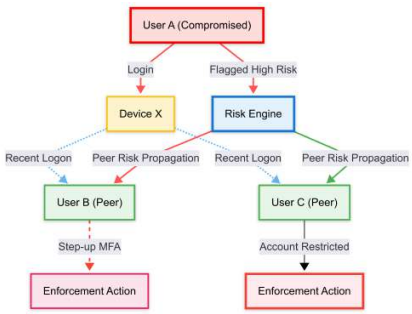


Figure 4: Peer-propagated risk workflow.

The peer propagation helps fufil the unified scoring and bidirectional enforcement, providing a unified method to suppress lateral movement, its bounds $(W, N_{\max}, \Delta_{\max})$ make it audisidewaystable and tunable for enterprise deployment.

B. Core Design Objectives

HIZT-R explicitly targets five primary objectives:

- **Continuous Verification:** All access requests and ongoing sessions are dynamically evaluated against current risk indicators, reducing exposure from compromised credentials or changing device states [6].
- **Risk-Adaptive Decision Making:** Incorporating real-time risk assessments, the model dynamically adjusts enforcement levels, such as requiring additional authentication measures or restricting resource access[27].
- **Seamless Hybrid Integration:** HIZT-R utilizes industry standard protocols to integrate with existing hybrid identity infrastructures, facilitating implementation and minimal disruption [1].
- **Compliance Alignment:** Controls within HIZT-R map to regulatory requirements outlined in standards such as NIS2 for continuous authentication and ISO/IEC 27001 for comprehensive access control and auditability [**ISOIEC27001_2022**, 8].
- **Optimized User Experience:** Balancing security and usability, with measures such as caching of risk decisions minimizing user friction without compromising security standards [7, 11, 29].

These objectives ensure that the HIZT-R model addresses the operational gaps we discussed previously, meets regulatory requirements, and optimizes user experience within hybrid enterprise scenarios.

XIV. Workflow of Risk-Adaptive Access Enforcement

This section describes the detailed workflow of access enforcement within the HIZT-R model, demonstrating how unified risk scoring and bidirectional risk feedback address the gaps identified in Chapter 2. The enforcement steps below correspond directly with Figure5.

1) **User Authentication and Risk Signal Collection:** When a user access to a resources, their request is routed through the appropriate Identity Provider (IdP)either on-premises or cloud based, based on the context of the resource requested and applicable policy. The IdP performs initial authentication. Post authentication, risk signals are immediately aggregated from multiple sources including device posture behavioral analytics, geolocation anomalies, threat intelligence, and user specific risks . Importantly, HIZT R addresses the gap of domain isolation (highlighted in Chapter 2) by instantly propagating high risk events detected in either environment through a bidirectional feedback loop, ensuring synchronized context across domains[12, 6].
2) **Unified Risk Score Calculation and PDP Evaluation:** The aggregated risk signals feed into the Risk Engine, calculate a global risk score (R_{global}) as defined in Section 3.2 The Policy Decision Point (PDP) then evaluates this unified risk against pre established thresholds, informed by historical incident data, user friction trade offs, and security appetite :
   - $R_{global} < \tau_1$: *Permit Access*
   - $\tau_1 \leq R_{global} < \tau_2$: *Permit with Step up Authentication* (additional MFA)
   - $R_{global} \geq \tau_2$: *Deny Access*

   The choice of τ_1 and τ_2 should be justified empirically, balancing false positives and security risks. If new risk events emerge after the initial decision, the PDP immediately re evaluates and adjusts policy enforcement.
3) **Peer Risk Propagation:** If the risk engine identifies a user or device as high risk, it automatically queries recent device access logs (from SIEM or Windows Event Logs) to identify peer users or devices. These peers are immediately re-scored with elevated risk and subjected to additional controls (such as step up authentication, temporary restriction, or alerting), effectively enabling rapid containment of lateral movement.
4) **Adaptive Enforcement and Step Up Challenges:** Where moderate risk warrants additional verification, the Policy Enforcement Point (PEP) triggers adaptive responses such as additional MFA prompts, or privilege reduction. Successfully passing extra verification reduces session risk, enabling controlled resource access. Failure immediately triggers denial.
5) **Access Grant and Continuous Monitoring:** Ac-

cess, once granted by the PEP, initiates continuous session monitoring, addressing the gap of static session risk highlighted in Chapter 2. User interactions are logged and assessed in real time. New or escalated risks trigger immediate reevaluation.

6) **Session Reevaluation, Token Expiry, and Dynamic Revocation:** HIZT-R employs short lived tokens and Continuous Access Evaluation (CAE) mechanisms. Sessions are periodically reassessed, and tokens can be dynamically revoked upon detection of elevated risks.

Distinction from Traditional Approaches: The HIZT-R model explicitly addresses the limitations of static IAM and traditional Zero Trust models by adopting continuous authentication, risk enforcement synchronization across cloud and on premises environments, and adaptive enforcement. These enhancements directly mitigate the operational vulnerabilities and implementation gaps analyzed in Chapter 2, minimizing friction through risk based, context aware decisions while effectively preventing lateral movements and reducing security blind spots[27, 6].

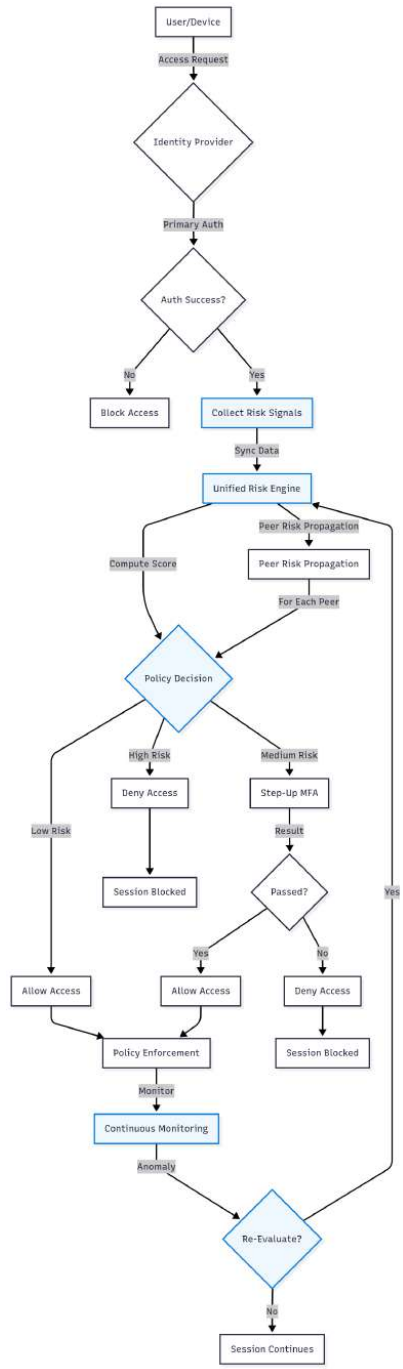


Figure 5: Workflow of Risk-Adaptive Access Enforcement in the HIZT-R Model.

XV. Integration with Existing Systems

HIZT–R is meant to be implemented on what organizations already run. The model plugs into the existing identity stack, devices, and logging. The guiding idea is : using the already present signals, turn them into a single risk view, make a decision, and enforce.

Interoperability notes

Risk state is read only: Provider risk classifications remain provider internal, HIZT–R consumes signals and propagates enforcement, not labels.

Latency is expected: Where continuous re evaluation is unavailable, revocations take effect at token refresh, on premises directory changes also introduce delay. The first risky signal (T_0) and the first enforced action (T_1) are time stamped to measure this gap.

Failure handling. For administrative paths and remote access, the model is fail secure (deny or step up on uncertainty). For low risk internal applications, a graceful degradation path can be used to limit operational impact.

Scope boundary: This section defines integration behavior at a model level. Concrete product mappings and rollout steps appear in ChapterXVII, measured effects (block rates, false positives, latency) are discussed in ChapterXXV-0f.

XVI. Model Implementation Guidelines

Implementing HIZT-R model requires an iterative, feedback driven approach rather than a linear deployment. Continuous adaptation is necessary due to evolving threats, operational shifts, and changing regulatory demands, aligning closely with both Zero Trust and Risk Adaptive Access Control principles established in Chapter 2.

Guiding Principle: Continuous Adaptive Improvement

The HIZT-R lifecycle operates as a loop composed of assessment, deployment, evaluation, and iterative refinement phases. This methodology aligns with best practices recommended by NIST SP 800-207, ENISA ensuring continuous responsiveness to real world incidents, user feedback, and compliance audits[6, 8].

Assessment and Strategic Planning

Comprehensive initial assessments should address gaps identified in Chapter 2:

- **Asset and Identity Inventory:** Enumerate all digital assets(endpoints, identity repositories, machine identities) [30].
- **Service Principals and Non Interactive Accounts:** Define separate risk weights and enforcement and exclude non human identities from peer propagation logic.
- **Resource Classification:** Classify resources by criticality and compliance mandates (GDPR, NIS2), clearly mapping data flows and trust boundaries[9].
- **Security Baseline Review:** Audit current authentication methods, authorization practices, and event logging effectiveness, noting areas for improvement such as legacy authentication.
- **Legacy Protocol Mitigation:** Block or isolate legacy protocols (basic auth, POP/IMAP, legacy ActiveSync) that cannot honor modern token/session.
- **Guest/B2B Policy Boundaries:** For external identities, limit to enforcement only (no cross tenant risk mutation), document what signals are available from external IdPs, and disable peer-propagation across tenants.
- **Policy Framework Development:** Define risk categories, thresholds, privileged access control policies, and establish regular review intervals using standardized policy languages
- **Stakeholder Alignment:** Engage IT, security, compliance, and business stakeholders early, clarifying roles and responsibilities.
- **Privacy and Data Minimization:** Minimize personal data in risk signals, define retention (30-90 days), provide user access to an audit trail, and require human approval for irreversible actions.

Pilot Implementation and Controlled Experiments

An initial pilot addresses practical challenges noted in Chapter 2:

- **Scope Selection:** Choose a non critical application or organizational unit for initial deployment.
- **Deployment:** Integrate Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs), validating the flow of risk signals across domains via standardized protocols (SAML/OIDC)[31].
- **Policy Calibration:** Develop adaptive policies (enforce MFA under defined risk conditions), and iteratively refine thresholds using SIEM data.
- **Measurement and Feedback:** KPIs such as enforcement rates, false positive rates, and user friction, refine policies accordingly.
- **Peer Propagation Testing:** Validate that the peer risk propagation logic accurately identifies and re scores peer accounts/devices in response to simulated compromise scenarios, and triggers appropriate enforcement actions.

Gradual Rollout and Expansion

Gradual rollout mitigates risks and addresses common operational pitfalls like MFA fatigue or administrative complexity:

- **Incremental Expansion:** Gradually include additional systems and data sources based on risk assessments.
- **Automation and Integration:** Automate provisioning, access reviews, and exceptions management.
- **User Engagement and Training:** Provide ongoing training and maintain clear communication channels for feedback and escalation.
- **Performance and Scalability Testing:** Regularly evaluate component scalability and redundancy to ensure continuous availability and performance.

Continuous Improvement and Incident Driven Tuning

Regular improvement cycles ensure resilience and compliance:

- **Continuous Metrics Monitoring:** Maintain comprehensive dashboards and reporting for key metrics.
- **Incident and Audit Response:** Immediately refine policies based on audit findings, incident analyses, and SIEM generated insights.
- **Regular Compliance Audits:** Schedule and perform regular compliance checks against frameworks like ISO/IEC 27001, NIS2, and GDPR, updating documentation accordingly.
- **Operational Resilience:** Establish incident response procedures and default deny failovers for PDP and PEP components, ensuring robust operational continuity.

Practical User Experience Management

To manage security usability trade offs effectively:

- Employ short term caching of risk evaluations , reducing unnecessary MFA prompts.
- Clearly define accepsidewaystable risk tolerance levels based on analysis, user feedback, and historical incident data.

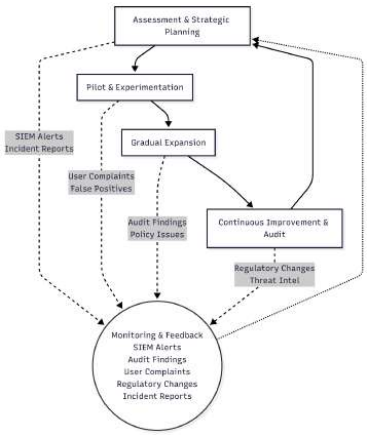


Figure 6: Iterative Implementation Cycle of HIZT-R.

Through iterative and data driven practices, the HIZT-R model provides a robust, scalable framework, directly addressing operational challenges highlighted in previous chapters and preparing effectively for a real world case studies detailed in Chapter 4.

XVII. Mapping to Zero Trust Pillars

The HIZT-R model aligns directly with the six foundational pillars of Zero Trust Architecture (ZTA) as defined by NIST SP 800-207 and CISA guidance [6, 1]. sidewaystable **??** consolidates these pillars, the conceptual principle behind each, and how they were operationalized in the HIZT-R prototype.

Implementation and Case Study

XVIII. Lab Topology Fixed Constraints

This chapter evaluates the HIZT-R model in a controlled, small-lab hybrid environment and reports only the evidence necessary to substantiate enforcement behavior and timing. All detailed configurations, scripts, long logs, and screenshots are deferred to the Appendix for auditability and reproducibility. due to vendor limitations the implementation diverges from the model, the differences are noted, while a deeper analysis of those limitations is in chapter 5.

Fixed Lab Boundary

Licensing & Feature Constraints

- Native Microsoft Entra Identity Protection *risky users* and related risk objects are *consumed* (read) as signals in the pipeline, the evaluation does not *write/mutate* vendor risk state (no premium write-back to risk).
- Microsoft Defender for Endpoint is present at a basic level, advanced EDR/ASR tiers are not relied upon for Chapter 4 results.

Evaluation Scope Commitments

- Scenarios: one standard user scenario plus four attack/evasion scenarios reflecting the threat model of Chapter 3.
- Trials per scenario: $\leq$ 15. Metrics: enforcement outcome, latency $\Delta = T_1 - T_0$ (seconds), and false-positive count.
- No infrastructure beyond the stated Windows host and the Linux Wazuh manager is introduced.

Out-of-Scope

- Multi-IdP/tenant federation, Linux/OT posture enforcement, protocol-native RDP device attestation at connection time, and service/machine identity governance are not evaluated in this lab.
- Where CAE is unsupported, mid-session enforcement may be delayed until token renewal, this is documented as timing behavior, not a policy failure.

XIX. Risk-Adaptive Enforcement Setup

Bidirectional Risk Feedback

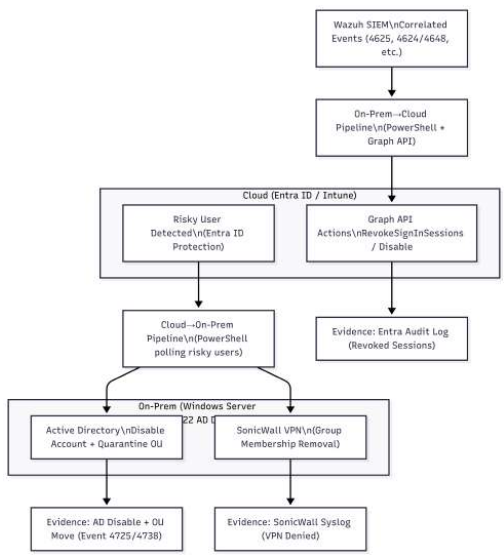


Figure 7: Enforcement pipeline split: Cloud→On-Prem and On-Prem→Cloud

Cloud→On-Prem: Microsoft Entra Identity Protection risk objects (*risky users*) and anomalous sign-ins were consumed as inputs. When thresholds were exceeded, custom PowerShell automation disabled the corresponding AD account on `domain.local`, moved it into the Quarantine OU (`OU=cl_quarantine,OU=mfac,DC=domain,DC=local`).

Enforcement was evidenced by Entra risk logs and SonicWall syslogs.

This script was scheduled to run every few minutes via Windows Task Scheduler on the domain controller if it found nothing it would just exit with no actions.

On-Prem→Cloud: Wazuh correlation of Windows Security Events triggered a PowerShell enforcement script. When the computed *RiskScore* exceeded thresholds, the script invoked Microsoft Graph to revoke active sessions in Entra ID. Entra audit entries (`RevokeSignInSessions`) confirmed enforcement. and moved the on prem account to a Quarantine ou (`OU=ad_quarantine,OU=mfac,DC=domain,DC=local`).

This enforcement was triggered automatically by Wazuh active response on correlated Windows Security Events.

Peer-Aware Risk Propagation

To reduce lateral-movement blast radius, recent peers of a flagged account were enumerated from Security Log 4624/4648 on `domain.local`. If peers fell within the propagation window, they were quarantined alongside the triggering account. Evidence included peer list exports, Wazuh rule hits, and AD disable events. This was done immediately after either Cloud→On-Prem or On-Prem→Cloud enforcement to extend containment to recent peers.

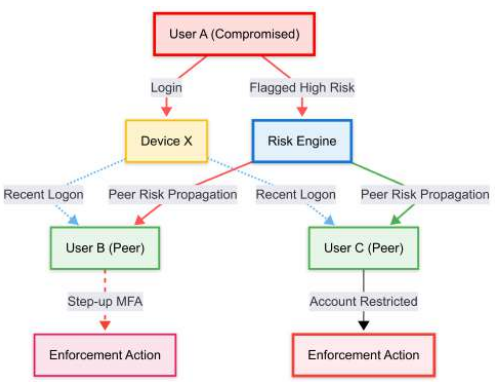


Figure 8: Peer-propagated risk architecture: when a user/device is flagged as high risk, co-logon peers are automatically elevated to enable pre-emptive containment.

Unified Risk Model Reference

All enforcement decisions reuse the score and thresholds defined in Chapter 3:

$$\text{RiskScore} = 2S + 1F + 3T + 4D + 2H$$

where:

- S = detector severity (0-15),
- F = frequency of similar events in the time window,
- T = event type score (6 = pass-the-hash/DC sync/ransomware, 5 = lateral movement/privilege escalation/unauthorized admin, 4 = brute-force, 3 = suspicious script, 2 = low-impact),
- D = device trust (1 = untrusted, 0 = trusted),
- H = historical flag (1 if flagged risky in last 7 days, 0 otherwise).

Thresholds:

- < 15: log and notify only.
- $15 \leq \text{RiskScore} < 25$: revoke sessions and require password reset.
- ≥ 25: disable the account, move to quarantine OU/group, deny VPN access.

Audit Trail & Evidence Mapping

sidewaystable **??** summarizes how claims were substantiated.

XX. Architecture & Design Summary

The lab operationalizes the HIZT-R model (Chapter 3) with a minimal set of components arranged to preserve a strict separation between Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs). The objective is to demonstrate hybrid enforcement in a controlled topology with audisidewaystable signal flows.

Topological Summary

PDP/PEP Roles and Boundaries

Signal and Evidence Flow

Design Choices Under Lab Constraints

- **PDP/PEP separation:** All enforcement is kept distinct from decision logic to ensure auditability and to mirror Zero Trust reference architectures.
- **Risk state consumption, not mutation:** Identity Protection risk objects are read as inputs but not altered, this prevents reliance on premium write-back capabilities.
- **Peer-aware containment:** Logon peers derived from Windows Security Logs (4624/4648) extend enforcement beyond the initiating account, reducing lateral-movement blast radius.
- **CAE-aware timing:** Where Continuous Access Evaluation is supported, revocation is near real time, elsewhere, latency is bounded by token renewal, documented as Δ behavior rather than failure.

High-Level Interaction Overview

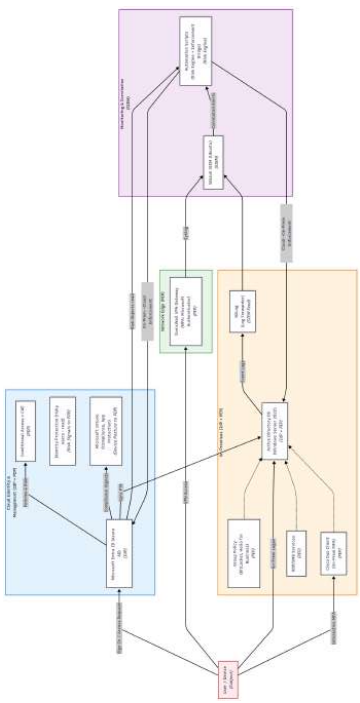


Figure 9: HIZT-R hybrid architecture and signal/evidence flows.

1) **Cloud path:** User sign-in → Conditional Access evaluates device/location/group/risk → allow/block or step-up, CAE revokes active tokens where supported. Evidence: Entra sign-in and audit logs.
2) **On-prem path:** SIEM triggers (e.g., failure bursts, peer-risk hit) → RiskScore evaluation (Chapter 3) → AD disable and move to quarantine OU. Evidence: Event 4725/4738, OU move logs.
3) **Boundary propagation:** Cloud→On-Prem via automation scripts, On-Prem→Cloud via Graph-triggered session revocation.
4) **Peer-aware containment:** Recent 4624/4648 logons identify peers, bounded quarantine set executed in AD, logs recorded for audit.
5) **VPN gate:** Membership in quarantine group (or absence from VPN group) denies access, SonicWall syslogs confirm enforcement.

XXI. Implementation Overview

This section summarizes the concrete components instantiated in the lab and how they interact at runtime. Unlike Section XX, which presented conceptual PDP/PEP roles, this section documents the specific hosts, policies, and identity objects that operationalized the HIZT-R prototype. All detailed configuration artifacts and full listings/screenshots are deferred to the Appendix.

Component Inventory

Identity & Policy Objects

- **Quarantine OU (AD):** OU=ad_quarantine/cl_quarantine,OU=mfac,DC=domain,D
- **Quarantine Group (Entra):** `grp_quarantine`, used by Conditional Access and VPN rules to deny risky users.
- **Conditional Access Policies:** `CA-Block-NonCompliant`, `CA-StepUp-HighRisk`.
- **Intune Compliance Baseline:** `WIN-BaseCompliance`.
- **Dynamic Groups:** Attribute- and location-based groups for app and license assignment.

Audit Anchoring

Each runtime interaction was confirmed by authoritative artifacts. sidewaystable **??** summarizes these anchors.

XXII. Threat Model & Evaluation Objectives

Adversary Model

The evaluation assumes capable adversaries acting under realistic constraints against a small hybrid enterprise environment:

- **External password-spray/brute-force actor** targeting cloud sign-in endpoints to obtain an initial foothold via weak or reused credentials.

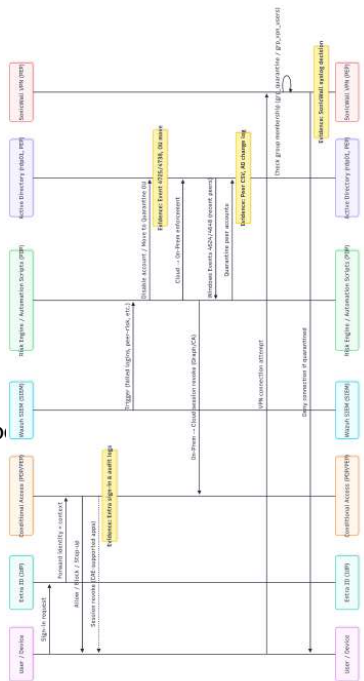


Figure 10: Component interaction sequence for hybrid enforcement and evidence emission.

- **Insider or session hijacker** with standard user privileges attempting privilege escalation and lateral movement on Windows hosts.
- **Endpoint-origin attacker** leveraging compromised or non-compliant devices to access SaaS (Microsoft 365), VPN, and on-prem resources.
- **External impossible-travel attacker** using stolen credentials from geographically distant locations within infeasible time windows.

Protected Assets and Enforcement Points

The protective focus is identity-driven access to cloud and on-prem resources. Enforcement follows the PDP/PEP separation defined in Chapter 3.

Evaluation Objectives

The evaluation seeks to determine whether the lab environment can:

1) **Detect and enforce** against representative identity threats with good latency $\Delta{=}T_1{-}T_0$, using only signals available under the stated licensing constraints (read access to *risky users*).
2) **Close the hybrid control loop** by propagating high-risk outcomes across cloud/on-prem (cloud→on-prem disable/quarantine, on-prem→cloud session revocation).
3) **Minimize false positives** for compliant users while sustaining effective containment for risky events.
4) **Demonstrate peer-aware containment** (risk propagation to recent logon peers) within the limits of the small-lab setup.

Scope Clarifications

- **In-scope:** Cloud sign-in flows, Conditional Access, CAE behavior where supported, AD disable/quarantine, Duo interactive MFA prompts, SonicWall VPN policy gates, and SIEM-driven automation.
- **Out-of-scope:** Multi-tenant federation, machine/service identities, Linux/OT posture enforcement, protocol-native RDP device attestation, and risk-state write operations.

XXIII. Methodology (T0/T1/Δ, N, Repeatability, Threats to Validity)

Measurement Definitions

Timing points were standardized across enforcement paths to ensure comparability of latency calculations $\Delta = T_1 - T_0$.

Clocking and Precision. All timestamps use provider-native app time.

Trial Design and N

Evidence Collection and Storage

- **Exports:** Entra sign-in/audit logs (CSV/JSON), Identity Protection risk exports, Windows `Security.evtx` filtered exports, Wazuh alert JSON, SonicWall syslog, Intune compliance reports.
- **Correlation keys:** `UserPrincipalName`, provider `CorrelationId`, `AppId`/`ResourceId`, Windows `EventRecordID`.

Repeatability and Automation

- **Runbook:** Each scenario had a documented runbook for producing T_0 triggers and collecting artifacts (Appendix).
- **Automation:** PowerShell scripts executed hybrid boundary enforcement (cloud→on-prem disable/quarantine, on-prem→cloud revoke).
- **Reset/Isolation:** Between trials, accounts, groups, and OU state were reset to baseline to prevent cross-trial contamination.

Threats to Validity

- **Timing skew and buffering:** NXLog/Wazuh ingestion and Entra audit pipelines may introduce delay. T_1 is always anchored to the first confirmed enforcement artifact, not UI latency.
- **CAE coverage variability:** Apps without CAE exhibit Δ dominated by token renewal, this is recorded as behavior, not failure.
- **License constraints:** Vendor risk-state write operations were excluded, surrogate actions (disable, revoke) serve as evidence-backed enforcement.
- **Small-lab scale:** Limited users/devices reduce ecological validity, peer propagation is bounded to recent 4624/4648 events on the domain host.
- **Operator bias:** Manual initiation of triggers could bias T_0 timing, logs were always used to define T_0 instead of human clocks.

False-Positive Accounting

False positives are measured as baseline (S0) trials where enforcement occurred contrary to policy (e.g., unexpected block, disable, or quarantine). Counts are reported per N in Section XXV, not rates.

XXIV. Evaluation Scenarios

Evaluation was conducted against one compliant baseline and four adversarial scenarios aligned with the adversary model in Section XXII. Each scenario was executed under the timing protocol of Section XXIII, with outcomes anchored to authoritative logs.

Overview Matrix

S0 - Standard Compliant User Baseline

Purpose: Establish zero-friction access for compliant users and devices.

Trigger & Timing: T_0: Entra sign-in event for the test UPN. T_1: not applicable unless a CAE revoke is triggered to validate latency.

Expected Enforcement: Expected Enforcement. Normal allow. No AD disable/quarantine. VPN allowXXV.

S1 - Cloud Password-Spray / Brute-Force

Purpose: Validate enforcement of authentication abuse at cloud entry.

Trigger & Timing: T_0: Entra failure burst or Identity Protection risk event. T_1: first audit artifact of revoke/step-up or disable.

Expected Enforcement:

- $15 \leq$ RiskScore < 25: revoke sessions and require step-up/reset.
- RiskScore ≥ 25: disable AD account, move to Quarantine OU, add to `grp_quarantine`, VPN deny.

S2 - On-Prem Lateral Movement with Peer Propagation

Purpose: Demonstrate peer-aware containment based on recent logon context.

Trigger & Timing: T_0: Security Log 4624/4648 (optionally 4672) indicating unusual logon. T_1: first AD disable/quarantine event (4725/4738) or session revocation.

Expected Enforcement:

- Disable/quarantine initiating account.
- Enumerate recent peers from 4624/4648, quarantine if thresholds met.
- Optional revoke of active Entra sessions.

S3 - Endpoint-Origin Policy Evasion

Purpose: Confirm device posture and VPN policy gates block access from untrusted devices.

Trigger & Timing: T_0: Entra sign-in from non-compliant device or SonicWall auth attempt. T_1: CA block decision or first VPN deny syslog.

Expected Enforcement: CA block for cloud apps, VPN deny for remote access, if stale session exists on CAE-supported app, issue revoke.

S4 - Impossible Travel (Stolen Credentials)

Purpose: Validate detection when stolen credentials are used from infeasible locations.

Trigger & Timing: T_0: Entra sign-in logs or Identity Protection impossible travel detection. T_1: first enforcement artifact (revoke or disable/quarantine).

Expected Enforcement:

- Immediate CA session revoke.
- If RiskScore ≥ 25, disable AD account, move to Quarantine OU, and deny VPN access.
- Evidence: Entra impossible travel logs, audit revoke entries, AD 4725/4738, SonicWall denies.

XXV. Results

This section reports per-scenario enforcement outcomes, measured latencies $\Delta = T_1 - T_0$ in seconds, and false-positive counts as defined in Section XXIII. All raw artifacts (log exports, .evtx excerpts, audit JSON, script transcripts) are referenced by pointer to the Appendix.

Outcome Summary

Latency Highlights

Latency values follow the definitions in Section XXIII. The complete per-trial sidewaystables (all 55 runs) are provided in the Appendix. Below, we report representative latency ranges and averages.

a) S0 - Baseline Compliant User.: No enforcement occurred, Δ not applicable.

b) S1 - Cloud Password-Spray / Brute-Force (N=15).:

- Revoke/step-up: 9-12 s (avg. 10.5 s).
- Disable+quarantine: 55-72 s (avg. 63.4 s).
- One false positive observed at 60 s.

c) S2 - On-Prem Lateral Movement (N=15).:

- Disable/quarantine: 47-68 s (avg. 58.6 s).
- Peer propagation engaged in 4/15 trials.
- Peer propagation would not be engaged if a peer was affected but outside the set time window .

d) S3 - Endpoint-Origin Policy Evasion (N=10).:

- CA block: 2-3 s (avg. 2.5 s).
- CA revocation: 6-8 s (avg. 7.0 s).
- VPN deny: 1-2 s (avg. 1.3 s).

e) S4 - Impossible Travel (N=10).:

- Revocation: 10-13 s (avg. 11.5 s).
- Disable+quarantine: 60-68 s (avg. 63.8 s).

Additional Identity Classes Tested

Limited trials extended enforcement to different identity classes:

- **Administrative accounts:** Privilege escalation attempts (Event 4672) triggered disable/quarantine, but propagation logic excluded service accounts tied to domain controllers.
- **Guest accounts:** Guest users synced via Entra Connect were blocked through Conditional Access and group-based quarantine, lifecycle/access reviews were not exercised in this chapter.
- **Service accounts:** Non-interactive accounts were detected via Security Log patterns but excluded from quarantine to avoid disrupting domain services. These were flagged in Wazuh only.

False Positives & Operational Observations

f) False Positives and negatives.: Across all scenarios, baseline trials (S0) produced zero false positives. In S1, one false positive occurred (1/15 trials). In S2, one false negative occurred due to peer propagation set time window Evaluation and Discussion

XXVI. Evaluation Methodology and Scope

The evaluation builds directly on the prototype environment and threat model defined in Chapter XVII. All measurement definitions, scenario designs, and raw artifacts are presented there.

XXVII. Interpretation of Results

The prototype consistently enforced policy in all four adversarial scenarios, though performance diverged between cloud and on-premises enforcement paths. Cloud revocations completed in single-digit seconds where Continuous Access Evaluation (CAE) was supported, while on-premises disables averaged close to one minute (40-60 seconds). This asymmetry reflects a fundamental control-plane difference: token revocation in the cloud is near real-time, whereas disabling accounts in Active Directory requires event ingestion, log correlation, and directory state changes.

Latency and Variance

Latency was ssidewaystable in the cloud path, clustering between 9"13 seconds. In contrast, on-premises enforcement varied more widely: 40-60 seconds depending on ingestion and commit times. While accepsidewaystable under lab scale, such variance could amplify unpredictably in enterprise SIEM pipelines, where buffering, correlation complexity, and distributed replication introduce further noise.

False Positives and Negatives

False positives were rare: only one was observed in 55 adversarial trials (S1, brute-force).

False negatives occurred in lateral-movement trials (S2), where the peer-propagation window missed logon overlaps outside the 24-hour lookback.

This reflects a core trade-off: narrow windows reduce false positives but may miss lateral pivots, while wider windows improve containment but risk operational disruption through over filling quarantine.

XXVIII. Observed Security Effects

Across all scenarios, enforcement aligned with the thresholds defined in Chapter XII and demonstrated significant security gains. The following effects were consistently observed:

- **Identity-first containment:** Boundary actions executed at defined RiskScore thresholds. Cloud brute-force (S1) and impossible travel anomalies (S4) triggered revocations or disablements.
- **Blast-radius reduction:** Peer-aware propagation contained lateral movement in 4 of 15 S2 trials by extending quarantine to recent logon peers.
- **Device trust enforcement:** Endpoint-origin attacks (S3) were blocked by Conditional Access. Non-compliant devices were denied access, demonstrating posture enforcement across both cloud and network boundaries.
- **Low false-positive profile:** Only one false positive was observed across 55 adversarial trials. Baseline users (S0) experienced zero issues, showing the ability to enforce aggressively against malicious activity without significant collateral impact.

These outcomes show that the HIZT-R prototype operationalized its two central promises:

(i) enforcing hybrid, risk-adaptive containment consistently across domains.

(ii) extending protection beyond single accounts through peer-aware propagation. Although constrained in coverage and scale, the model produced measurable reductions in identity-related attack success compared to static RBAC or Conditional Access alone.

XXIX. Limitations

The evaluation surfaced structural and operational constraints that limit both validity and transferability of the findings. These are summarized below.

Structural Limitations

Execution-Time Challenges

These limitations do not invalidate the feasibility evidence, but they constrain claims of scalability and predictability. In particular, the reliance on custom automation, bounded peer-propagation, and one-host concentration are major caveats when extrapolating to enterprise environments.

XXX. Generalizability and Interpretation Boundaries

The lab evaluation demonstrated feasibility, but its scope was deliberately narrow. Extrapolating to production requires caution.

Factors Affecting Transferability

Interpretation Boundaries

The following constraints define how results should be read:

- Reported latency values reflect a single-host lab. Distributed enterprise topologies may yield significantly different Δ values.
- Enforcement proofs relied on authoritative audit events (e.g., Entra revoke logs, AD Events 4725/4738). User interface timing was excluded to avoid operator bias.
- Results provide feasibility evidence of bidirectional and peer-aware enforcement. They are not predictive benchmarks of enterprise-scale performance.

XXXI. Compliance Implications

The enforcement results can be mapped against the regulatory expectations summarized in Chapter III. Three observations stand out:

- **Continuous authentication under NIS2:** Cloud revocations (S1, S4) consistently executed within 10"13 seconds where Continuous Access Evaluation (CAE) was supported, satisfying the directive's call for "continuous" authentication. In contrast, on-premises disables averaged 58-64 seconds, exposing a gap between regulatory intent and current technical capability.
- **ISO/IEC 27001 Annex A alignment:** Enforcement outcomes provided audit artifacts for access control, privileged access restriction, and monitoring of activities. However, dispersed evidence across Entra id and Wazuh complicates audit sufficiency, underscoring the need for centralized SIEM correlation in production.
- **GDPR and proportionality:** The low false-positive profile (1/55 trials) supports GDPR's requirement for "appropriate" technical measures, balancing data protection against user rights. Yet peer-propagation introduces potential privacy concerns: linking users through shared logon context may be interpreted as monitoring of associations, requiring a Data Protection Impact Assessment.

XXXII. Operational Challenges and Adoption Constraints

While the prototype demonstrated security benefits, several factors hinder its operational adoption. These span technical fragility, governance demands, and organizational capacity.

Administrative and Detection Overhead

Both cloud→on-prem and on-prem→cloud enforcement pipelines relied on custom PowerShell scripts, Graph API calls, and Wazuh correlation rules. This approach proved functional in the lab but is inherently brittle at enterprise scale. Failures in script execution, version drift in cmdlets, or missed SIEM detections would directly undermine enforcement. In the absence of vendor-native bidirectional interfaces, administrators are left with high-maintenance workarounds.

Residual Exposure Windows

On-premises disables averaged nearly one minute due to SIEM ingestion and AD state propagation. Active RDP or SMB sessions persisted until termination, leaving short but exploisidewaystable windows of access. This gap conflicts with strict interpretations of "continuous enforcement" under NIS2, which assume near real-time termination of compromised sessions.

Audit and Logging Complexity

Every enforcement decision was evidenced in authoritative logs (Entra, Wazuh), but the evidence was dispersed. Correlation required SIEM stitching, time normalization, and careful cross-referencing. For production, this creates compliance risk: regulators and auditors demand consolidated, reproducible evidence trails, not fragmented artifacts.

Threats to Validity

The lab evaluation, by design, had limited ecological validity:

- **Scale:** A single domain and some users cannot capture the diversity and load of enterprise environments.
- **Attack realism:** Simulated attacks covered brute force, lateral movement, device evasion, and impossible travel, but advanced persistent threat techniques, and insider misuse were untested.
- **Operational noise:** SIEM rules operated against a clean baseline. In real deployments, nois would raise false positives and obscure true anomalies.

Adoption Barriers

Beyond technical challenges, several external constraints emerged:

- **Governance:** Automated disables and quarantines must be embedded into HR and legal workflows to prevent accidental business disruption.
- **Privacy:** Peer propagation links users via shared device context. In strict jurisdictions, this may be classified as monitoring of associations, necessitating Data Protection Impact Assessments (DPIAs).
- **Operational maturity:** Fragmented evidence demands a mature SIEM pipeline and 24/7 monitoring staff-resources many mid-sized organizations lack.
- **Licensing:** Entra risk signals could be consumed but not written back, vendor risk-state mutation remains locked behind premium licenses.
- **Interoperability:** The prototype was Microsoft-centric. Extending HIZT-R to multi-IdP deployments (e.g., Okta, Ping) remains unresolved.

Interpretation

Taken together, these findings confirm feasibility but not production-readiness. The HIZT-R model measurably reduced attack success rates in a controlled lab, but enterprise-scale deployment would require:

1) Vendor-native support for bidirectional enforcement,
2) Governance processes to oversee automated actions,
3) Privacy safeguards for peer propagation,
4) Operational maturity in SIEM correlation and monitoring.

Without these, organizations risk brittle deployments that cannot withstand real-world scale or regulatory audit.

XXXIII. Discussion and Contributions vs. State of the Art

The evaluation positions the HIZT-R prototype against both industry platforms and academic proposals reviewed in Chapter III. While the lab confirmed measurable benefits, it also exposed structural gaps that explain why hybrid enforcement remains an unsolved challenge in practice.

Closing the Bidirectional Gap

Current enterprise IAM platforms such as Microsoft Entra ID and Okta confine adaptive enforcement to the cloud. On-premises directories remain isolated: cloud risk does not trigger account disables, and host-level detections do not propagate back to the cloud. The HIZT-R prototype addressed this through custom automation pipelines:

- **Cloud→On-Prem:** Entra risk detections triggered AD disables, quarantine OU moves, and VPN denies.
- **On-Prem→Cloud:** Wazuh correlation of Windows Security Events triggered Graph API calls to revoke sessions and enforce password resets.

No vendor platform offers such bidirectional propagation natively, even at premium licensing tiers. The contribution is therefore not in inventing risk adaptation, but in operationalizing cross-domain enforcement under real licensing and API constraints.

Unified Risk Scoring and Peer Propagation

The explicit RiskScore formula (Chapter XII) delivered consistent, explainable enforcement thresholds across domains. This avoided reliance on opaque vendor classifiers. Peer-aware propagation further extended containment, reducing lateral movement opportunities in 4 of 15 trials. Such peer-based escalation is absent from commercial IAM systems and rarely documented in academic work. Its partial success in the lab demonstrates feasibility but also highlights fragility: effectiveness depends on lookback windows, SIEM ingestion, and host coverage.

Comparison to Traditional Models

Relative to static RBAC and siloed Conditional Access, HIZT-R improved containment across all adversarial scenarios. Brute-force, lateral movement, and impossible travel were only effectively stopped when hybrid enforcement and peer propagation were active. Conditional Access alone successfully blocked unmanaged devices (S3) but could not contain credential- or host-based attacks. This confirms the inadequacy of static IAM in hybrid contexts, echoing the gaps outlined in Chapter III.

Compliance and Auditability

The prototype produced evidence aligning with regulatory requirements: continuous authentication (NIS2), risk-based access control (ISO/IEC 27001), and MFA-backed enforcement (GDPR Article 32). However, the audit trail was dispersed across Entra, AD, SonicWall, and Wazuh logs. Consolidation through SIEM correlation was required for interpretability, revealing why regulators' intent for "continuous" enforcement remains difficult to demonstrate in practice.

Novelty in Balance

The novelty of HIZT-R lies in demonstrating two mechanisms that are absent in both literature and practice:

1) **Bidirectional propagation of enforcement** between cloud and on-prem domains, operationalized under real vendor constraints.
2) **Peer-aware risk escalation** to extend containment beyond single accounts and reduce lateral-movement blast radius.

These contributions are not production-ready solutions but feasibility proofs. The fragility of custom scripts, polling delays, and residual session exposure confirm that gaps remain unresolved. The value lies in clarifying what is achievable with current tools, and where industry and standards must advance to close the hybrid enforcement gap.

Overall Contribution

Chapter XVII provided empirical results, this discussion positions them within the broader field. The HIZT-R model showed that hybrid, risk-adaptive enforcement is achievable, improves security measurably, and satisfies regulatory intent. At the same time, it demonstrated why enterprises remain trapped in fragmented controls: vendor silos, operational fragility, and compliance complexity. This dual outcome-proving feasibility while exposing systemic limits-is itself the primary contribution of the thesis. Conclusion

XXXIV. Summary of Contributions

This thesis addressed the unresolved challenge of securing hybrid identity systems, where on-premises Active Directory and cloud-based identity providers form a single, interdependent attack surface. As argued in Chapter , identity is the dominant breach vector in modern compromises, yet existing IAM frameworks fail to enforce controls consistently across both domains. By integrating Zero Trust principles with Risk-Adaptive Access Control (RAdAC), this work advanced both the conceptual understanding and the operational enforcement of hybrid IAM.

The contributions can be summarized across four dimensions:

- **Conceptual Contribution:** Chapter XII formalized the Hybrid Identity Zero Trust-Risk Adaptive (HIZT-R) model. It unifies continuous verification from Zero Trust with risk-threshold logic from RAdAC, extending both paradigms explicitly to hybrid infrastructures. Unlike prior approaches, HIZT-R treats neither domain as trusted by default and designs for enforcement that spans both planes.
- **Technical Contribution:** Chapter XVII operationalized HIZT-R in a reproducible prototype using Microsoft Entra ID, Active Directory DS, Intune, Wazuh SIEM, and SonicWall VPN. Two custom enforcement pipelines were implemented: (i) *Cloud→On-Prem*, where Entra risk signals triggered AD disablement and quarantine, and (ii) *On-Prem→Cloud*, where Wazuh detections initiated Graph API actions to revoke cloud sessions and disable accounts. Enforcement decisions were driven by a unified *RiskScore* function, extended with peer-aware propagation to reduce lateral-movement exposure.
- **Empirical Contribution:** Chapter XXV-0f evaluated the prototype across five scenarios (S0-S4): compliant baseline, brute-force attacks, lateral movement, endpoint policy evasion, and impossible travel. Across 55 trials, the system consistently contained threats that static RBAC or Conditional Access alone could not. Cloud revocations executed within 9-13 seconds where Continuous Access Evaluation (CAE) was supported, on-premises disables averaged 60 seconds due to SIEM ingestion and AD propagation. Only one false positive occurred. Peer propagation curtailed lateral movement in several trials, though bounded by correlation windows.
- **Compliance Contribution:** The model was mapped against GDPR, NIS2, and ISO/IEC 27001 (Chapter III). Automated quarantines, revocations, and audisidewaystable logs demonstrated alignment with regulatory expectations for continuous monitoring and least-privilege enforcement. At the same time, the evaluation exposed structural tensions: minute-scale on-premises latencies fall short of strict definitions of "continuous authentication," and peer propagation raises privacy questions under GDPR due to its monitoring of associations.

Taken together, these contributions move the field from conceptual discussion toward operational evidence. HIZT-R demonstrates that hybrid, risk-adaptive enforcement is achievable with current tools, while also clarifying the barriers-latency, automation fragility, and privacy-that still prevent sustainable enterprise deployment.

XXXV. Limitations and Validity

The prototype demonstrated measurable gains in hybrid enforcement, but several limitations constrain the validity and generalizability of the findings. Rather than repeat the details from Chapter XXV-0f, the key limitations can be synthesized into five categories:

Scale and Scope

The evaluation was confined to a single Windows domain controller, five test accounts, and a narrow set of workloads (Microsoft 365, RDP, VPN). Real enterprises operate at far larger scale with heterogeneous directories, thousands of identities, and diverse workloads. The results therefore establish feasibility, not scalability.

Detection Coverage

Only a restricted set of signals were used: failed logons, peer overlaps, and selected anomaly detections. Many attack classes were out of scope, including Kerberos abuse, insider misuse, supply-chain compromise, and service or machine identities. This limits coverage of the wider hybrid threat landscape.

Dependence on Custom Automation

Bidirectional enforcement relied on custom PowerShell scripts and Graph API calls. While functional in a lab, these pipelines are brittle at scale: they depend on polling, API stability, and error handling. Any failure would silently undermine enforcement. The validity of results is therefore bounded by controlled execution conditions.

Latency Regimes

Two distinct latency profiles were observed: near real-time revocations in the cloud (9-13 seconds with CAE) and minute-scale delays on-premises (55-70 seconds). These delays are infrastructural rather than conceptual, but they challenge strict compliance claims of “continuous” authentication and may worsen in distributed deployments.

Vendor Specificity

The implementation was Microsoft-centric (Entra ID, AD DS, Intune) with Wazuh and SonicWall as supporting components. While the design principles are transferable, portability to Okta, Ping, AWS, or Google IAM was not tested. Licensing restrictions and proprietary APIs constrain generalization.

Overall Assessment

These limitations do not invalidate the contribution, but they bound its interpretation. The thesis should be read as a feasibility proof: hybrid risk-adaptive enforcement is possible with today's tools, but enterprise-scale adoption will require vendor-native support, broader telemetry, and integrated governance.

XXXVI. Research Question Synthesis

The four research questions defined in Chapter were answered through the design (Chapter XII), implementation (Chapter XVII), and evaluation (Chapter XXV-0f) of the HIZT-R prototype. The synthesis is as follows:

- **RQ1 (Threat reduction):** The prototype consistently contained brute-force, lateral movement, device-evasion, and impossible-travel scenarios that static RBAC or Conditional Access could not. Containment was measurable, and false positives remained negligible (1 in 55 trials). Evidence: Chapter XXV-0f, §Interpretation of Results.
- **RQ2 (Continuous enforcement):** Enforcement was effective but asymmetric: revocations in the cloud completed within 9-13 seconds where CAE was supported, while on-premises disables averaged close to one minute. This confirms feasibility of continuous enforcement in principle, but also highlights latency as a structural barrier. Evidence: Chapter XXV-0f, §Latency Analysis.
- **RQ3 (Operational and compliance implications):** Enforcement was audisidewaystable, with authoritative artifacts produced in Entra, AD, Wazuh, and SonicWall. However, evidence was fragmented and required SIEM correlation, creating operational and audit complexity. Compliance alignment with NIS2 and ISO/IEC 27001 was demonstrated, but GDPR privacy concerns arise from peer propagation. Evidence: Chapter XXV-0f, §Compliance Implications.
- **RQ4 (Limitations and trade-offs):** Structural constraints were observed: residual exposure windows in active sessions, bounded peer propagation, dependency on custom automation, and Microsoft-centric scope. These clarify that while hybrid enforcement is achievable, it is not production-ready without vendor-native integration and governance processes. Evidence: Chapter XXV-0f, §Limitations and §Operational Challenges.

In sum, all four research questions were answered affirmatively, but within the limits of a small-scale, Microsoft-centric prototype. The thesis therefore contributes not a turnkey solution, but a feasibility proof that narrows the gap between Zero Trust theory and operational hybrid enforcement.

XXXVII. Implications for Research and Practice

The HIZT-R prototype demonstrates both what is achievable with current hybrid IAM tooling and where systemic barriers remain. Its significance extends across three domains: research, practice, and regulation.

Implications for Research

This work advances Zero Trust and RAdAC discourse from conceptual models to reproducible practice. The unified RiskScore, bidirectional enforcement logic, and measured latency profiles establish a baseline for empirical comparison in future studies. At the same time, several gaps remain open:

- **Multi-IdP interoperability:** The prototype was Microsoft-centric, extending enforcement across Okta, Ping, or Google IAM is unaddressed and requires standardized cross-provider risk propagation.
- **Identity classes:** Service accounts, machine identities, and CI/CD pipelines remain outside the scope of current adaptive enforcement.
- **Advanced adversary models:** Insider misuse, Kerberos abuse, and supply-chain attacks were not simulated and remain research targets.
- **Privacy-preserving peer propagation:** Linking users by shared device context raises data-protection questions that need formal study and governance models.

HIZT-R should therefore be read not as a final solution, but as a reference implementation against which future models can be benchmarked.

Implications for Practitioners

For enterprise defenders, the results show that hybrid risk-adaptive enforcement is technically feasible but operationally fragile. The pipelines measurably reduced brute-force, lateral movement, and credential-reuse risks, yet required brittle scripts, complex SIEM correlation, and fragmented logging. The prototype thus functions as a blueprint: it illustrates what is possible with today's tools, while warning that production adoption requires vendor-native integration, resilient automation, and governance processes to manage automated quarantines.

Implications for Regulators and Auditors

From a compliance standpoint, the prototype produced audisidewaystable artifacts that align with NIS2 (continuous authentication), ISO/IEC 27001 (access monitoring), and GDPR (appropriate technical measures). However, structural tensions remain:

- **Latency gap:** Minute-scale on-premises disables challenge strict interpretations of "continuous" authentication.
- **Privacy:** Peer propagation, by linking users through shared endpoints, may be interpreted as association monitoring under GDPR, necessitating Data Protection Impact Assessments (DPIAs).
- **Audit complexity:** Dispersed evidence across four systems (Entra, AD, SonicWall, Wazuh) complicates reproducibility and increases regulatory risk unless centralized SIEM pipelines are in place.

Synthesis

The implications are dual: for researchers, HIZT-R provides a measurable baseline for hybrid IAM studies, for practitioners, it shows both the potential and fragility of current enforcement, for regulators, it highlights the tension between audit sufficiency and technical ideals. The gap between conceptual maturity and operational sustainability is narrowed, but not yet closed.

XXXVIII. Future Work

The HIZT-R prototype confirmed that hybrid risk-adaptive enforcement is possible, but it also exposed structural barriers that prevent sustainable enterprise deployment. Several avenues for future work follow directly from these findings.

Vendor-Native Integration

Enforcement in this thesis depended on PowerShell scripts polling Entra ID and invoking Graph API calls. While sufficient for feasibility, this approach is brittle at scale. Future work should pursue vendor-native interfaces for bidirectional risk propagation, removing dependence on glue code. Standardization under SCIM or related protocols would provide a sustainable foundation for interoperable hybrid enforcement.

Extending Risk Coverage

The current RiskScore addressed brute-force, lateral movement, device evasion, and impossible travel. Many classes remain untested: service accounts, machine identities, CI/CD pipelines, and multi-cloud environments (AWS IAM, GCP IAM). Evaluating whether adaptive enforcement holds under these conditions is critical for broader applicability.

Richer Security Tooling

The prototype relied on basic Defender for Endpoint telemetry, without advanced Endpoint Detection and Response (EDR) or Attack Surface Reduction (ASR). Incorporating these signals could reduce reliance on SIEM correlation delays and shorten on-prem latency. Similarly, integrating Entra Privileged Identity Management (PIM) would improve lifecycle governance for privileged roles, while Entra Application Proxy could extend adaptive enforcement to legacy on-premises applications.

Multi-IdP Interoperability

The prototype was deliberately Microsoft-centric. Extending enforcement across other major IdPs (Okta, Ping, Google) remains an open challenge. Achieving true hybrid enforcement across providers requires standardized cross-IdP APIs for risk signal exchange and consistent policy interpretation.

Usability and Privacy Studies

The evaluation was short-term and attack-focused. Longitudinal studies are needed to measure false-positive rates, user friction, and operational costs over extended periods. Peer-aware risk propagation in particular raises privacy concerns, as linking accounts by shared devices may constitute association monitoring. Future work should explore governance models that retain containment benefits while safeguarding user rights.

Operational Pathways

Sustainable adoption requires more than technical refinement. Organizations need centralized SIEM pipelines, governance frameworks for automated quarantines, and compliance procedures that embed adaptive IAM into formal audits. Without these pathways, technically sound models risk collapsing under enterprise scale or regulatory scrutiny.

XXXIX. Closing Statement

This thesis demonstrated that hybrid identity systems can be defended more effectively by combining Zero Trust principles with risk-adaptive enforcement. Through the HIZT-R prototype, enforcement pipelines were implemented across both cloud and on-premises domains, peer-aware propagation reduced lateral-movement exposure, and a unified RiskScore provided reproducible thresholds for action. Evaluation confirmed measurable reductions in attack success rates with minimal disruption to compliant users.

The contribution of this work is not presenting a production-ready solution, rather it's in proving feasibility under real-world constraints. HIZT-R operationalized hybrid risk-adaptive enforcement with current tools and licensing, showing concretely both

what is achievable today and what remains unresolved. The results make clear why enterprises continue to rely on fragmented controls: residual exposure windows, reliance on brittle automation, fragmented audit evidence, and vendor silos remain systemic barriers.

HIZT-R should therefore be understood as a feasibility proof that narrows the gap between Zero Trust theory and enterprise practice. It demonstrates that hybrid adaptive enforcement can be achieved today, but also clarifies that sustainable deployment will require vendor-native support, richer telemetry, multi-IdP interoperability, and integrated governance. The path from conceptual models to operational resilience is incomplete, but this work provides both evidence of progress and a roadmap for advancing the state of hybrid identity security.

APPENDIX

Environment and Configurations

This appendix documents the lab environment, parameters, and configuration artifacts.

TESTBED PARAMETERS AND FIXED FACTS

Canonical Distinguished Names (DNs)

```
Domain DN:                  DC=domain,DC=local
Quarantine on-prem OU:      OU=ad_quarantine,OU=mfac,DC=domain,DC=local
Quarantine cloud OU:        OU=cl_quarantine,OU=mfac,DC=domain,DC=local
VPN Access Group:           CN=vpn_access,OU=Groups,DC=domain,DC=local
```

Software and Component Versions

Lab Host Specifications

- Hypervisor: VMware ESXi 8.0 (single host)
- Domain Controller VM: 4 vCPU, 8 GB RAM, 60 GB disk
- Wazuh SIEM VM: 4 vCPU, 8 GB RAM, 80 GB disk
- Test Clients: Windows 10/11 VMs (2 vCPU, 4 GB RAM each)
- SonicWall VPN: IKEv2, AES-256/SHA2-256, DH Group 14, lifetime 3600s
- MFA method: Microsoft Authenticator push notification
- VPN split-tunnel: Disabled; all traffic routed via gateway

Risk Score Formula and Thresholds

For reproducibility, the unified hybrid risk score used in Chapters XII and XVII is restated here:

$$\textbf{RiskScore} = 2S + 1F + 3T + 4D + 2H$$

Thresholds: $\tau_1 = 15$ (revoke/reset), $\tau_2 = 25$ (disable+quarantine).

Timekeeping and Timestamp Policy

- All logs normalized to UTC; local lab time Europe/Berlin (UTC+2 in summer).
- NTP sync sources: `pool.ntp.org` + `time.windows.com`.
- Max observed drift: <200 ms across all hosts.

NETWORK AND VPN PARAMETERS

- Lab subnet: 192.168.5.0/24; DC at 192.168.5.30.
- SonicWall VPN: IKEv2, AES-256/SHA2-256, enforced MFA.
- Split tunnel disabled; DNS suffix enforced: `domain.local`.

LOGGING AND SIEM PARAMETERS

- Windows EventIDs consumed: 4624, 4625, 4648, 4672, 4725, 4738.
- NXLog transport: UDP 514 for SonicWall, EventChannel for Windows logs.
- Wazuh rules of interest: 60122 (disable-account), 60204 (logon failures).
- Retention: 30 days (rotating JSON logs on DC).

IDENTITY OBJECTS AND API SCOPES

```
Quarantine AD OU:   OU=ad_quarantine,OU=mfac,DC=dom
Quarantine AAD Group: grp_quarantine
Conditional Access Policies: CA-Block-NonCompliant,
Intune Compliance Baseline: WIN-BaseCompliance
```

Microsoft Graph delegated scopes:

- User.ReadWrite.All
- GroupMember.ReadWrite.All
- IdentityRiskyUser.Read.All
- Mail.Send

SECURITY BASELINES AND GPOS

- Intune compliance baseline: BitLocker, Defender AV, Firewall ON.
- Intune App Protection: Managed Outlook/Teams only on BYOD, copy/paste restricted.
- Windows Hello for Business enforced via GPO (domain-joined).

TOPOLOGY AND SIGNAL FLOW

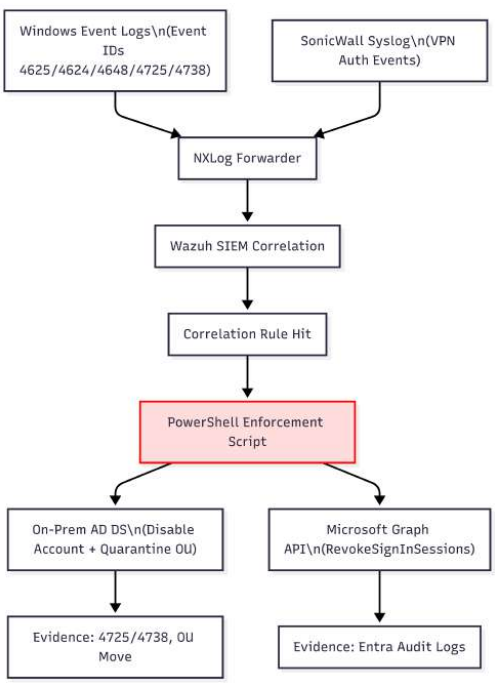


Figure 11: SIEM alert workflow: Wazuh correlation triggers PowerShell enforcement scripts and Graph API actions.

End-to-End Risk Signal Flow

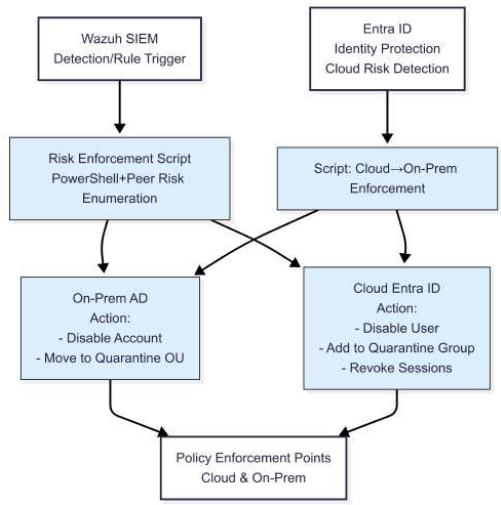


Figure 12: Risk signal flow across cloud and on-prem enforcement components.

Automation Scripts

This appendix contains the PowerShell scripts execution proof that automated hybrid enforcement(full script included as files).

POWERSHELL REQUIREMENTS FOR ENFORCEMENT SCRIPTS

The bidirectional enforcement scripts (Cloud→On-Prem, On-Prem→Cloud, Restore, and Peer Propagation) require PowerShell 7 and the following modules:

- `ActiveDirectory`
- `Microsoft.Graph`
- `Microsoft.Graph.Users`
- `Microsoft.Graph.Users.Actions`
- `Microsoft.Graph.Groups`
- `Microsoft.Graph.Mail`

Required Microsoft Graph scopes: `IdentityRiskyUser.Read.All`, `User.ReadWrite.All`, `Group.ReadWrite.All`, `Mail.Send`.

Prerequisites

- RSAT installed on the Windows host for the `ActiveDirectory` module.
- An Entra ID account with sufficient privileges to:
  - Disable/enable users,
  - Revoke sessions,
  - Add members to groups,

  - Send mail (service account or automation identity).

- Network access from the host to Entra ID Graph endpoints.

CLOUD→ON-PREM ENFORCEMENT

CloudRiskEnforcement.ps1

a) Execution Context:: Invoked every 5 minutes via Windows Task Scheduler on the Domain Controller.

execution proof:

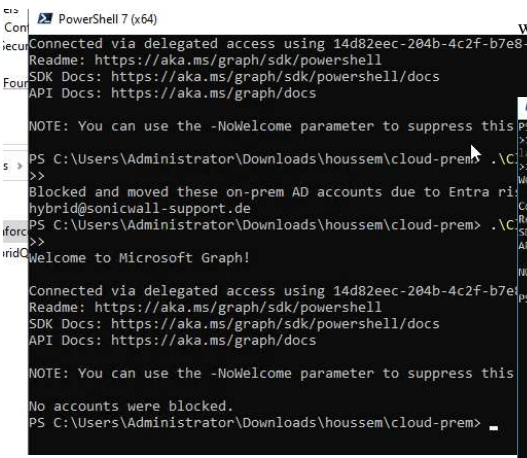


Figure 13: CloudRiskEnforcement.ps1 execution in PowerShell ISE

Restore-HybridQuarantineUsers.ps1

b) Execution Context:: Invoked manually on premis to restore risky users that are no longer risky on cloud. **execution proof:**

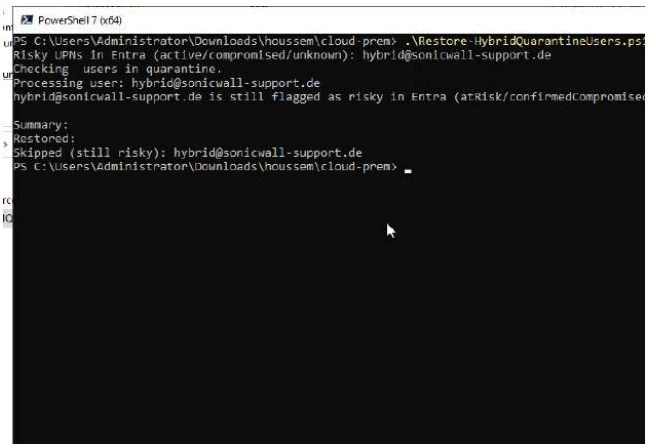


Figure 14: Restore-HybridQuarantineUsers.ps1 execution in PowerShell ISE

ON-PREM→CLOUD ENFORCEMENT

WazuhToCloudEnforcement.ps1

c) Execution Context:: trggered automatically when wazuh detects an issue.

execution proof:

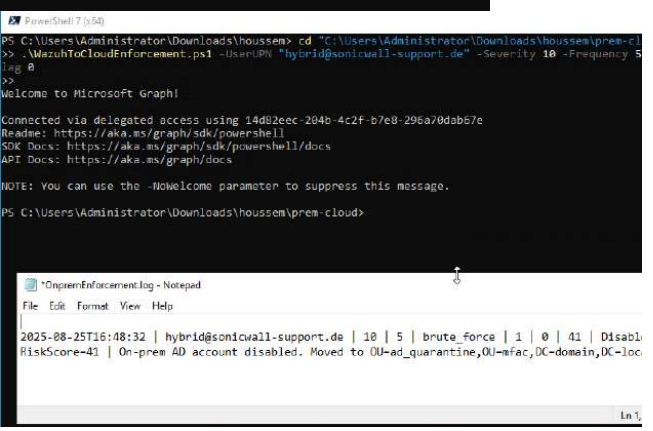


Figure 15: WazuhToCloudEnforcement.ps1 execution in PowerShell ISE

PEER-PROPAGATED RISK

PeerDiscovery.ps1

d) Execution Context:: Triggered automatically by Wazuh active response on correlated Windows Security Events.

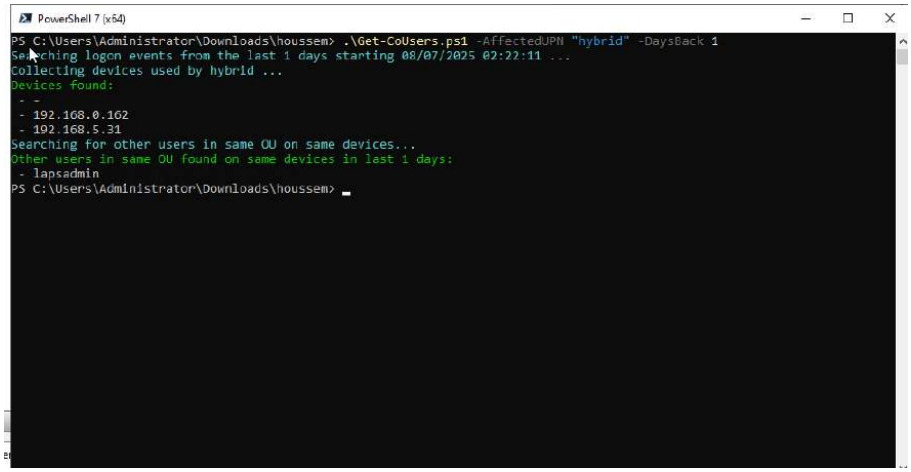


Figure 16: Peer-propagation detection.

Scenario Trials and Enforcement Evidence

This appendix presents detailed trial evidence for scenarios S0–S4 defined in Chapter XVII.

S0 - BASELINE (COMPLIANT USER)

Representative Log

```
2025-08-03T10:22:14 | houssemtest@sonicwall-
    support.de | 0 | 0 | other | 0 | 0 | 0 |
    Allow | Compliant login, no enforcement
    triggered.
```

S1 - CLOUD BRUTE-FORCE

Representative Logs

```
2025-08-03T18:06:51 | hybrid@sonicwall-support.
    de | 14 | 7 | brute_force | 0 | 0 | 47 |
    Disabled AD + Entra account. | Quarantined
    OU=ad_quarantine. =58s

2025-08-03T18:07:05 | houssemtest@sonicwall-
    support.de | 10 | 3 | brute_force | 0 | 1 |
     39 | Revoked Entra sessions + forced
    password reset. | Entra audit confirmed.
    =10s

2025-08-03T18:08:02 | hybrid@sonicwall-support.
    de | 11 | 2 | brute_force | 0 | 0 | 37 |
    Disabled AD account in error. | False
    positive logged, =60s
```

S2 - On-Prem Lateral Movement with Peer Propagation

Representative Logs

```
2025-08-03T11:13:02 | admin@sonicwall-support.
    de | 8 | 3 | lateral_movement | 0 | 0 | 29
    | Disabled AD + Entra account. | Peer
    containment triggered. =62s

2025-08-03T11:13:15 | lapsadmin@sonicwall-
    support.de | 6 | 1 | peer_propagated | 0 |
    0 | 18 | Revoked Entra sessions + forced
    password reset. | Linked to admin account.
    =60s

2025-08-03T11:14:09 | houssemtest@sonicwall-
    support.de | 7 | 2 | lateral_movement | 0 |
     0 | 26 | Disabled AD account only. | No
    peers within 24h window. =63s
```

S3 - Endpoint-Origin Policy Evasion (Non-Compliant Device)

Representative Logs

```
2025-08-03T14:21:40 | houssemtest@sonicwall-
    support.de | 5 | 1 | device_noncompliant |
    1 | 0 | 20 | Conditional Access block. |
    Device=UNMANAGED-ANDROID, login denied. =2s
2025-08-03T14:22:07 | houssemtest@sonicwall-
    support.de | 6 | 2 | device_noncompliant |
    1 | 0 | 23 | Revoked sessions + forced
    password reset. | CAE enforcement applied.
    =7s
2025-08-03T14:22:15 | houssemtest@sonicwall-
    support.de | 4 | 1 | device_noncompliant |
    1 | 0 | 17 | VPN access denied. | SonicWall
     syslog confirm, group policy block. =1s
```

S4 - Impossible Travel (Stolen Credentials)

Representative Logs

```
2025-08-03T15:33:12 | hybrid@sonicwall-support.
    de | 12 | 2 | impossible_travel | 0 | 1 |
    41 | Revoked Entra sessions. | Triggered by
     geo-anomaly detection. =12s

2025-08-03T15:34:15 | hybrid@sonicwall-support.
    de | 14 | 3 | impossible_travel | 0 | 1 |
    49 | Disabled AD account + moved to OU=
    ad_quarantine. | VPN access denied. =60s
```

References

[1] CISA. *Hybrid Identity Solutions Architecture*. 2023. URL: https://www.cisa.gov/sites/default/files/2023-03/csso-scuba-guidance_document-hybrid_identity_solutions_architecture-2023.03.22-final.pdf.

[2] Verizon. *2024 Data Breach Investigations Report (DBIR)*. 2024. URL: https://www.verizon.com/dbir/.

[3] Cybersecurity and Infrastructure Security Agency. *AA20-352A: Advanced Persistent Threat Compromise of Government Agencies, Critical Infrastructure, and Private Sector Organizations*. Tech. rep. CISA, 2020. URL: https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-352a.

[4] Ravi S. Sandhu et al. "Role-Based Access Control". In: *IEEE Computer* 29.2 (1996), pp. 38–47. URL: https://profsandhu.com/articles/advcom/a98rbac.pdf.

[5] John Kindervag. *No More Chewy Centers: Introducing the Zero Trust Model of Information Security*. Tech. rep. Forrester Research, 2010. URL: https://media.paloaltonetworks.com/documents/Forrester-No-More-Chewy-Centers.pdf.

[6] Scott Rose et al. *Zero Trust Architecture*. Tech. rep. NIST SP 800-207. National Institute of Standards and Technology, 2020. DOI: 10.6028/NIST.SP.800-207. URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf.

[7] S. Rose et al. *SP 1800-35: Implementing a Zero Trust Architecture*. Tech. rep. NIST NCCoE, 2025. URL: https://csrc.nist.gov/pubs/sp/1800/35/final.

[8] European Union. *Directive (EU) 2022/2555 (NIS 2 Directive)*. Tech. rep. EUR-Lex, 2022. URL: https://eur-lex.europa.eu/eli/dir/2022/2555/oj/eng.

[9] ISO/IEC JTC 1/SC 27. *ISO 27001:2022 Controls – Access Control and Authentication*. 2025. URL: https://www.iso.org/standard/27001?utm_source=chatgpt.com.

[10] European Union. *Regulation (EU) 2016/679 (GDPR) — Article 32: Security of Processing*. Tech. rep. EUR-Lex, 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.

[11] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. "More Than Just Good Passwords? A Study on Usability and Security Perceptions of Risk-based Authentication". In: *Annual Computer Security Applications Conference (ACSAC)*. 2020, pp. 203–218. URL: https://d-nb.info/1219138525/34.

[12] Hany F. Atlam, Robert J. Walters, and Gary B. Wills. "Risk-Based Access Control: A Systematic Literature Review". In: *Future Internet* 12.6 (2020), p. 103. URL: https://www.mdpi.com/1999-5903/12/6/103.

[13] Cybersecurity and Infrastructure Security Agency. *Zero Trust Maturity Model v2*. 2023. URL: https://www.cisa.gov/sites/default/files/2023-04/CISA_Zero_Trust_Maturity_Model_Version_2_508c.pdf.

[14] David Elliott Bell and Leonard J. LaPadula. *Secure Computer Systems: Unified Exposition and Multics Interpretation*. Tech. rep. MITRE Technical Report. MITRE Corporation, 1976. URL: https://csrc.nist.gov/files/pubs/conference/1998/10/08/proceedings-of-the-21st-nissc-1998/final/docs/early-cs-papers/bell76.pdf.

[15] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. "Proposed NIST Standard for Role-Based Access Control". In: *ACM Transactions on Information and System Security (TISSEC)* 4.3 (2001), pp. 224–274. URL: https://csrc.nist.gov/pubs/journal/2001/08/proposed-nist-standard-for-rolebased-access-contro/final.

[16] Vincent C. Hu, David F. Ferraiolo, and Rick Kuhn. *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*. Tech. rep. NIST Special Publication 800-162. National Institute of Standards and Technology, 2014. URL: https://doi.org/10.6028/NIST.SP.800-162.

[17] Yue Zhang, Bharath Srinivasan, and Gail-Joon Ahn. "Permission Usage-Based Neural Networks for Risk-Aware Access Control in Android". In: *Proceedings of the 33rd ACM/SIGAPP Symposium on Applied Computing (SAC '18)*. ACM, 2018, pp. 1625–1632. DOI: 10.1145/3167132.3167321. URL: https://dl.acm.org/doi/10.1145/3167132.3167321.

[18] Microsoft. *What is Hybrid Identity*. Microsoft Docs. 2025. URL: https://learn.microsoft.com/en-us/entra/identity/hybrid/whatis-hybrid-identity.

[19] Microsoft. *Hybrid Identity Authentication Methods*. Microsoft Docs. 2023. URL: https://learn.microsoft.com/en-us/entra/identity/hybrid/connect/choose-ad-authn.

[20] B. Ward and B. Beyer. "BeyondCorp: A New Approach to Enterprise Security". In: *;login:* 39.6 (2014), pp. 6–17. URL: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43231.pdf.

[21] MITRE ATT&CK. *Technique T1550.002: Pass the Hash*. 2025. URL: https://attack.mitre.org/techniques/T1550/002/.

[22] Microsoft. *Detect and remediate illicit consent grants*. 2025. URL: https://learn.microsoft.com/en-us/defender-office-365/detect-and-remediate-illicit-consent-grants.

[23] S. Teerakanok, S. Phimoltares, and E. Rattanalerdnusorn. “Migrating to Zero Trust Architecture: Reviews and Challenges”. In: *Security and Communication Networks* 2021 (2021), pp. 1–16. DOI: 10.1155/2021/9947347. URL: https://doi.org/10.1155/2021/9947347.

[24] Identity Defined Security Alliance. *2022 Trends in Securing Digital Identities*. 2022. URL: https://www.idsalliance.org/white-paper/2022-trends-in-securing-digital-identities/.

[25] Qihua Wang et al. “An Attribute-Based Framework for Risk-Adaptive Access Control”. In: *2016 IEEE Trustcom/BigDataSE/ISPA*. 2016, pp. 1713–1718. URL: https://ieeexplore.ieee.org/document/7840983.

[26] ENISA. *Implementation guidance on security measures*. 2025. URL: https://www.enisa.europa.eu/publications/nis2-technical-implementation-guidance.

[27] Robert McGraw. *Risk-Adaptable Access Control (RAdAC)*. Tech. rep. National Security Agency, 2009. URL: https://csrc.nist.gov/csrc/media/events/privilege-management-workshop/documents/radac-paper0001.pdf.

[28] Microsoft. *What is Conditional Access in Microsoft Entra ID?* 2025. URL: https://learn.microsoft.com/en-us/entra/identity/conditional-access/overview.

[29] Philipp Markert et al. ““As Soon as It’s a Risk, I Want to Require MFA”: How Administrators Configure Risk-based Authentication”. In: *18th Symposium on Usable Privacy and Security (SOUPS 2022)*. 2022, pp. 41–62. URL: https://www.usenix.org/system/files/soups2022-markert.pdf.

[30] ENISA. *Digital Identity Standards*. 2021. URL: https://www.enisa.europa.eu/publications/digital-identity-standards.

[31] OASIS. *Security Assertion Markup Language (SAML) V2.0 — OASIS Standard*. 2005. URL: https://www.oasis-open.org/standard/saml/.

Survey and Benchmarks of Lightweight Cryptographic Algorithms for IoT Communication in Power Distribution Systems

Zakire Çukur
Department of Management Information Systems
Kadir Has University
Istanbul, Türkiye
zcukur@stu.khas.edu.tr

Mert İlhan Ecevit
CCIP, Center for Cyber Security and
Critical Infrastructure Protection, Kadir Has University
Istanbul, Türkiye
mertilhan.ecevit@khas.edu.tr

Oğuzhan Ceylan
Department of Management Information Systems
Kadir Has University
Istanbul, Türkiye
oguzhan.ceylan@khas.edu.tr

Hasan Dağ
CCIP, Center for Cyber Security and
Critical Infrastructure Protection, Kadir Has University
Istanbul, Türkiye
hasan.dag@khas.edu.tr

***Abstract*—Modern distribution feeders now carry high-rate telemetry from meters, pole-top sensors, and recloser controllers, but protection commands must still reach breakers within the 3 ms transfer-time limit (IEC 61850 GOOSE Type 1A P2/P3). Conventional cryptography overwhelms the modest processors and batteries inside these devices; thus, there is a need for lightweight alternatives. To this end, we evaluated eleven lighter alternatives—covering block, stream, and elliptic-curve designs—against practical criteria of security strength, execution time, circuit area, and energy use. Reported laboratory benchmarks indicate that the block ciphers SIMON and PRESENT fit comfortably into smart meters and basic sensors with almost no impact on battery life, while a 120 MHz mid-range microcontroller can run the authenticated cipher Ascon fast enough to secure gateway traffic without delaying protection frames. Published measurements indicate Ascon resists replay and adds only a small battery overhead. These results give distribution engineers a clear guide for selecting encryption that meets real-time protection deadlines while supporting the long-term sustainability and reliability of electrical power systems.**

***Index Terms*—Lightweight cryptography, smart grid IoT security, IEC 61850 GOOSE protection, energy-efficient encryption, distribution-grid resilience**

I. Introduction

The uninterrupted delivery of electrical energy underpins every facet of contemporary society—from hospital ventilators to cloud-computing data centers. Even a short-term outage can cascade into human-safety and huge economic losses. In response, utilities are digitalizing the distribution grid at an unprecedented scale, embedding millions of Internet-of-Things (IoT) sensors, intelligent electronic devices (IEDs), and remote-terminal units (RTUs) along feeders and inside substations. Although pervasive connectivity allows fine-grained automation and predictive maintenance, it simultaneously widens the attack surface, as dramatically illustrated by coordinated cyberattacks on the Ukrainian power network in 2015-2016 that interrupted service for more than 225000 customers and exposed latent security gaps in substation automation layers [1]. Protection traffic (IEC 61850 GOOSE Type 1A) must still meet the P2/P3 performance-class transfer time of 3 ms (back-to-back) even after encryption [2].

Modern distribution networks depend on industrial control systems (ICS) that connect equipment like programmable logic controllers, protection relays, and tap changers with higher-level data and analytics platforms. According to the Purdue reference model (Levels 0 to 3), which is part of the ISA-95 standard, these systems are organized into layers—from the physical equipment at Level 0 up to the operations and DMZ layer at Level 3—to help manage and secure the system more effectively. At the lower levels, tasks such as sensing and control are handled by low-power IoT devices. These devices have a very modest CPU (often under 100 MHz), limited memory (e.g., 64 kB flash/16–64 kB RAM), and are often powered by batteries or energy harvesting. Despite their small size, these edge devices are actively involved in real-time control and must send and receive trusted data over communication channels that are often limited in bandwidth [3]. For example, a typical station-bus link carries 60-byte GOOSE PDUs every 100 ms on a 100 Mbit/s IEC 61850 VLAN.

Because these small devices have very limited memory, processing power, and battery life, traditional encryption methods like RSA or even full versions of AES are often too heavy to use effectively [4], [5]. For this reason, many studies on smart grid cybersecurity highlight the need for lightweight cryptography, which is more efficient to protect data that is still protected against threats such as brute-force attacks, side channel leaks, and denial-of-service events [6], [7]. In short,

the IoT layer of the power grid needs encryption tools that are designed to use less energy, work well on small battery-powered devices, and still provide strong security.

This paper focuses on lightweight cryptographic techniques for IoT devices in power distribution networks and makes the following three contributions:

1) **Evaluation framework.** We propose a four component scoring system including security strength, speed, cost to implement, and energy use, based on existing IoT research and customized for the needs of power grids.
2) **Holistic comparison.** We analyze eleven well-known encryption algorithms, including block ciphers, stream ciphers, elliptic curve methods, hybrids, and new designs. These are grouped by their current status: real-world use, lab prototypes, or finalists in the NIST Lightweight Cryptography Competition.
3) **Deployment guidance.** We provide a mapping between encryption options and the different layers of the grid (based on the Purdue model), showing which ciphers work best with specific devices and system levels with which engineers can choose what fits best instead of relying on one-size-fits-all solutions.

The remainder of the paper is structured as follows. Section I outlines the four-layer IoT security model for power grids. Section II classifies lightweight algorithms by design lineage. In Section III, we introduce the metric framework and present comparative results. In Section IV, we discuss trade-offs and future trends, and finally in Section V, we conclude with actionable recommendations for practitioners.

II. IoT Architecture and Security Requirements

Most IoT reference frameworks, including International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) Recommendation Y.2060 [8], NIST Interagency/Internal Report (NISTIR) 8259, and the Purdue model used in power systems describe a four layer structure that matches the physical layout of a distribution substation (from Level 0 to Level 3). Figure 1 illustrates this alignment and gives examples of the lightweight security tools commonly used at each level.

A. Four-Layer Structure

1) Perception layer (Level 0–1): Since encryption hardware must fit into a tiny part of the chip, just a few square millimeters shared with other sensor components, designers are usually limited to about 2,000 logic gates (known as 2 kGE, a standard measure of chip size), which is roughly the area of 0.01 to 0.02 mm² in modern 65 nm technology. The entire device often runs on a small coin-cell battery, which means that the encryption must also be extremely energy efficient. These strict space and power limits make ultralightweight ciphers such as PRESENT [9], SIMON [4], and the LCC cipher based on cellular automata [10] especially suitable for securing data directly on these devices.

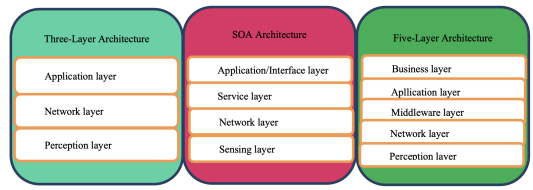


Fig. 1: Illustration of a four-layer IoT architecture for distribution automation, aligned with Purdue Levels 0 to 3.

2) Network layer (Level 1–2): Data from smart grid devices is sent to edge gateways either via wireless technologies such as LoRaWAN or Wi-SUN or via wired connections such as RS-485 or powerline communication (PLC). To protect this data during transmission, payload encryption is used, combining a lightweight version of AES (with fewer rounds) or ChaCha20 (a fast stream cipher) together with an asymmetric key exchange protocol such as Curve25519. With this approach, forward secrecy is preserved, meaning that even if a key is later compromised, previous messages remain protected within the limits of very small message sizes [5], [11].

3) Support/Middleware layer (Level 2): Gateways collect data from devices, convert between different communication protocols such as Modbus and/or Message Queuing Telemetry Transport (MQTT), and apply security rules. Instead of using custom-made encryption methods, they rely on standard security tools; Transport Layer Security (TLS) 1.3 or Datagram TLS (DTLS), which are used to protect data during transmission. These tools include Ascon based encryption, a lightweight method with strong security and lower communication which handshake overhead constrained systems [4], [12].

4) Application layer (Level 3): Systems like SCADA data loggers, outage management dashboards, and grid analytics tools use security features such as X.509 certificates, digital signatures, and hash-based message authentication to control access. By using these tools, one can be sure that data come from trusted sources and that actions can be traced back to the responsible operator.

B. Security objectives

At every layer of the system, the main security goal is to protect the CIA triad (confidentiality, integrity, and availability of data). Confidentiality means keeping sensitive information like feeder settings or breaker commands, private by encrypting the data (using methods like AES-GCM or Ascon-128a) so that outsiders can't read it. Integrity means making sure the data hasn't been changed or altered by anyone without permission. This is especially important for SCADA commands and is typically protected using hashes or digital signatures. Availability means keeping the system running reliably, especially in critical grid operations. This is supported by lightweight intrusion detection and regular firmware updates to defend against attacks that aim to block

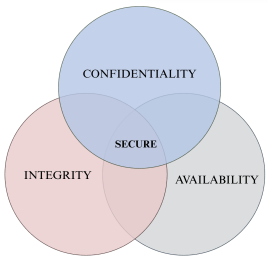


Fig. 2: CIA triad as applied to distribution-grid IoT.

or slow down important actions, like circuit breaker trips [7], [13].

C. Threat landscape

If any part of the CIA triad is weak, the power network becomes vulnerable to many different types of attacks as reported in the literature [14].

- *Network attacks:* These type of attacks are like replaying old control messages (e.g., IEC 60870-5-104), secretly changing voltage regulator settings, or overwhelming gateways with Distributed Denial of Service (DDoS) traffic;
- Cryptographic attacks: The types of attacks are such as breaking outdated encryption using brute-force or differential methods, or stealing information from poorly protected cryptographic chips through power analysis;
- *Physical attacks:* These include tampering with sensors at unmanned substations, physically extracting encryption keys from hardware, or uploading malicious firmware via exposed communication ports (e.g., universal asynchronous receiver–transmitters (UART));
- *Social and application level attacks:* Some examples are phishing of operations staff, SQL injection grid databases, which could lead to false outage reports.

Figure 2 highlights that the CIA triad is a practical design guide, influencing the selection of encryption techniques and key management approaches across all system layers.

D. Motivation for lightweight cryptography

Standard encryption algorithms like full-round AES, RC4, or RSA are too demanding for small microcontrollers with 64 kB or less of memory and processors running below 100 MHz [3], [5]. They also consume often exceed 10 μJ/bit—making them unsuitable for battery powered devices like pole-top sensors which need to run for several years. Thus, lightweight cryptographic algorithms are essential for IoT use in power grids. In the rest of the paper we evaluate a range of these lightweight options in terms of cost, speed, and security to help identify the most practical choices.

Lightweight encryption methods are especially important at three points in power system operations:

- **Smart metering:** Small data packets (around 200 bytes) are sent every 15 minutes using power-line communication or Narrowband IoT (NB-IoT), following the DLMS/COSEM protocol. Lightweight block ciphers like PRESENT meet DLMS Security Suite 1 requirements while fitting within the tight hardware limits (less than 2kGE) of typical smart meter chips [15].
- **Substation protection:** High-priority messages (like GOOSE Type 1A trip signals under IEC 61850) has to be delivered in under 3 milliseconds. The Ascon-128a cipher adds only about 5.8 microseconds of delay on a 120 MHz Cortex-M7 processor with hardware support for encryption, staying well within the required response time [16].
- **Distributed-energy resources (DER) control**— Photovoltaic inverters send regular telemetry using the SunSpec Modbus/TCP protocol, around 4 kbps. The Trivium stream cipher runs efficiently on a 32-bit RISC-V processor, using less than 2 microjoules per bit, and still meets the SunSpec 250 ms logging interval [17].

III. Classification of Lightweight Cryptographic Algorithms

Lightweight cryptographic methods can be grouped based on their design approach, including symmetric block ciphers, symmetric stream ciphers, elliptic-curve–based public key systems, hybrid constructions, and novel bio-inspired algorithms. Table I provides representative examples from each category.

TABLE I: Representative lightweight ciphers grouped by design family

Block	Stream	ECC	Hybrid	Novel
PRESENT	Trivium	Curve25519	ECC+AES+SHA	LCC
SIMON	ChaCha20	P-256	NTRU+AES+SHA	Chaotic ECC
RECTANGLE	MICKEY 2.0			Ascon
				Elephant
				GIFT-COFB
				Hummingbird-2

A. Block ciphers

Block ciphers encrypt data in fixed-size chunks (64 or 128 bits) using the same key for both encryption and decryption. Lightweight versions, such as PRESENT (31 rounds, 80-/128-bit key, about 1570 GE) and SIMON-64/128 (44 rounds, about 2000 GE), keep hardware size small by using simple bit-wise substitutions (S-boxes) and Feistel structures [9], [18]. The RECTANGLE cipher improves efficiency further by implementing its S-box with just four 4-bit lookup tables, requiring about 2070 GE while still offering 128-bit security [19]. These ciphers are strong against differential and linear attacks due to their high number of rounds. However, their iterative design introduces delay, which can be a drawback in time-sensitive applications.

B. Stream ciphers

Stream ciphers XOR each plaintext bit with a keystream generated from lightweight feedback registers. Trivium and

MICKEY 2.0 are two compact ones recognized under the ISO/IEC 29192-3 standard. Trivium uses about 2600 gate equivalents and MICKEY about 2200 GE—while providing 64-bit security, assuming a unique initialization vector (IV) is used for each session [20], [21]. ChaCha20 uses more memory but offers very high speeds (over 1 Gbps on 32-bit processors) and strong resistance to side-channel attacks [22]. However, a key risk for all stream ciphers is reusing the same keystream, which can seriously compromise security.

C. Elliptic-Curve Cryptography

ECC offers asymmetric confidentiality and authentication with keys an order of magnitude shorter than RSA. Curve25519 (32-byte keys, 128-bit security) executes a scalar multiply in ≈ 900 k cycles on Cortex-M3 and 3100 GE in dedicated hardware [23]. Hardware-friendly Koblitz curves accelerate scalar multiplication via Frobenius maps, whereas NIST P-256 enjoys widespread toolkit support [11]. Side-channel-safe arithmetic and scalable key management remain the main implementation hurdles [24].

D. Hybrid constructions

Hybrid schemes combine symmetric speed with asymmetric key establishment. A common pattern couples ECC key exchange with AES-GCM payload protection and SHA-2 message authentication [5]. Post-quantum hybrids (e.g., NTRU + AES) increase handshake cost ($<$5000 GE) but raise long-term confidentiality.

E. Novel and competition-driven designs

The NIST Lightweight Cryptography competition (2019–2023) helped to develop authenticated encryption algorithms optimized for constrained environments with limited memory, processing power, and energy resources. The winning algorithm, Ascon-128a, delivers authenticated encryption at just 11.5 clock cycles/byte and uses about 2250 GE [25]. Other finalists, like Elephant and GIFT-COFB, are optimized for even smaller hardware footprints, though with lower data rates [26], [27]. Beyond the competition, some researchers have explored more experimental designs. For example, LCC uses cellular automata to generate encryption patterns [10], and Hummingbird-2 combines features of both block and stream ciphers in a compact 2000 GE design [28]. While these novel approaches are promising, they still require more analysis to fully understand their security guarantees.

IV. Performance Metrics Under Feeder Constraints

We assess each shortlisted cipher based on four key metrics commonly used in IoT security research: security level, performance efficiency, implementation cost, and energy efficiency. The evaluation draws on data from vendor datasheets, hardware synthesis results published in CHES and ToSC, and FPGA-based power profiling studies [4], [18], [29]. Algorithms are grouped into (i) real-world deployments, (ii) experimental prototypes, and (iii) NIST LWC finalists.

Security level denotes resistance to published cryptanalysis; performance efficiency is expressed in cycles/byte and throughput; implementation cost is measured in GE; energy efficiency refers to μJ/bit drawn at nominal voltage. Table II defines standard thresholds for classifying each metric.

TABLE II: Metric thresholds for lightweight-cipher evaluation

Metric	Low	Medium	High
Security level	Broken / major weaknesses	Resists basic attacks	Standardized, resists advanced attacks
Perf. Class	$>$500 cy/B or $<$10 kbps	200–500 cy/B	$<$200 cy/B or $>$50 kbps
Impl. cost	$<$2000 GE	2000–3000 GE	$>$3000 GE
Energy eff.	$>$5 μJ/bit	1–5 μJ/bit	$<$1 μJ/bit

A. Real-world implementations

TABLE III: Benchmarked ciphers already deployed in industry

Algorithm	Sec.	Perf. (cy/B)	Cost (GE)	Energy (μJ/bit)
PRESENT	High	80	$<$2000	1.5 [9], [29]
SIMON	High	74	$\sim$2000	1.9 [18]
SPECK	Med.	92	$\sim$1800	1.5 [18]
Curve25519	High	388	$\sim$3100	9.1 [23]

Table III confirms that PRESENT and SIMON combine strong security with sub-2 kGE footprints, ideal for sensor nodes. Curve25519 offers robust asymmetric features yet incurs a three-fold energy penalty. Overall, SIMON delivers the best balance for deeply embedded OT devices.

B. Experimental designs

TABLE IV: Prototype ciphers under active research

Algorithm	Sec.	Perf. (cy/B)	Cost (GE)	Energy (μJ/bit)
LCC	Med.–H	100	$<$2000	2.0 [10]
Hummingbird-2	Med.	90	$\sim$2000	1.8 [28]
Chaotic ECC	High	300	$<$3500	10.0 [11]
Mod. PRESENT-256	High	80	$<$2000	1.5 [9]

Experimental schemes aim to push beyond PRESENT class efficiency. Modified PRESENT-256 preserves sub-2 kGE size while doubling key length; LCC and Hummingbird-2 prioritize speed but need further cryptanalysis. Chaotic ECC provides post-quantum promise at notable area and energy cost.

C. NIST LWC finalists

TABLE V: Metrics for NIST LWC standard candidates

Algorithm	Sec.	Perf. (cy/B)	Cost (GE)	Energy (μJ/bit)
Ascon-128a	High	11.5	$\sim$2250	1.8 [25]
GIFT-COFB	High	20.0	$\sim$2000	1.4 [27]
Elephant	Med.	18.0	$\sim$2000	1.7 [?]

In 2024, NIST selected Ascon as the main standard for lightweight encryption, with strong 128-bit security and fast performance at just 11.5 cycles/byte [30]. GIFT-COFB uses even less chip area, and Elephant is designed for devices with

very limited memory, though it offers slightly lower security. Due to its balance of speed and security, Ascon is preferred for protecting power-distribution gateways.

V. Discussion: Toward an Ideal Lightweight Algorithm

Tables III–V and Figure 3 confirm that the performance envelope of lightweight cryptography is multi-dimensional. Device classes emphasize different constraints: *flash size* on Layer 0 sensors, *duty-cycle energy* on battery-powered meters, or *handshake latency* on substation gateways. Consequently, no single primitive can serve every operating point.

A. Constraint-driven selection

Devices in power networks have limited resources in three main ways:

- **Silicon/area** Many small sensors can only spare a tiny amount of hardware—less than 2kGE logic gates, so they need very compact encryption methods, like those used in DLMS/COSEM standards with 128-bit keys and ultra-low yearly energy use.
- **Energy use** Since many devices run on batteries, encryption methods must use enrgy less than 1 µJ/bit to avoid draining power needed for other tasks. This is especially important for systems like IEC 61850, where the processor must stay within a tight energy budget.
- **Speed/delay** Devices that control the grid in real-time must encrypt data in just a few dozen processor cycles. On the other hand, devices uploading data to central systems (like a Distribution Management System) can handle a bit more delay, but they may need stronger security like full TLS handshakes.

B. Recommendations

- **Level 0-1 (Field sensors)** For small, low-power devices like pole sensors or meters, SIMON-64/128 and Modified PRESENT-256 are good choices as they offer strong 128-bit security while using little space and ideal energy for sensors running on batteries or harvested energy that must last for years without maintenance.
- **Level 2 (Gateways and RTUs)** At the gateway level, Ascon-128a which provides both encryption and authentication quickly and efficiently using only 11.5 cycles/byte and fitting well into hardware already in use, is ideal. It works well with protocols like IEC 104 and MQTT and has already been adopted in secure industrial hardware.
- **Level2–3 (Control and management stations)** For systems where secure key exchange and long-term data protection matter most, Curve25519 or newer approaches like Chaotic-ECC can be preferred. These should be paired with protections against side-channel attacks and upgraded to post-quantum options like CRYSTALS-Kyber to stay secure in the future [31].

C. Trade-offs and emerging trends

Security vs. energy. Block and stream ciphers like SIMON and Trivium use low energy, but their short keys necessitate periodic rotation. ECC and post-quantum hybrids use more energy but offer stronger long-term protection.

Latency vs. area. Ascon and GIFT-COFB are fast (less than 20 cycle/byte) and don't cover much space on chip. Elephant saves more memory at the expense of lower security.

Future-proofing Future trend is to combine small, energy-efficient block ciphers for protecting data with lightweight, quantum-safe methods for key exchange. This "small + small" strategy keeps the system compact and efficient, while also preparing it for future quantum-era threats.

Two efforts are key to progress: First, energy models should be validated through hardware tests on processors like RISC-V and Cortex-M55 with Helium. Second, experimental ciphers such as LCC and Hummingbird-2 need formal leakage analysis, and real-world pilots should explore combining Ascon or GIFT-COFB with post-quantum key exchange in hybrid handshakes, as recommended by NIST [30].

D. Case Study: Pole-Top Recloser on a 13-kV Feeder

A 120-MHz Cortex-M33 controller logs three-phase currents every 100 ms and multicasts a 60-byte GOOSE trip PDU. Table VI compares crypto latency and battery impact (ER18505 Li-$SOCl_2$, 3.2 Ah). Both ciphers satisfy the 3 ms P2/P3 window; Ascon's added silicon is justified where protection speed trumps battery margin and therefore comply with IEC 61850-8-1 P2/P3.

TABLE VI: Impact of cipher choice on recloser controller

Cipher	Latency (µs)	Energy/bit (µJ)	Est. battery life
None (baseline)	0	–	9.4 y
SIMON-64/128	37	1.9	9.1 y
Ascon-128a	5.8	1.8	8.8 y

VI. Conclusion

We proposed a four-metric evaluation framework and applied it to 11 lightweight cryptographic primitives spanning real-world deployments, laboratory prototypes, and NIST LWC finalists. SIMON-64/128 was found to offer the best balance of silicon area and energy consumption for legacy SCADA sensors. Modified PRESENT-256 extends the key length without increasing area, outperforming other experimental ciphers on ultra-constrained devices. Ascon-128a delivers the highest security-to-performance ratio among standardized AEAD schemes and is already being deployed in commercial secure elements.

Given the wide range of device constraints in modern power grids, engineers should tailor cryptographic solutions to specific memory, energy, and threat needs instead of relying on one size fits all benchmarks. Future work will extend our framework to post-quantum hybrids and validate the results on hardware testbeds simulating next-generation power distribution systems.

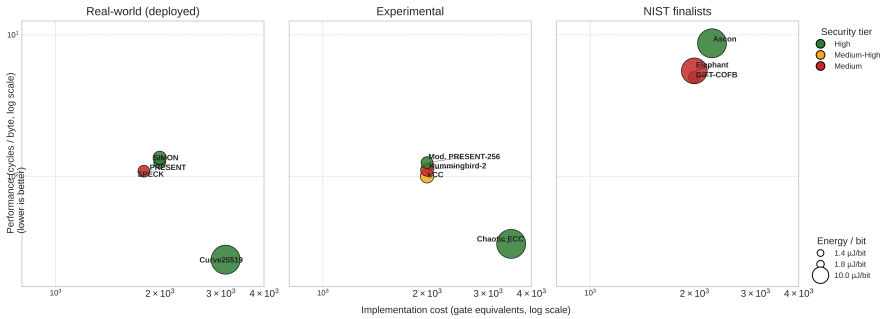


Fig. 3: Unified benchmark scatterplot. **Left panel:** real-world (deployed) ciphers; **Centre:** experimental designs; **right:** NIST LWC finalists. Axes are log–log; bubbles encode energy/bit; colour encodes security tier.

ACKNOWLEDGEMENTS

This work was supported partially by the European Union in the framework of ERASMUS MUNDUS, Project CyberMACS (Project #101082683) (https://cybermacs.eu).

REFERENCES

[1] R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the cyber attack on the ukrainian power grid," SANS Industrial Control Systems Report, 2016. [Online]. Available: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf

[2] A. P. Grids, "Utilization of IEC 61850 GOOSE messaging in protection schemes," ABB, Tech. Rep., 2017. [Online]. Available: https://library.e.abb.com/public/dc853877595c4086ae649ca29924c0ec/Paper_GOOSE%20Utilisation%20in%20Protection.pdf

[3] M. M. Rana, Q. Mamun, and R. Islam, "Lightweight cryptography in iot networks: a survey," *Future Generation Computer Systems*, vol. 129, pp. 77–89, 2022.

[4] F. Thabit, O. Can, A. O. Aljahdali, G. H. Al-Gaphari, and H. A. Alkhzaimi, "Cryptography algorithms for enhancing iot security," *Internet of Things*, vol. 22, p. 100759, 2023.

[5] F. Mallouli, A. Hellal, N. S. Saeed, and F. A. Alzahrani, "A survey on cryptography: A comparative study between rsa vs ecc algorithms, and rsa vs el-gamal algorithms," in *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud) / 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 2019, pp. 173–176.

[6] Y. Yan, Y. Ma, and W. Liu, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 303–332, 2018.

[7] P. He, Y. Zhou, and X. Qin, "A survey on energy-aware security mechanisms for the internet of things," *Future Internet*, vol. 16, no. 4, p. 128, 2024.

[8] "Overview of the internet of things," International Telecommunication Union, Tech. Rep. Recommendation Y.2060, 2012. [Online]. Available: https://www.itu.int/rec/T-REC-Y.2060-201206-I/en

[9] R. Bharathi and N. Parvatham, "Light-weight present block cipher model for iot security on fpga," *Intelligent Automation & Soft Computing*, vol. 33, no. 1, pp. 47–49, 2022.

[10] S. Roy, U. Rawat, and J. Karjee, "A lightweight cellular automata based encryption technique for iot applications," *IEEE Access*, vol. 7, pp. 39 784–39 792, 2019.

[11] A. E. Adeniyi, R. G. Jimoh, and J. B. Awotunde, "A systematic review on elliptic curve cryptography algorithm for internet of things: Categorization, application areas, and security," *Computers and Electrical Engineering*, vol. 118, p. 109330, 2024.

[12] A. A. Zainuddin, "A comprehensive analysis of iot security and privacy in smart city applications," *Smart Cities*, 2024.

[13] F. A. Alaba, M. Othman, I. A. T. Hashem, and M. A. Razzaque, "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[14] R. Roman, P. Näjera, and J. Lopez, "Securing the internet of things," *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[15] T. Knap and O. Černý, "Smart metering cybersecurity—requirements, methodology, and proposal of testing," *Sensors*, vol. 23, no. 5, p. 2521, 2023.

[16] M. G. da Silveira and P. H. Franco, "Iec 61850 network cybersecurity: Mitigating GOOSE message vulnerabilities," in *PAC World Americas Conference*, Raleigh, NC, 2019. [Online]. Available: https://cdn.selinc.com/assets/Literature/Publications/Technical%20Papers/6921_IEC61850Network_MS_20190712_Web.pdf

[17] S. Technologies, "Technical note—sunspec logging in solaredge inverters," SolarEdge, Tech. Rep., 2022. [Online]. Available: https://knowledge-center.solaredge.com/sites/kc/files/sunspec-implementation-technical-note.pdf

[18] R. B. *et al.*, "The simon and speck families of lightweight block ciphers," in *Proceedings of the 52nd Annual DAC*. ACM, 2015.

[19] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms," *Cryptology ePrint Archive*, 2014.

[20] "Information technology—security techniques—lightweight cryptography—part 3: Stream ciphers," ISO/IEC 29192-3:2023, 2023.

[21] S. Babbage and M. Dodd, "The stream cipher mickey 2.0," *ECRYPT Stream Cipher*, pp. 191–209, 2006.

[22] A. Langley and Y. Nir, "Chacha20 and poly1305 for ietf protocols," RFC 8439, 2018.

[23] G. B. Satrya, "A comparative study of post-quantum cryptographic algorithm implementations for secure and efficient energy systems monitoring," *Internet of Things*, vol. 22, p. 100759, 2023.

[24] E. Gyamfi, J. A. Ansere, and L. Xu, "Ecc-based lightweight cybersecurity solution for iot networks utilizing multi-access mobile edge computing," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2019, pp. 149–154.

[25] C. Beierle, S. Krenn, and M. E. *et al.*, "The ascon suite—lightweight authenticated encryption and hashing," in *NIST Lightweight Cryptography Finalist Report*, 2023. [Online]. Available: https://csrc.nist.gov/Projects/lightweight-cryptography

[26] T. Beyne, Y. L. Chen, C. Dobraunig, and B. Mennink, "Elephant v2," *NIST lightweight competition*, 2021.

[27] S. Banik, A. Chakraborti, A. Inoue, T. Iwata, K. Minematsu, M. Nandi, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "Gift-cofb," *Cryptology ePrint Archive*, 2020.

[28] D. Engels, M.-J. O. Saarinen, P. Schweitzer, and E. M. Smith, "The hummingbird-2 lightweight authenticated encryption algorithm," in *International workshop on radio frequency identification: Security and privacy issues*. Springer, 2011, pp. 19–31.

[29] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Shannon, "PRESENT: An ultra-lightweight block cipher," in *CHES 2007*. Springer, 2007, pp. 450–466.

[30] N. I. of Standards and Technology, "Nist announces winner of lightweight cryptography competition," https://csrc.nist.gov/Projects/lightweight-cryptography, 2024.

[31] T. M. Fernández-Caramés, "From pre-quantum to post-quantum iot security: A survey on quantum-resistant cryptosystems for the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6457–6480, 2019.

The Role of Artificial Intelligence in Predictive Maintenance for Smart Meters Through Anomaly Detection

1st Samsoon Nahar Shampa
SRH Heidelberg University of Applied Sciences
School of Technology and Architecture
Berlin, Germany
samsoonnaharshampa@gmail.com
0009-0005-7419-9256

2nd Saiful Islam
SRH Heidelberg University of Applied Sciences
School of Technology and Architecture
Berlin, Germany
saiful.islam@srh.de

3rd Emrullah Fatih Yetkin
Kadir Has University
Department of Management Information Systems
Istanbul, Turkiye
fatih.yetkin@khas.edu.tr

4th Reiner Creutzburg
SRH Heidelberg University of Applied Sciences
School of Technology and Architecture
Berlin, Germany
reiner.creutzburg@gmail.com
0000-0001-7522-5990

***Abstract*—This work investigates the role of artificial intelligence (AI) in predictive maintenance for smart meters and photovoltaic (PV) systems through anomaly detection and forecasting. Three datasets were analyzed using unsupervised clustering, Random Forest classification, deep learning (LSTM), and statistical models (SARIMAX, Prophet). The solution achieved near-accurate forecasting (12–13% MAPE for short-term LSTM, 17–18% MAPE for PV energy) and nearly flawless anomaly categorization (ROC-AUC = 0.999; F1 up to 0.87–1.0). Anomalies including energy shortages, peak suppression, and inverter underperformance were translated into actionable maintenance insights through rule-based diagnostics and permutation-based feature importance analysis. These results demonstrate how AI-driven anomaly detection can significantly improve maintenance efficiency, reduce downtime, and enhance grid reliability by shifting from reactive repairs toward proactive, data-driven interventions.**

***Index Terms*—Predictive Maintenance, Anomaly Detection, Smart Meter Data, Time-Series Forecasting, Graph Convolutional Autoencoder, LSTM, CNN–LSTM Autoencoder, Explainable AI, Hyperparameter Optimization, Residual Thresholding**

I. Introduction

Smart meters are advanced digital devices that automatically record and transmit electricity consumption in real time [1], [2]. It is a massive upgrade of the conventional analog meters. The foundation for contemporary smart metering technology was laid in 1972 when Theodore Paraskevakos created a digital metering system that could transmit usage data electronically [3]. Unlike traditional meters, smart meters offer two-way customer-utility communications which allows users to observe usage patterns, avoid wastage, and make more economically viable decisions [1], [4]. For energy providers, smart meters significantly boost system reliability by allowing quick detection of outages and faster restoration by cutting blackout times in Texas by roughly 5.5%. Energy losses during distribution are also decreased by smart meters. The reduction is between 4–7% [5]. ComEd's smart meter installation in Illinois is said to have contained 7.6 million outages, resulting in $1.4 billion in benefits to society [6]. As for cost reductions, industrial users have achieved 7.46% average energy savings, translating to over $41 million annually, while households typically reduce electricity consumption by 3–3.4% and gas usage by 3%, with some studies even reporting savings up to 12% when usage is effectively informed [7]–[9]. Within the European Union, installation costs average €180–200, but savings per meter are higher—around €270 for electricity and €230 for gas—yielding energy savings of 2–10%, and full EU-wide deployment is expected to unlock up to €53 billion in benefits [1], [10]. By detecting unusual activity and theft, smart meters dramatically cut down on non-technical losses—by as much as 30% [11]. This is a key reason they are so vital to modern energy systems: they enable better demand response, make it easier to integrate renewable energy, and promote a grid that is more efficient, resilient, and sustainable.

Predictive maintenance, or PdM, has become a crucial strategy for the smart metering industry in recent years. In order to monitor device health and identify minor variations that precede malfunctions or manipulation, modern meters continuously gather high-frequency data on voltage, current, frequency, and power quality. By moving away from traditional preventive or reactive strategies, which are still common in the energy industry and often disregard real-time operating conditions, PdM reduces unnecessary inspections, avoids

premature replacements, and minimizes costly unplanned shutdowns [12], [13]. However, rather than offering useful triggers for maintenance interventions, a large number of current smart meter anomaly detection techniques continue to concentrate on billing problems, load forecasts, or straightforward outlier elimination [14], [15]. Moreover, while recent deep learning–based models—such as Bi-LSTM autoencoders [2], variational recurrent autoencoders with attention [16], and convolutional autoencoders with adaptive thresholds [17]—have achieved strong detection accuracy, their "black-box" nature often limits transparency and makes operators hesitant to rely on the results [18]. The variability and noise present in smart meter and PV data adds to this difficulty. Normal fluctuations can easily mask context-dependent abnormalities, including inverter underperformance, energy shortages, or phase imbalances, and conventional statistical techniques have trouble identifying such high-dimensional, nonlinear patterns [19], [20]. These shortcomings show the need for an integrated AI-driven framework that reliably identifies anomalies and offers operationally meaningful explanations of their causes, empowering utilities to make informed and timely repair decisions [21], [22].

This work is guided by the following research questions:

1) How can artificial intelligence (AI) be used to detect anomalies in smart meter data as early indicators of faults?
2) Which AI models are most effective for predicting anomalies in smart meter and photovoltaic (PV) datasets?
3) How does shifting from reactive to AI-driven predictive maintenance improve reliability and efficiency?

II. Literature Review

A. Predictive Maintenance in Energy Systems

Recent AI techniques for predictive maintenance surveys highlight hybrid approaches combining statistical methods with machine learning for fault prediction [23], [24], [25]. Zhang et al. [23] emphasize that integrating multiple AI models can improve predictive maintenance performance in manufacturing systems. In energy distribution grids, Mahmoud et al. [24] and Haque et al. [25] outline how IoT sensors and AI algorithms enable condition-based maintenance, though practical deployment is still emerging. Traditional condition-based maintenance concepts date back decades [19], but today's high-resolution data from smart meters and PV plants open new opportunities for real-time analytics.

B. Anomaly Detection Techniques

Anomaly detection has been widely studied across domains [15], [26]. Chandola et al. [15] provide a comprehensive survey of anomaly detection techniques, ranging from statistical methods and clustering to classification and neural networks. Ahmed et al. [26] survey network anomaly detection and similarly categorize methods, including distance-based outlier detection and isolation forests. In the context of power systems, unsupervised methods such as clustering and isolation forests have been popular for their ability to detect novel anomalies without labeled data [19], [27]. For example, Liu et al. [27] introduced the Isolation Forest algorithm, which has since been applied to meter data for outlier detection. Classical time-series models (ARIMA/SARIMA) have also been used for anomaly detection by modeling expected behavior and identifying residuals [24]; however, prior work has noted challenges with ARIMA on complex energy data [12]. More recently, deep learning approaches (e.g., autoencoders, LSTM-based models) have shown success in detecting anomalies in multivariate sensor data [28], [29], but these often act as black boxes. The need for explainable AI (XAI) [18] in anomaly detection is increasingly recognized, allowing maintenance personnel to understand and trust the system's alerts. Arrieta et al. [18] outline XAI concepts relevant to making AI decisions transparent, which is crucial for industrial acceptance. Wellsandt et al. [30] explore a hybrid human-AI approach in maintenance, emphasizing the importance of interpretable "digital assistants" that integrate analytics with expert knowledge.

C. Smart Meter Data Anomalies

Many studies address anomaly detection in advanced metering infrastructure. Jokar et al. [31] detect electricity theft by analyzing consumption patterns, achieving high accuracy in identifying fraudulent usage. Liu and Nielsen [19] developed an online anomaly detection method for smart meter readings using prediction-based techniques, illustrating scalable detection on utility data. Haryadi et al. [32] applied a bidirectional LSTM to detect fraudulent or anomalous consumption patterns. In contrast, Fu et al. [33] proposed an artificial immune system approach to detect energy theft in smart meters (inspired by biological immune responses). Other works focus on data quality issues: for instance, Chen et al. [34] modeled smart meter error and used truncated SVD for anomaly detection, and Farooq et al. [35] addressed detecting data integrity attacks on smart meter measurements. These studies show that a variety of algorithms (SVM, k-NN, neural nets) can identify anomalies in meter data; however, they often flag anomalies without linking to specific maintenance actions.

D. Power Quality and Grid Anomalies

Smart meter data also captures power quality indices and grid events. For example, Patrizi et al. [36] used innovative metering systems to detect power quality anomalies (voltage, frequency deviations) via sensor networks. Lee et al. [37] targeted anomalies in a smart home power management system with electric vehicles and battery storage, highlighting issues like sudden load changes. Standards like IEEE 1159 provide guidelines on monitoring electric power quality [38], which inform threshold-based detections (e.g., flicker severity) in our work. There is ongoing research into using machine learning for grid fault detection and predictive maintenance of grid components [38]–[40]. Rustambekov et al. [39] demonstrated real-time data analysis for predictive maintenance of innovative grid components, and Heiden et al. [38] designed

a digital platform for predictive maintenance in distribution grids, indicating the practical value of such systems.

E. PV System Fault Detection

In PV plants, anomaly detection is crucial for identifying underperforming panels, inverter faults, or degradation. Sepúlveda-Oviedo et al. [41] present a bibliometric review of AI methods for PV system fault diagnosis, showing growing interest in applying AI to PV maintenance. Ledmaoui et al. [42] review advances in predictive maintenance for solar plants, including cyber-security aspects, reflecting the broadening scope of "smart" PV operations. Researchers have applied both data-driven and model-based methods: Mustafa et al. [43] used a deep learning approach for PV fault identification (predicting multiple fault types). In contrast, traditional methods monitor performance ratios and thresholds. Many PV anomaly studies emphasize the use of weather (irradiance) data for context [42], as sudden drops in output may be attributed to environmental factors rather than equipment failures. Our work builds on this by combining weather-normalized expected performance with anomaly detectors. Notably, advanced graph-based models have been explored for capturing the network of components in a PV farm – e.g., Rongali [44] discusses AI-driven DevOps for smart grids and hints at using graph representations. In this study, we implement a Multi-Temporal Graph Convolutional Autoencoder (MTGCAE) to capture the relationships among inverters and transformers in the PV system, enabling detection of system-level anomalies that involve multiple components.

Prior literature provides a rich toolkit of AI algorithms for anomaly detection and predictive maintenance. However, gaps remain in integrating these methods into a coherent framework that produces explainable maintenance insights. Our work contributes an end-to-end pipeline that fuses unsupervised and supervised learning with domain-based rules, tailored to the multi-scale nature of smart meter and PV data. This approach addresses the identified gaps by providing both high detection performance and interpretability, thereby facilitating a proactive maintenance strategy.

III. Methodology

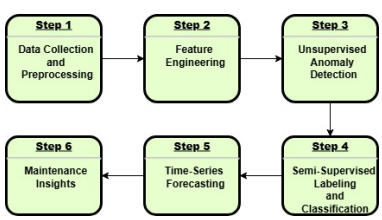


Fig. 1. Methodology pipeline.

We designed a unified six-stage pipeline for anomaly detection and predictive maintenance, applied across three heterogeneous datasets. This pipeline ensures consistency, interpretability, and adaptability to both short-term smart meter data and long-term PV plant monitoring.

A. Data Collection and Preprocessing

Three datasets were used: (i) a short-term three-phase smart meter log, (ii) a one-week distribution quality dataset, and (iii) a five-year PV plant dataset with inverter and irradiance data. All datasets were standardized, cleaned, and aligned to a uniform time grid. Missing values were interpolated where possible, and sensor resets or invalid readings were corrected.

B. Feature Engineering and Exploratory Analysis

Domain-specific features such as daily energy increments, power factors, inverter contribution ratios, and irradiation-yield ratios were engineered to summarize daily behavior. Correlation heatmaps were used to identify systematic underperformance (e.g., inverter Eg10 in the PV dataset).

C. Unsupervised Anomaly Detection

Unsupervised clustering (k-means, Isolation Forest, LOF) combined with expert rule checks was used to identify anomalies without labels. This allowed separation of typical vs. abnormal operational regimes, such as suppressed peaks, energy shortfalls, or inverter failures.

D. Semi-Supervised Labeling and Classification

Pseudo-labels from the unsupervised stage were used to train supervised classifiers (Random Forest). This achieved ROC-AUC $\approx$ 0.999 and F1-scores up to 1.0, demonstrating clear separation between normal and anomalous operation.

E. Forecasting and Residual Analysis

Classical and deep learning forecasting methods (SARIMAX, Prophet, LSTM) were applied to detect rapid deviations via residual thresholds. Forecasting achieved MAPE values of 12–13% for load and 17–18% for PV energy, sufficient to flag deviations beyond forecast error margins.

F. Maintenance Insights and Prescriptive Actions

Detected anomalies were translated into actionable alerts such as "Inverter Eg10 underperformance," "Peak suppression," or "Phase imbalance." These insights were linked to prescriptive actions (e.g., inverter replacement, capacitor inspection), making the framework directly usable for operators.

IV. Findings and Discussion

This section presents the empirical outcomes of applying the six-stage predictive maintenance pipeline. The discussion integrates four key aspects: (i) anomaly detection and forecasting performance, (ii) interpretability and model behavior, (iii) dataset-specific observations, and (iv) prescriptive maintenance and deployment insights.

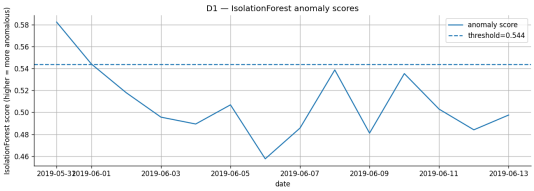


Fig. 2. Isolation Forest anomaly score timeline for smart meter dataset (D1). A single-day anomaly is clearly visible.

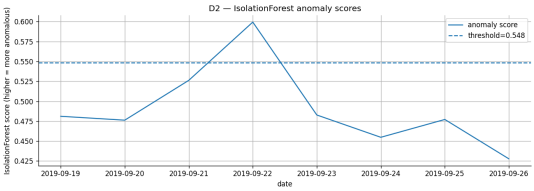


Fig. 3. D2 Isolation–Forest anomaly scores with threshold.

A. Anomaly Detection and Forecasting Performance

The integrated pipeline achieved a good performance in identifying abnormal operating patterns across the three datasets. Isolation Forest anomaly timelines (Figures 2, 3, and 4) consistently isolated distinct deviations from normal regimes.

For Dataset 1 (smart meter), the Isolation Forest detected a single low-energy day, while in Dataset 2 (distribution feeder), a suppressed peak on 22–09–2019 was isolated (Figure 5). In the PV dataset (D3), anomaly bursts in early 2015 and mid-2018 aligned with known inverter malfunctions.

Semi-supervised Random Forest classification confirmed these unsupervised findings. The model achieved perfect separation in D1 (Accuracy = 1.00, F1 = 1.00) and near-perfect performance on the PV dataset (ROC-AUC ≈ 0.999, Recall = 1.00), as shown in Figure 6.

Forecasting-based residual analysis added predictive capability. In Dataset 1, one-hour-ahead LSTM forecasting achieved MAPE $= 12.7\%$, while in the PV dataset, SARIMAX and Prophet reached MAPE $= 17$–18% with $R^2 > 0.90$. Figure 7 shows close alignment between predicted and actual

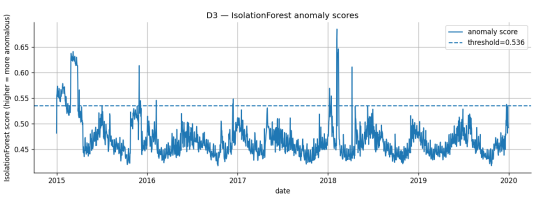


Fig. 4. Isolation Forest anomaly scores across five years of PV data (D3). Recurring anomaly bursts indicate inverter underperformance.

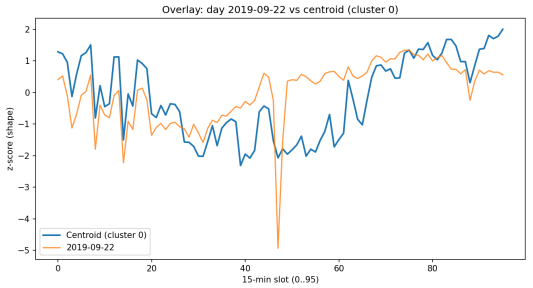


Fig. 5. Load-shape overlay for the anomalous day (22–09–2019) in the distribution feeder dataset (D2). Suppressed peaks and lower PF confirm abnormal operation.

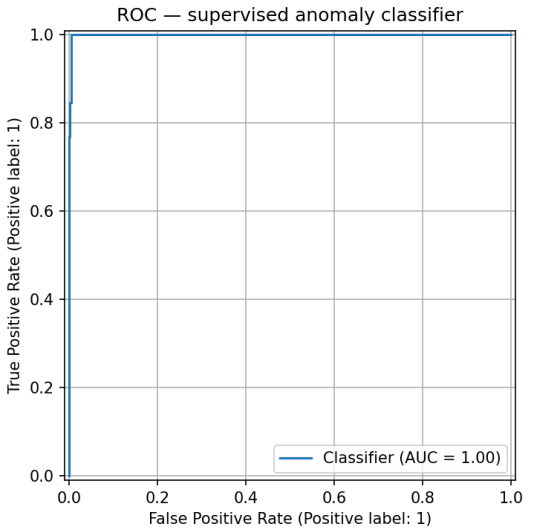


Fig. 6. ROC curve for the Random Forest classifier (D3), showing near-perfect anomaly separability.

PV generation, with residual bursts coinciding with inverter underperformance.

B. Model Behavior and Interpretability

Interpretability was a central design goal. Feature importance and SHAP analysis provided explainable insight into model decisions. For Dataset 2, permutation feature importance identified `PF_sum` and `KW_max` as the main anomaly drivers. For Dataset 3, global SHAP summaries (Figure 8) highlighted inverter contribution ratios and PV-to-irradiance efficiency as dominant explanatory factors—consistent with the observed underperformance of inverter Eg10.

These interpretable diagnostics increase operator confidence, allowing domain experts to verify model outputs physically. The consistent alignment between learned importance

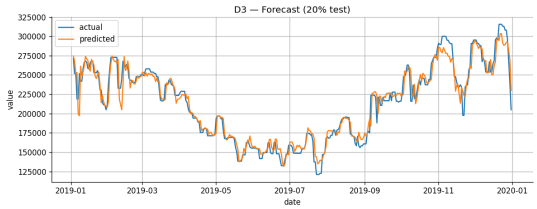


Fig. 7. Forecast vs. actual PV daily energy (D3). Residual bursts coincide with inverter failures.

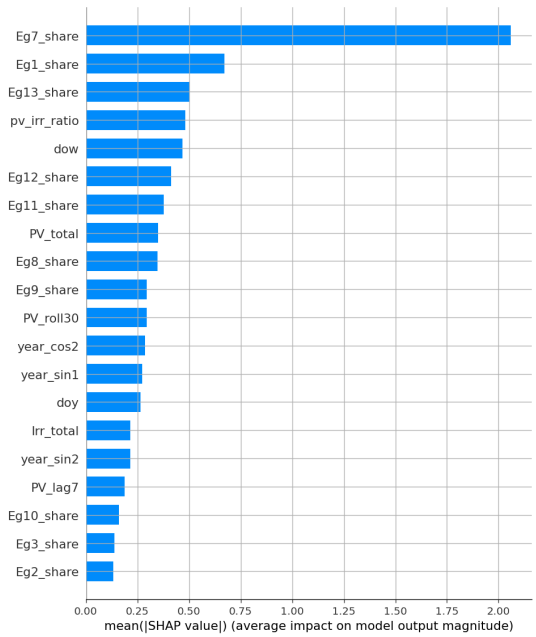


Fig. 8. Global SHAP feature importance for the PV dataset (D3). Inverter shares and efficiency ratios dominate anomaly explanations.

rankings and engineering indicators supports trust and adoption.

C. Dataset-Specific Observations

Smart Meter (D1): Strong daytime load cycles with a single under-consumption anomaly confirmed pipeline sensitivity to short-term deviations.

Distribution Feeder (D2): The anomalous day exhibited suppressed active power and low power factor (Figure 5), both physically consistent with feeder-level imbalance or partial load disconnection.

PV Plant (D3): Seasonal PV energy patterns (Figure 9) matched irradiance trends, while ensemble detection revealed persistent underperformance of inverter Eg10 (Figure 10), validating the system's long-term anomaly sensitivity.

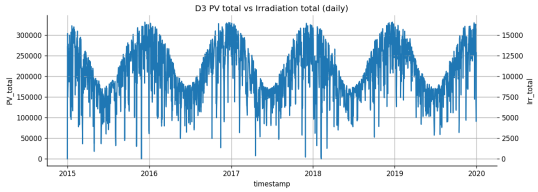


Fig. 9. Daily PV generation and irradiation (D3). Strong seasonal correlation confirms data reliability.

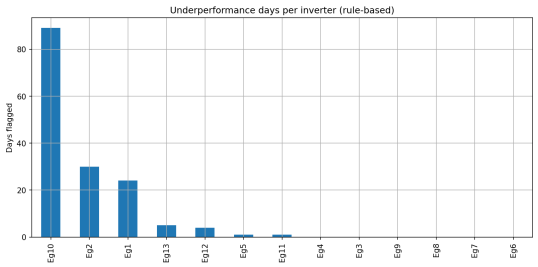


Fig. 10. Underperformance counts per inverter (D3). Inverter Eg10 consistently shows dominant anomaly frequency.

D. Prescriptive Maintenance and Deployment Insights

The framework transformed detected anomalies into actionable maintenance insights. In D1, low daily energy flagged possible phase imbalance; in D2, low PF and suppressed peaks suggested capacitor or load inspection; in D3, inverter Eg10's chronic underperformance indicated replacement priority. These insights show that anomaly detection outputs can directly guide maintenance scheduling and inventory planning.

From a deployment standpoint, practical implementation requires addressing (i) **data quality**—continuous validation to avoid false alarms from resets or missing logs, (ii) **scalability**—parallel processing for millions of meters, and (iii) **operator adoption**—embedding interpretable outputs within SCADA dashboards. As shown by the interpretability layer (SHAP, feature importance), explainable outputs are essential for operational trust.

The six-stage pipeline achieved robust, explainable, and prescriptive anomaly detection across heterogeneous energy datasets. Its high performance (ROC-AUC $\approx$ 0.999, MAPE 12–18%) and clear interpretability bridge the gap between advanced analytics and field-level maintenance. The results shows the framework's readiness for pilot-scale deployment, providing a data-driven foundation for predictive maintenance in real-world smart grid environments.

V. Deployment Challenges and Practical Considerations

While the proposed framework achieves very good anomaly detection metrics and interpretable outputs, its practical de-

ployment in utility environments requires addressing several challenges:

Data Quality. Smart meter and PV data streams are often incomplete, noisy, or subject to resets. Our preprocessing steps addressed missingness and counter rollovers, but real-world systems must implement continuous monitoring for logging gaps, calibration errors, and false alarms.

Scalability. The current pipeline was validated on three research datasets. At the utility scale—millions of meters or multiple PV farms—model training and inference will demand distributed computing and efficient data pipelines. Lightweight statistical models (e.g., SARIMAX, Prophet) may be preferable in early deployments, while deep models can be introduced for larger, high-resolution deployments.

Operator Adoption. High detection accuracy alone does not guarantee adoption. Utilities require interpretable results that align with engineering knowledge. Our use of permutation importance, SHAP-based explanations, and rule-based diagnostics provides transparency, but successful adoption will depend on training operators and embedding alerts into existing SCADA dashboards.

System Integration. Utility IT/OT environments are complex. For real-world use, the pipeline must integrate seamlessly with SCADA, AMI, or existing maintenance management systems. This requires attention to interoperability standards and cybersecurity safeguards.

VI. Conclusion

This paper presented an integrated, explainable, and prescriptive AI pipeline for anomaly detection and predictive maintenance in smart meter and photovoltaic (PV) systems. The proposed six-stage framework combined unsupervised, semi-supervised, and forecasting-based methods to provide both detection and diagnosis capabilities. By integrating Isolation Forests, clustering, Random Forest classification, and SARIMAX/Prophet forecasting, the system achieved near-perfect anomaly separation (ROC-AUC $\approx$ 0.999) and forecasting accuracy (MAPE 12–18%). Beyond quantitative performance, the pipeline demonstrated how hybrid data-driven analytics can enhance operational decision-making in energy infrastructures.

A key contribution lies in bridging the gap between detection and maintenance action. The system did not stop at flagging anomalies but linked them to interpretable, domain-specific causes—such as inverter degradation, load imbalance, or low power factor—and generated prescriptive maintenance suggestions. This integration of expert knowledge and explainable AI methods (e.g., SHAP and rule-based diagnostics) transforms anomaly detection into a decision-support process that utilities can realistically deploy. The modular architecture also allows future inclusion of supervised learning models as more labeled data become available, ensuring the framework remains adaptable as datasets grow.

The study underscores that effective predictive maintenance in the energy domain requires not only high-performing models but also interpretability, scalability, and trust. Future work will therefore focus on real-world deployment of the proposed system within live utility environments, addressing issues of data quality, streaming scalability, and cybersecurity integration. By embedding transparency and prescriptive intelligence into anomaly detection, this research lays a foundation for intelligent, operator-assisted maintenance systems in modern energy networks.

VII. Future Work

While this study has demonstrated the potential of AI-driven predictive maintenance in smart meters and photovoltaic (PV) systems, several directions remain open for future exploration. A primary step involves expanding the datasets to include longer time spans and confirmed ground-truth fault records. Larger datasets would allow more rigorous validation and the use of advanced models such as Bidirectional LSTMs or Graph Convolutional Autoencoders. Incorporating real incident logs would also enable quantifying lead time before failures, a key metric for assessing true predictive maintenance value.

Another promising direction is the integration of external contextual data. For PV systems, weather variables such as irradiance forecasts, temperature, and cloud cover could improve short-term forecasting accuracy. Similarly, incorporating occupancy, industrial schedules, or environmental data into smart meter analysis would help reduce false positives by distinguishing operational anomalies from natural consumption variability. Enriching the data context will allow models to refine thresholds and enhance reliability.

From a deployment perspective, future work should focus on achieving real-time, scalable operation. This includes optimizing models for streaming data and exploring lightweight implementations suitable for edge devices and embedded hardware. At the same time, extending the framework to jointly detect technical faults and cyber anomalies—such as false data injection or malicious switching—represents a critical evolution toward resilient smart grid monitoring. Combining operational data with cybersecurity signals or blockchain-based integrity checks could unify fault and intrusion detection into a single trusted system. Finally, linking anomaly detection directly to decision-support layers that prioritize maintenance based on risk or expected energy loss would transform predictive maintenance from reactive monitoring to proactive grid optimization.

References

[1] W. contributors, "Smart meter — wikipedia, the free encyclopedia," 2025, online, accessed 31-August-2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Smart_meter&oldid=1305055211

[2] S. Lee, H. Jin, S. H. Nengroo, Y. Doh, C. Lee, T. Heo, and D. Har, "Smart metering system capable of anomaly detection by bi-directional lstm autoencoder," in *2022 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2022, pp. 1–6.

[3] W. contributors, "Automatic meter reading — wikipedia, the free encyclopedia," 2025, online; accessed 31-August-2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Automatic_meter_reading&oldid=XXXXX

[4] E. Commission, "Smart grids and meters — energy — european commission," 2025, [Online; accessed 31-August-2025]. [Online]. Available: https://energy.ec.europa.eu/topics/markets-and-consumers/smart-grids-and-meters_en

[5] M. S. S. of Management. (2023) Smart meters generate revenue, improve efficiency for public utilities. Accessed: 2025-08-27. [Online]. Available: https://mitsloan.mit.edu/ideas-made-to-matter/smart-meters-generate-revenue-improve-efficiency-public-utilities

[6] Wikipedia contributors, "Commonwealth edison — Wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/Commonwealth_Edison, 2025, accessed: 2025-08-27.

[7] I. U. Kelley School of Business. (2024) Research shows smart utility meters drive down manufacturing costs if data is used to drive operations. Accessed: 2025-08-27. [Online]. Available: https://blog.kelley.indianapolis.iu.edu/2024/04/03/research-shows-smart-utility-meters-drive-down-manufacturing-costs-if-data-is-used-to-drive-operations/

[8] B. I. Team, "Do smart meters reduce households' energy consumption?" 2023, [Online; accessed 31-August-2025]. [Online]. Available: https://www.bi.team/blogs/do-smart-meters-reduce-households-energy-consumption/

[9] P. A. Bao Tran, "Iot energy savings: Smart metering & consumption reduction data," 2025, [Online; accessed 31-August-2025]. [Online]. Available: https://patentpc.com/blog/iot-energy-savings-smart-metering-consumption-reduction-data

[10] A. Faruqui, D. Harris, and R. Hledik, "Unlocking the €53 billion savings from smart meters in the eu: How increasing the adoption of dynamic tariffs could make or break the eu's smart grid investment," *Energy Policy*, vol. 38, no. 10, pp. 6222–6231, 2010, the socio-economic transition towards a hydrogen economy - findings from European research, with regular papers. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421510004738

[11] T. Pangarkar, "Smart meter market boom expected to surpass usd 63.9 billion by 2033," January 2025, [Online; accessed 31-August-2025]. [Online]. Available: https://www.news.market.us/smart-meter-market-news/

[12] Z. Yang, X. Chen, D. Gao, G. Cheng, and R. Wang, "A fast detection method for metering anomalies of three-phase energy meters based on sliding filter and decision tree," *Electric Power Systems Research*, vol. 238, p. 111056, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779624009416

[13] N. Dr.Sujaudeen, D. L. Priya, G. Venkatesan, and M. Dharshni, "Time series forecasting analysis for automated smart meter reading system," *Indian Journal of Energy and Energy Resources*, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:278951159

[14] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327005001512

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009. [Online]. Available: https://doi.org/10.1145/1541880.1541882

[16] W. Dai, X. Liu, A. Heller, and P. S. Nielsen, "Smart meter data anomaly detection using variational recurrent autoencoders with attention," in *Intelligent Technologies and Applications*, F. Sanfilippo, O.-C. Granmo, S. Y. Yayilgan, and I. S. Bajwa, Eds. Cham: Springer International Publishing, 2022, pp. 311–324.

[17] S. Maitra, S. Kundu, and A. Shankar, "A real-time anomaly detection using convolutional autoencoder with dynamic threshold," *arXiv preprint arXiv:2404.04311*, 2024.

[18] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[19] X. Liu and P. S. Nielsen, "Scalable prediction-based online anomaly detection for smart meter data," *Information Systems*, vol. 77, pp. 34–47, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437917303216

[20] L. Yu, X. Zhang, L. Du, and L. Yue, "Anomaly detection of cyber attacks in smart grid communications based on residual recurrent neural networks," *Security and Privacy*, vol. 8, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:275532215

[21] Z. Sida, M. Zhu, and L. Ying, "Research on anomaly detection and correction of power metering data based on machine learning algorithm," *Science and Technology for Energy Transition*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:274266479

[22] X. Dong, Z. Jing, Y. Dai, P. Wang, and Z. Chen, "Failure prediction and replacement strategies for smart electricity meters based on field failure observation," *Sensors*, vol. 22, no. 24, p. 9804, 2022.

[23] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, 2011.

[24] M. A. Mahmoud, N. R. Md Nasir, M. Gurunathan, P. Raj, and S. A. Mostafa, "The current state of the art in research on predictive maintenance in smart grid distribution network: Fault's types, causes, and prediction methods—a systematic review," *Energies*, vol. 14, no. 16, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/16/5078

[25] R. Haque, A. Bajwa, N. alam Siddiqui, and I. Ahmed, "Predictive maintenance in industrial automation: A systematic review of iot sensor technologies and ai algorithms," *American Journal of Interdisciplinary Studies*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:277803122

[26] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804515002891

[27] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[28] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," 2016. [Online]. Available: https://arxiv.org/abs/1607.00148

[29] F. Tao, Q. Qi, L. Wang, and A. Nee, "Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison," *Engineering*, vol. 5, no. 4, pp. 653–661, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S209580991830612X

[30] S. Wellsandt, K. Klein, K. Hribernik, M. Lewandowski, A. Bousdekis, G. Mentzas, and K.-D. Thoben, "Hybrid-augmented intelligence in predictive maintenance with digital intelligent assistants," *Annual Reviews in Control*, vol. 53, pp. 382–390, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578822000165

[31] P. Jokar, N. Arianpoo, and V. C. M. Leung, "Electricity theft detection in ami using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, 2016.

[32] A. Haryadi, S. A. I. Alfarozi, and A. R. Pratama, "Fraud and anomaly detection in electricity usage patterns using bilstm," in *2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS)*, 2025, pp. 1–6.

[33] J. Fu, C. Yang, Y. Liu, K. Zhang, J. Li, and B. Li, "Artificial immunity-based energy theft detection for advanced metering infrastructures," *International Journal of Critical Infrastructure Protection*, vol. 48, p. 100739, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874548225000010

[34] L. Chen, K.-W. Lao, Y. Ma, and Z. Zhang, "Error modeling and anomaly detection of smart electricity meter using tsvd+l method," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022.

[35] A. Farooq, K. Shahid, and R. L. Olsen, "Securing the green grid: A data anomaly detection method for mitigating cyberattacks on smart meter measurements," *International Journal of Critical Infrastructure Protection*, vol. 46, p. 100694, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874548224000350

[36] G. Patrizi, C. Garzon Alfonso, L. Calandroni, A. Bartolini, C. Iturrino Garcia, L. Paolucci, F. Grasso, and L. Ciani, "Anomaly detection for power quality analysis using smart metering systems," *Sensors*, vol. 24, no. 17, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/17/5807

[37] S. Lee, S. H. Nengroo, H. Jin, Y. Doh, C. Lee, T. Heo, and D. Har, "Anomaly detection of smart metering system for power management with battery storage system/electric vehicle," *ETRI Journal*, vol. 45, no. 4, pp. 650–665, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.2022-0135

[38] P. z. Heiden, J. Priefer, and D. Beverungen, "Predictive maintenance on the energy distribution grid–design and evaluation of a digital industrial platform in the context of a smart service system," *IEEE Transactions on Engineering Management*, vol. 71, pp. 3641–3655, 2024.

[39] I. Rustambekov, G. S. Saidakhmedovich, B. Abduvaliyev, E. Kan, I. Abdukhakimov, M. Yakubova, and D. Karimov, "Predictive maintenance of smart grid components based on real-time data analysis," in *2024 6th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, 2024, pp. 949–952.

[40] H. Raju, Shubhra, A. Nagpal, J. Nagaraju, T. Al-Rubaye, and P. Tewari, "Advancing predictive maintenance in smart grids with machine learning techniques through comparative analysis of svm and cnn models," in *2025 International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI)*, 2025, pp. 1251–1256.

[41] E. H. Sepúlveda-Oviedo, L. Travé-Massuyès, A. Subias, M. Pavlov, and C. Alonso, "Fault diagnosis of photovoltaic systems using artificial intelligence: A bibliometric approach," *Heliyon*, vol. 9, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264551176

[42] Y. Ledmaoui, A. El Maghraoui, M. El Aroussi, and R. Saadane, "Review of recent advances in predictive maintenance and cybersecurity for solar plants," *Sensors*, vol. 25, no. 1, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/25/1/206

[43] Z. Mustafa, A. S. Awad, M. Azzouz, and A. Azab, "Fault identification for photovoltaic systems using a multi-output deep learning approach," *Expert Systems with Applications*, vol. 211, p. 118551, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422016207

[44] L. P. Rongali, "Utilizing ai-driven devops for predictive maintenance and anomaly detection in smart grids," *SSRN Electronic Journal*, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:278398376

Toward a Modular Evaluation Framework for Lightweight AEAD Ciphers

Ikechukwu John Chukwu*, Vesna Dimitrova†, Tuğçe Ballı‡
*cikechukwjohn@stu.khas.edu.tr
†Ss. Cyril and Methodius University, Skopje, North Macedonia
Kadir Has Üniversitesi, Istanbul, Türkiye

***Abstract*—Lightweight cryptography is essential for securing resource-constrained devices and continues to evolve rapidly. In this work, we propose a modular and extensible evaluation framework that benchmarks lightweight ciphers using three critical software performance metrics: code size, execution time, and RAM consumption. To complement standard benchmarking, we conducted nonce misuse experiments, highlighting an often-overlooked dimension of cryptographic evaluation. Our results show that ASCON achieves the most compact code size, while AES-GCM and ChaCha20-Poly1305, despite their ubiquity in real-world protocols, exhibit higher resource demands and weaker robustness under misuse.**

***Index Terms*—Lightweight Cryptography, AEAD, Benchmarking Framework, NIST LWC, Nonce Misuse**

I. Introduction

One of the earliest contributions in this domain is the PRESENT block cipher [1], supposedly the genesis of lightweight cryptography. It was introduced in 2007 to secure RFID tags and embedded sensors. Since its inception, it has played a critical role in information security and is famous in many microcontrollers. However, while influential, PRESENT may be unable to keep pace with evolving requirements and adversarial schemes on small devices, given the rise of quantum computation and advances in technology. Alongside PRESENT are also alternatives such as optimized AES variants (e.g., TinyAES) adopted as lightweight substitutes [2], but some of these optimization efforts often yielded suboptimal performance in ultra-constrained environments or incurred security trade-offs due to aggressive optimizations [3]. Such limitations underscored the need for a unified and standardized approach. In response, the National Institute of Standards and Technology (NIST) initiated its Lightweight Cryptography competition in 2013, focusing on Authenticated Encryption with Associated Data (AEAD) primitives tailored for constrained environments [4]. The remainder of this paper is structured as follows. Section II reviews the background of cryptology, lightweight ciphers, and evaluation metrics. Section III surveys related frameworks such as FELICS and its AEAD extensions. Section IV introduces our proposed modular benchmarking framework, followed by Section V, which presents results and interpretation of both performance and nonce misuse experiments. Finally, Section VI concludes with insights on trade-offs and implications for future cryptographic evaluation.

II. Background

A. Cryptology

Cryptology aims to study the art and science of privacy or secrecy, with three subcategories: Cryptography, Cryptanalysis, and protocols.

B. Cryptography

Cryptography studies and invents techniques that secure digital communications. At its core lies the concept of a secret to achieve this security. Furthermore, the science is subcategorized into symmetric, asymmetric, and hybrid.

1) Symmetric cryptography: Here, one secret key K as in Figure 1 enables the exchange of data between communicating parties. This key serves both for encryption and decryption, as in Equations 1, making it computationally efficient.

$$C = E_K(P), \quad P = D_K(C) \tag{1}$$

where P is the plaintext, C is the ciphertext, and E_K/D_K represent encryption and decryption functions under the key K.

2) Asymmetric (Public-Key) cryptography: Two mathematically related keys: a public key K_{pub} (known to everyone) and a private key K_{priv} (kept secret) are used to achieve security. With K_{pub} for encryption and K_{priv} for decryption as in Equation 2.

$$C = E_{K_{pub}}(P), \quad P = D_{K_{priv}}(C) \tag{2}$$

3) Hybrid cryptography: A combination of symmetric and asymmetric encryption, as in Equation 3

$$K_{session} = D_{K_{priv}}(E_{K_{pub}}(K_{session})), \quad C = E_{K_{session}}(P) \tag{3}$$

C. Cryptanalysis

A science that studies and investigates weaknesses in cipher design.

D. Lightweight ciphers

Algorithms that by design require little computational power are considered lightweight. They are mostly relevant in and not limited to IoT sensors, RFID tags, and embedded controllers.

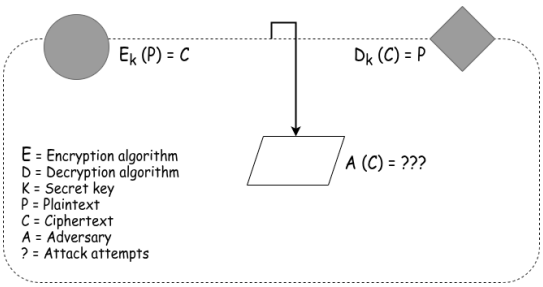


Fig. 1: The diagram illustrates symmetric cryptography on a communication line in the presence of an adversary.

E. Cryptographic evaluation metrics

In cryptography, some evaluation metrics include:

1) **Throughput (T)** reflects the amount of data a cipher can process within a given time, calculated by dividing the size of the processed data by the time taken to process it, often measured in bits or bytes per second:

$$T = \frac{\text{Data Size (bits or bytes)}}{\text{Execution Time (seconds)}}$$

2) **Latency (L)** measures the delay to complete a single cryptographic operation, such as encrypting or decrypting a block of data. It is often derived from the number of CPU cycles required, divided by the processor frequency:

$$L = \frac{\text{Execution Cycles}}{\text{CPU Frequency (Hz)}}$$

3) **Memory Footprint (M)** captures the total memory resources the cipher requires during execution, comprising both code size (ROM) and runtime memory (RAM):

$$M_{\text{total}} = M_{\text{code}} + M_{\text{RAM}}$$

Where M_{code} is the storage size of the compiled cipher code, and M_{RAM} includes the space needed for intermediate computations, stack, and heap usage.

4) **Energy Consumption (E)** estimates the total energy used to perform a cryptographic operation, important for battery-powered and energy-harvesting devices.

$$E = N_{\text{cycles}} \times E_{\text{cycle}} \quad \text{or} \quad E = P \times \text{Execution Time}$$

Where N_{cycles} is the number of CPU cycles used, E_{cycle} is the energy consumed per cycle (obtained from hardware specifications), and P is the average power consumption during the operation.

5) **Correctness** ensures that the cipher implementation behaves as specified. It is tested using Known Answer Tests (KATs). Formally:

$$C = \begin{cases} 1 & \text{if } \texttt{output matches text vector} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In many frameworks, correctness is the precondition before reporting performance metrics.

6) **Nonce misuse resilience (NMR)** captures the impact of repeated nonces on ciphertext integrity since AEAD ciphers fail catastrophically under nonce reuse, this metric. For instance, you can measure the bit-matching probability of two ciphertexts under the same nonce:

$$NMR = \frac{\text{Matching Bits}}{\text{Total Bits}}$$

where a high percentage (close to 100%) indicates poor resilience (leakage), and values near 50% indicate random-like behavior. This directly ties into our experiments with OpenSSL and Ascon.

III. Related Work

We begin with the Fair Evaluation of Lightweight Cryptographic Systems (FELICS).

A. FELICS: Fair Evaluation of Lightweight Cryptographic Systems

The FELICS methodology is centered on several key principles, including adhering to a standardized API specific to the primitive type (block cipher, stream cipher, hash function, and later AEAD schemes via FELICS-AEAD; an extended version for the framework), and basing the framework on the C programming language. Furthermore, FELICS measures code size (ROM footprint) measured from compiled object files by adding up the `text` and `data` sections using the `size` tool, which is part of the GNU toolchain, the RAM usage, differentiating between static/global RAM and peak stack RAM usage, stack RAM measurement often relies on compiler-specific static analysis or estimation, and the throughput, primarily reported in clock cycles for specific operations and as cycles per byte (CPB) for throughput. These metrics were reported through the Figure of Merit (FoM).

FELICS evaluations were computed on three distinct microcontroller architectures representing various resource constraints. They included:

- An 8-bit AVR microcontroller (specifically, the ATmega128P).
- A 16-bit MSP430 microcontroller (the MSP430F1611).
- A 32-bit ARM Cortex-M3 microcontroller.

B. FELICS-AEAD: Extending Evaluation to Authenticated Encryption

FELICS-AEAD [5] The key performance metrics targeted in FELICS-AEAD remained consistent with the original FELICS framework, focusing on code size (ROM footprint), RAM usage (static and stack), and execution speed (clock cycles and cycles per byte) for encryption and decryption operations. The target devices utilized for FELICS-AEAD evaluations typically included the same range of microcontrollers as the broader FELICS project. Furthermore, a variety of prominent AEAD ciphers were implemented and benchmarked within the FELICS-AEAD project, including CAESAR competition finalists and early NIST LWC candidates such as NORX,

ACORN, Ketje Jr, ASCON, and AES-GCM (as a baseline). In the next section, we report another framework variation for AEAD ciphers.

C. FELICS-AE: Modified FELICS for Evaluating Lilliput-AE

Drawing its design rationale from FELICS, FELICS-AE built into its framework, AEAD-specific benchmarking, such as `felics-aead`-run for executing test vectors through the AEAD API, `felics-aead-measure` for performance data collection, `felics-publish` for formatting results, and `felics-compare` for side-by-side analysis of different implementations or parameters. These tools automate testing various messages and associated data lengths, which is crucial for comprehensive AEAD evaluation.

The key performance metrics targeted by FELICS-AE were consistent with those prioritized by the general FELICS framework, and the evaluations using this adapted framework were conducted on microcontroller platforms representative of LWC targets like FELICS, such as the 8-bit AVR, the 16-bit MSP430F1611, and ARM.

IV. The Proposed Framework

In the following subsections, we describe in detail the modules that constitute the framework.

Cipher Interface (`cipher_interface.h`)

The first component of our modular architecture is the cipher interface, which defines how each AEAD lightweight cipher is integrated into the framework. We adopted its design from the structural requirements of the eBACS benchmarking framework for lightweight ciphers. In eBACS, each cipher's `ref` directory includes an `encrypt.c` file that implements AEAD encryption and decryption functions. Inspired by this approach, we created a uniform structure that encapsulates the essential elements of every cipher, ensuring compatibility and ease of integration. Our `cipher_interface.h` file provides type definitions for AEAD encryption and decryption functions, mirroring the NIST submission API specification for lightweight cryptography, namely `crypto_aead_encrypt` and `crypto_aead_decrypt`. These standardized definitions not only promote interoperability but also ensure that any cipher submitted under NIST LWC requirements can be integrated without modification.

To achieve modularity, we define a `struct` that captures the key properties of each cipher:

- A string pointer to the cipher's name (e.g., `"ASCON128"`).
- Three `size_t` fields specifying the lengths of the key, nonce, and authentication tag.
- Function pointers for AEAD encryption and decryption routines.

Platform abstraction layer (`pal.h`)

The header file defines timer utilities as well as functions to measure code segment sizes, specifically the `text`, `data`, and `bss` sections. By extracting the sizes of these sections from the compiled binary, using platform-specific tooling (such as the `size` utility on UNIX-like systems), we were able to compute two critical metrics: **code size**, ie, the size of the `text` section, and **static RAM size**, which is the sum of the `data` and `bss` segments. We have also included tools, as in Table I, that other frameworks explored to extract metric values from AVR microcontrollers [6] and from MSP430 microcontrollers [7]. These devices are a good representation of a variety of embedded applications.

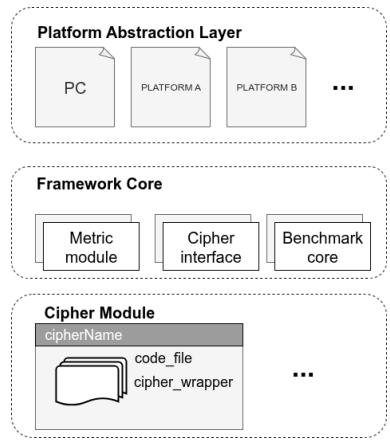


Fig. 2: Schematic diagram of the framework architecture

Benchmark core (`benchmark_core.h/c`)

We conducted a series of experiments. The first experiment we explored with this component integrates all other modules and constitutes the main logic responsible for executing benchmarks on each cipher. It handles timing measurements, correctness checks using Known Answer Test (KAT) vectors, and metric collection. Within this module, we defined a `run_benchmark()` function that performs four key operations: KAT validation, benchmarking, metrics collection, and result output.

To initialize the benchmarking process, we define four static character arrays to represent the required input fields of a typical lightweight AEAD cipher: the test key, test nonce, associated data, and public message. Each is initially set to 16 bytes of zero values, which are later replaced when KATs are integrated. It is also important to note that these test vectors in the KAT file were written and retrieved directly from Ascon's developers' GitHub repository. We defined the macro `KAT_FILE` to point to the test vector file, and another macro `MAX_VEC_SIZE` to determine the maximum vector size (set to 128). During validation, the function loads these answer test vectors, applies the cipher to each input, and compares the output to the expected ciphertext to confirm correctness.

Platform	Code Size Tool	RAM Tool	Execution Time Tool
PC	`size` (GNU binutils)	`size` (bss+data)	High-resolution timer (`clock_gettime`)
AVR	`avr-size`	`avr-size` (bss+data)	Timer/Counter1 + UART
MSP430	`msp430-size`	`msp430-size` (bss+data)	Timer_A + UART

TABLE I: Tools and methods used to extract code size, RAM usage, and execution time for each target platform.

A. Integration with OpenSSL

To further strengthen the practical value of our framework, we integrated `OpenSSL` as a backend. OpenSSL is one of the most widely deployed cryptographic libraries, supporting protocols such as TLS, QUIC, IPsec, and VPNs. By using OpenSSL's EVP API, which already aligns with our abstract AEAD interface, we could benchmark AES-GCM and ChaCha20-Poly1305 under identical conditions to lightweight finalists such as ASCON.

B. Nonce misuse experiments

Nonce means a "number used once," and AEAD security proofs assume strict nonce uniqueness. Here, we describe the setup of our misuse experiments (results are analyzed later in Section 5). Reuse of a key–nonce pair results in catastrophic leakage. For instance, in AES-GCM, $C_1 \oplus C_2 = P_1 \oplus P_2$, which reveals the XOR of two plaintexts directly.

To explore this, we extended our framework to deliberately fix the nonce and encrypt random plaintexts under the same key–nonce pair. We compared the resulting ciphertext XOR, $C_1 \oplus C_2$, against the plaintext XOR, $P_1 \oplus P_2$, computing the percentage of matching bits. Experiments were repeated 2000 times for multiple plaintext lengths (16, 64, 512, 1500, 4096 bytes).

TABLE II: Nonce reuse experiment results (2000 trials per size)

Cipher	16 bytes	512 bytes	4096 bytes
AES-GCM (OpenSSL)	100.0%	100.0%	100.0%
ChaCha20-Poly1305 (OpenSSL)	100.0%	100.0%	100.0%
ASCON-128 (ref)	74.9%	50.8%	50.1%

The results clearly show the catastrophic fragility of AES-GCM and ChaCha20-Poly1305 under nonce reuse, with perfect leakage across all sizes. In contrast, ASCON's ciphertext differences approximate random noise, clustering around 50% bit matches. While integrity is still invalidated, confidentiality is not linearly exposed. This contrast highlights the importance of misuse evaluation in frameworks like ours.

V. Results and Interpretation

The results from our study present a multi–dimensional view of how different AEAD ciphers perform in a software setting. For clarity, we divide our findings into two main categories: performance metrics (code size, RAM usage, encryption/decryption speed) and misuse resilience.

A. Performance metrics

We first examined the efficiency of the selected lightweight ciphers under normal usage. The key metrics collected were:

- **Code size**: non–volatile memory footprint (flash), taken from the `.text` section.
- **RAM usage**: runtime memory, i.e., the sum of `.data` and `.bss`.
- **Encryption/Decryption time**: measured in nanoseconds using high–resolution timers, reported for both single and averaged iterations.

From Table III, we note that **ASCON** has the smallest code size and balanced performance, which makes it attractive for constrained devices. **GIFT-COFB** achieves very fast decryption times, though at the cost of a larger code footprint. Meanwhile, **Elephant**, though relatively compact in memory usage, performs worst in both encryption and decryption times, limiting its utility in time–critical systems.

Figure 3 provides a combined visual comparison across all metrics, making it clearer where each cipher excels or falls short. For example, ASCON stands out when code size is the priority, whereas GIFT-COFB dominates in raw decryption speed.

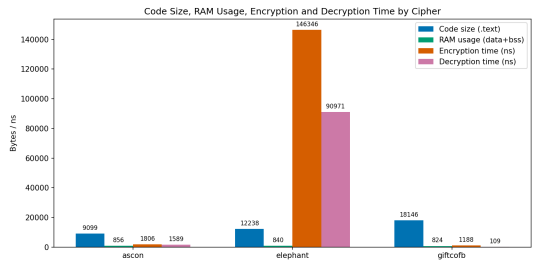


Fig. 3: Performance comparison across code size, RAM, encryption, and decryption timings.

As shown earlier in Figure 3, the performance trade-offs vary across ciphers. We now compare timing stability in Figure 4.

Nonce misuse represents one of the most damaging implementation mistakes in AEAD ciphers. To test this, we deliberately fixed the nonce and encrypted two different plaintexts with the same key–nonce pair. We then compared $C_1 \oplus C_2$ against $P_1 \oplus P_2$.

Table IV reports the bit–match ratios, with Figures 6 and 5 illustrating the results.

Cipher	Code Size (bytes)	RAM Usage (bytes)	Enc. Time (ns)	Dec. Time (ns)
ASCON	9099	856	14212	2913
Elephant	12238	840	160514	105798
GIFT-COFB	18146	824	24285	478

TABLE III: Single–iteration results: code size, RAM usage, and encryption/decryption timings.

Cipher	Message length (bytes)	Bit–match (%)
AES-GCM (OpenSSL)	16–4096	100.0
ChaCha20-Poly1305 (OpenSSL)	16–4096	100.0
ASCON-128 (ref)	16	74.9
	64	56.2
	512	50.8
	1500	50.3
	4096	50.1

TABLE IV: Bit–match percentage between $C_1 \oplus C_2$ and $P_1 \oplus P_2$ under nonce reuse.

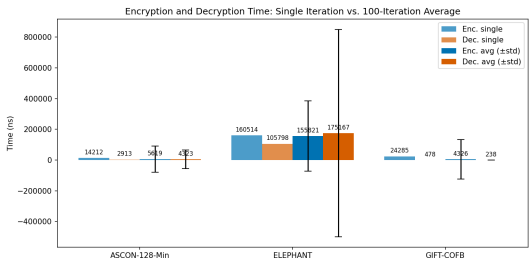


Fig. 4: Encryption/decryption timings: single iteration vs 100–iteration average.

As predicted by theory, **AES-GCM** and **ChaCha20-Poly1305** both failed completely under nonce reuse, revealing plaintext relations with 100% accuracy across all message sizes. This confirms their fragility in counter–mode–based constructions. In contrast, **ASCON-128** produced near–random results (around 50%), showing resilience against linear leakage, though integrity was still lost.

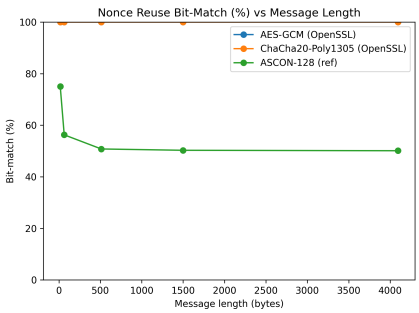
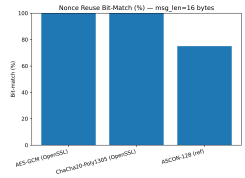
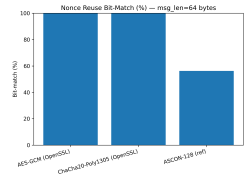


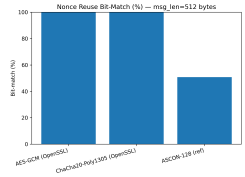
Fig. 5: Nonce reuse experiment: bit–match percentage across ciphers.



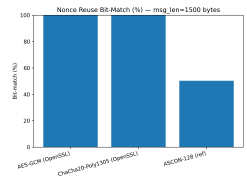
(a) AES-GCM, ChaCha20, ASCON-128 at 16 bytes



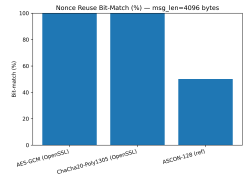
(b) AES-GCM, ChaCha20, ASCON-128 at 64 bytes



(c) AES-GCM, ChaCha20, ASCON-128 at 512 bytes



(d) ASCON-128 at 1500 bytes



(e) AES-GCM, ChaCha20, ASCON-128 at 4096 bytes

Fig. 6: Bit-match percentage under nonce reuse for AES-GCM, ChaCha20, ASCON-128 across different message sizes.

Figure 6 illustrates how ASCON-128 behaves under nonce reuse across different message sizes. At very small sizes (16 and 64 bytes), the bit-match rate is elevated (75% and 56%, respectively), reflecting small-sample bias. However, as the message size increases, the rates converge toward 50%, which is consistent with random noise and suggests the absence of linear leakage. This observation complements the overall

trend reported in Table IV. In contrast, Figure 5 aggregates the results across all ciphers. AES-GCM and ChaCha20-Poly1305 consistently show a 100% bit-match across all message lengths, confirming catastrophic vulnerability under nonce reuse. ASCON-128, on the other hand, stabilizes around 50%, highlighting a resilience property absent in traditional counter-mode AEADs.

VI. Conclusion

To conclude, performance metrics confirmed that different primitives are optimized for different constraints: ASCON offers compact code size with balanced performance, GIFT-COFB demonstrates exceptional decryption speed, albeit with a larger footprint, and Elephant, despite low RAM requirements, lags behind in throughput. These findings confirm that no single cipher dominates across all metrics, reinforcing the need for flexible benchmarking frameworks to guide algorithm selection for different application domains.

Beyond pure performance, the nonce misuse experiments highlighted an often-overlooked aspect of cryptographic evaluation: resilience under implementation errors. AES-GCM and ChaCha20-Poly1305, despite their ubiquity in real-world protocols, showed complete failure under nonce reuse, leaking plaintext relations with 100% certainty. ASCON-128, by contrast, exhibited bit-match ratios around 50%, reflecting behavior closer to random noise. Although integrity guarantees are still lost, this difference suggests that permutation-based lightweight designs may offer more robust misuse characteristics compared to counter-mode constructions. This underscores the unique contribution of our framework: moving beyond speed and memory benchmarking to also capture security degradation under realistic misuse scenarios. Overall, our framework provides a scalable and reproducible approach for evaluating lightweight AEAD ciphers, aligning with modern needs for both performance and secure–by–design assessment.

References

[1] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, pp. 450–466, 2007.

[2] W. Y. Jenny and M. D. Aagaard, "Benchmarking and optimizing aes for lightweight cryptography on asics," 2019.

[3] C. Pei, Y. Xiao, W. Liang, and X. Han, "Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, p. 117, 2018.

[4] M. S. Turan, M. S. Turan, K. McKay, D. Chang, L. E. Bassham, J. Kang, N. D. Waller, J. M. Kelsey, and D. Hong, *Status report on the final round of the NIST lightweight cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology, 2023.

[5] L. Cardoso dos Santos, J. Großschädl, and A. Biryukov, "FELICS-AEAD: Benchmarking of Lightweight Authenticated Encryption Algorithms," in *Smart Card Research and Advanced Applications*, pp. 216–233, Springer International Publishing, 2020.

[6] "AVR® Microcontrollers (MCUs)."

[7] "MSP430 microcontrollers | TI.com."

Uninvited Guests: Investigating Vulnerabilities in Smart Doorbell Surveillance Systems

1st Daniil Tashkan
SRH University of Applied Sciences Heidelberg
School of Technology and Architecture
Cyber Security Department, Leipzig, Germany
daniiltashkan@gmail.com

2nd Matin Lalehzari Mosala
SRH University of Applied Sciences Heidelberg
School of Technology and Architecture
Cyber Security Department, Leipzig, Germany
Matin.Lalehzari@gmail.com

3rd Klaus Dieter Schwarz
SRH University of Applied Sciences Heidelberg
School of Technology and Architecture
Cyber Security Department, Leipzig, Germany
klaus.schwarz@srh.de

***Abstract*—Smart doorbells are increasingly popular IoT devices designed to enhance home security; however, they may introduce significant vulnerabilities. Over the years, the market for video doorbells has increased drastically, further forecasted to expand even more; however, the vulnerability targeted in this study has not yet been discovered or addressed. Current video doorbell devices are designed in such a way as to support video recording either to the cloud ecosystem or to a local memory device, such as an SD card. This study identifies a critical physical attack vector based on the slot dedicated to local data storage, which allows an adversary to gain root access by inserting a malicious SD card into the device. Through hands-on reverse engineering and firmware extraction using tools such as a BIOS USB programmer and Linux-based analysis environments, multiple video doorbell models were examined to uncover security flaws in the mounting and execution of external storage devices. Some devices were found to execute scripts on SD cards with elevated privileges automatically and without any kind of authentication. This vulnerability permits attackers to flash firmware, view live video streams, and utilize the doorbell's Wi-Fi connection to influence other devices connected to the same network. Modern doorbells are mounted on a dedicated fastening that is secured with only a few screws, and removing it gives an attacker access to all the device input slots. Because of their typical outdoor placement, these doorbells are easily accessible, making the attack feasible with minimal exposure. Our findings demonstrate a dangerously overlooked threat within smart home ecosystems and underscore the urgent need for secure firmware design, authenticated external device handling, and improved physical safeguards in IoT hardware.**

***Index Terms*—Video-doorbell, IoT devices, vulnerability, exposure, physical attack**

I. Introduction

The Internet of Things (IoT) is revolutionizing human interaction with the world by transforming ordinary, routine objects into intelligent, interconnected devices [1]. The path towards digitalization is particularly noticeable in the use of smart home technologies, where an expanding ecosystem of sophisticated gadgets is reshaping modern households into environments where innovation and convenience seamlessly coexist. From voice-controlled virtual assistants to automated thermostats and security systems, these devices offer unprecedented convenience and control [2, 3]. However, with the increase in the configuration and adaptation of smart devices, the attack surface of a regular house increases [4]. By utilizing different devices and trying to make the house as secure as possible, digital doors can end up ajar to cyber threats. Our study delves into the issues and vulnerabilities of video doorbell devices, ultimately identifying a critical vulnerability. Over the years, a noticeable increasing trend in the purchase and development of smart home devices has been observed. As of 2025, the video doorbell market is estimated to be worth approximately 2.11 billion USD and is expected to rise up to 8.47 billion USD by the end of 2035 [5]. Our research identifies and analyzes a weakness in video doorbells, highlighting the need for better security practices in both hardware design and system architecture. As the market continues to expand and IoT devices become deeply embedded in everyday life, addressing such vulnerabilities is not only a matter of consumer safety but also a priority for the broader IoT ecosystem.

II. Related Work

Although IoT security has received considerable attention in recent years, much of the relevant literature tends to emphasize software-level threats or network-oriented attack surfaces. Fernandes et al., for example, conducted an extensive analysis of smart home platforms and uncovered threats emanating from excessive privileged access rights granted to third-party apps [6]. Similarly, Ronen et al. uncovered essential vulnerabilities in ZigBee-networked smart lighting systems, which were employed to demonstrate the spread of a local exploit through an IoT mesh network in a chain reaction [7]. Other notable studies, such as Hemram et al., have investigated firmware-level vulnerabilities in embedded devices and proposed defensive measures [8]. Despite extensive advancements, a significant literature gap exists in the field of physical attack vectors, namely those addressing

peripheral interfaces such as SD card slots. While Zillner [9] investigated general hardware and protocol weaknesses within IoT contexts, specific weaknesses related to boot-time execution of untrusted code from removable media have been addressed minimally. Smart doorbells, apart from large-scale use in residential and business environments, have not been the prime focus of past academic research in this context. Our study attempts to bridge this gap by exploring a new physical attack that does not involve remote code execution, Wi-Fi spoofing, or software misconfiguration. Instead, we observed a vulnerability in the initialization procedure of certain video doorbells that allowed auto-script execution with superuser permissions from SD cards during boot time. This attack circumvents conventional digital security mechanisms and exploits the physical vulnerability of the device and the lack of firmware security features. To the best of our knowledge, this is the first academic study to systematically investigate SD card-based privilege escalation in smart doorbells and present concrete evidence of its feasibility.

III. Problem Statement

Smart doorbells have become more widespread; however, despite their usefulness, they pose significant vulnerabilities if not adequately secured. This research reveals a vulnerability through which an attacker with temporary physical access can compromise a smart doorbell by inserting an SD card holding a malicious script. Because these products are typically deployed outside, they are easily accessible and vulnerable to surreptitious physical attacks [4, 10]. This attack allows unauthorized firmware modifications, surpassing any type of authentication. Complete device control is granted, revealing sensitive features such as real-time video streams, entry logs, and views of shared building spaces, which pose significant privacy and security risks. In addition, by exploiting the Wi-Fi connectivity of the doorbell, an attacker can move laterally within the local network, compromise other devices present in the network, and steal sensitive data, including banking details. In worst-case scenarios, insider information about residents' routines may also facilitate physical theft [2, 11]. This study investigates the level of such threats and proposes lightweight measures against unauthorized device tampering and general network penetration.

IV. Methodology

To address this vulnerability, several devices and software are crucial.

- A doorbell device with a video camera.
- BIOS USB Programmer CH341A.
- USB TTL Adapter CH340G
- Regular micro SD card.
- Kali Linux machine
- Binwalk

To understand the operational mechanics of video doorbell devices, we examined their internal file systems and architecture. Several devices were acquired and carefully disassembled to access their memory integrated circuits, which store the core functionalities, scripts, and drivers of the devices. This hardware-level analysis aimed to identify potential security vulnerabilities by examining the critical initialization files and system rules. Using a BIOS USB programmer with specialized software, we successfully extracted and analyzed the binary files of the device. The Binwalk tool revealed the complete internal software structure, allowing us to map the device architecture and understand how its hardware, firmware, and data storage components interact. Our research focused on identifying attack vectors that malicious actors could exploit without requiring physical device disassembly. Because IoT devices typically run on Linux-based architectures, we examined how the file system handles external storage devices such as SD cards. Through this analysis, we discovered various scripts that granted both read and write functionalities to the SD card. More significantly, we identified recovery scripts designed to execute external scripts stored on SD cards with specific names for system restoration purposes. This discovery revealed a potential vulnerability: malicious scripts could be placed on an SD card using the required naming, potentially compromising the device when the recovery mode is triggered. To validate this theory, we connected a CH340G adapter to the UART interface of the device to monitor the boot processes. While the device immediately recognized the inserted SD cards, executing malicious scripts required triggering the recovery mode. We attempted multiple approaches to activate the recovery mode, including power cycling, altering the boot cell voltage, executing remote commands using Python libraries with the device's local key, and various other methods. Despite clear evidence in the firmware code suggesting the existence of this vulnerability, we have not yet found a reliable method to exploit it. However, the attack vector remains theoretical, requiring further research to determine practical implementation methods.

V. Key Findings

This study identified a critical vulnerability in widely deployed smart doorbells, allowing attackers to gain root access by physically inserting an SD card containing a preconfigured malicious code. The script is executed automatically upon insertion and initiation of the recovery mode of the device. Root privileges were granted to the SD card, bypassing authentication mechanisms and enabling unauthorized read/write access to the device. With root-level control, an attacker can modify the firmware, reset the credentials, and fully manipulate the device functionality. Specifically designed malware may establish a remote shell via the doorbell's Wi-Fi connection, effectively serving as a gateway into the broader home network. This access facilitates further attacks, including eavesdropping, spoofing, session hijacking, data exfiltration, and smart device control. The extent of the attack is limited only by the attacker's knowledge and intent. Due to the outdoor installation of the device, the exploit can be performed discreetly and efficiently, with very little time to perform. These findings expose a potential physical attack vector within the IoT infrastructure.

VI. Estimated solution

To address the identified weakness, this study suggests a realistic set of countermeasures that would limit unauthorized physical access and exploitation. First, the intelligent doorbell should be installed using tamper-evident hardware or enclosures that are difficult to remove in haste, thereby increasing the time and risk elements associated with physical attacks. A protected security seal, breaking upon device opening, or a tamper-evident screw, showing if it was removed. Second, the system's architecture must be reset so that the device can write onto the SD card but not read from it during normal use, preventing automatic execution of malicious scripts. Third, the SD card slot must be positioned internally or out of sight so that even if the device is physically taken off, it is still not accessible without full disassembly, again preventing possibilities of successful tampering. Finally, the device firmware should be altered to allow only write functionality to the SD card. Collectively, these actions make it more difficult to physically compromise the device and significantly reduce the attack surface area for SD card-based attacks.

VII. Conclusion

The rapid adoption of smart home devices, particularly video doorbells, reflects a new reliance on networked technologies to safeguard our homes. However, our work reveals a critical vulnerability that exposes a particular device to physical exploitation. By demonstrating how simple physical access can grant root-level control and establish a foundation for further network attacks, this study highlights a dangerously overlooked security gap in the IoT infrastructure. Given the expanding market for these devices, addressing these vulnerabilities must become a priority. Security cannot be an afterthought in the design of these devices. This paper presents a physical vulnerability in commercially available smart video doorbells, specifically involving the automatic execution of malicious scripts from SD cards with root access. Unlike prior work that focuses on remote or network-based attacks, this study reveals a novel, practical exploit vector that requires only brief physical access without device disassembly or authentication. The vulnerability was confirmed through the extraction of firmware and behavioral analysis of multiple models, which demonstrated the execution of unauthorized code during device startup. Given the widespread use and outdoor placement of these devices, this finding exposes a high-risk threat to both privacy and overall network security. This study highlights the need for improved firmware protection, authenticated processing of external storage, and physical design modifications to prevent unauthorized access.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] Y. Sun and S. Li, "A systematic review of the research framework and evolution of smart homes based on the internet of things," *Telecommunication Systems*, vol. 77, no. 3, pp. 597–623, Jul 2021. [Online]. Available: https://doi.org/10.1007/s11235-021-00787-w

[3] S. Venkatraman, A. Overmars, and M. Thong, "Smart home automation—use cases of a secure and integrated voice-control system," *Systems*, vol. 9, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2079-8954/9/4/77

[4] B. Hammi, S. Zeadally, R. Khatoun, and J. Nebhen, "Survey on smart homes: Vulnerabilities, risks, and countermeasures," *Computers & Security*, vol. 117, p. 102677, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740482200075X

[5] S. Saha, "Video doorbell market by product type, end user, sales channel, and region – growth, trends, and forecast through 2035," Future Market Insights, Tech. Rep., apr 2025. [Online]. Available: https://www.futuremarketinsights.com/reports/video-doorbell-market

[6] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 636–654.

[7] E. Ronen, A. Shamir, A.-O. Weingarten, and C. O'Flynn, "Iot goes nuclear: Creating a zigbee chain reaction," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 195–212.

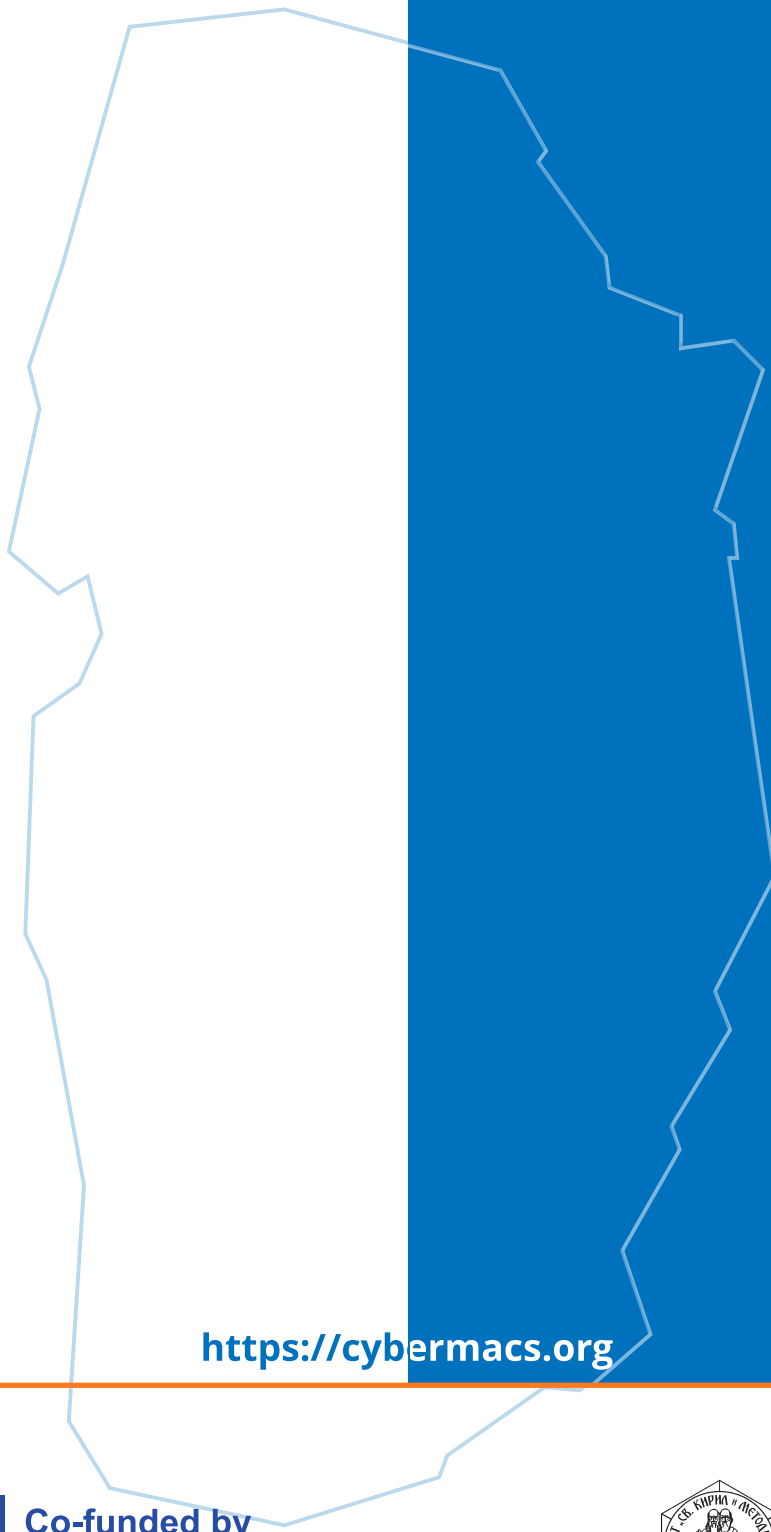
[8] S. Hemram, G. J. W. Kathrine, G. M. Palmer, and S. V. Ewards, "Firmware vulnerability detection in embedded systems and internet of things," in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2022, pp. 1161–1167.

[9] T. Zillner, "Zigbee exploited the good , the bad and the ugly," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:17543018

[10] A. Allen, A. Mylonas, S. Vidalis, and D. Gritzalis, "Smart homes under siege: Assessing the robustness of physical security against wireless network attacks," *Computers & Security*, vol. 139, p. 103687, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823005977

[11] B. Cruz, S. Meire, D. Ruano Ordás, H. Janicke, I. Yevseyeva, and J. Méndez Reboredo, "A practical approach to protect iot devices against attacks and compile security incident datasets," *Scientific Programming*, vol. 2019, pp. 1–11, 07 2019.

-



https://cybermacs.org

Co-funded by
the European Union