



Universitatea
Transilvania
din Brașov

INTERDISCIPLINARY DOCTORAL SCHOOL

Faculty of Electrical Engineering and Computer Science

Eng. Emmanuel TUYISHIME

Addressing Cybersecurity Challenges in IoT/IIoT-based Remote Engineering Systems

SUMMARY

Scientific supervisor

Prof.Dr. Petru Adrian COTFAS

BRASOV, 2026

TABLE OF CONTENTS

TABLE OF CONTENTS	1
ABSTRACT	3
Chapter 1. Introduction	4
1.1. Background and Research Motivation	4
1.2. Aim and Objectives of Research	4
1.3. Structure of the Thesis	5
Chapter 2. Remote Engineering Systems and Cybersecurity Challenges	6
2.1. Remote Engineering in Education	6
2.1.1 Background and Laboratory Paradigms	6
2.1.2 Cybersecurity Challenges and Mitigation Strategies in RL Systems	6
2.2. Remote Engineering in Industry	8
2.2.1 Background and Evolution	8
2.2.2 Cybersecurity Vulnerabilities, Threat Actors, and Attack Vectors in Industrial RMC Systems	9
2.2.3 Cybersecurity Mitigation Strategies and Best Practices for Industrial Systems	9
2.3. Conclusion	10
Chapter 3. Design and Implementation of Zero Trust Access Control Architecture for Remote Engineering Systems	11
3.1. Implementation in a Remote Laboratory Environment	11
3.1.1 System Architecture and Deployment Context	11
3.1.2 ZTNA Architecture, Access Control, and Device Security Policy	12
3.2. Discussions	13
3.3. Chapter Conclusion	14
Chapter 4. Feature Selection–Driven Resource-Efficient Machine Learning-Based Traffic Classification for MQTT-IoT Security Attacks	15
4.1. Methodology and Experimental Design	15
4.1.1 Dataset Description	15
4.1.2 Feature Selection	16
4.2. Experimental Design and Evaluation	16
4.2.1 Binary Classification Setup and Results	16
4.2.2 Multiclass Classification Setup and Results	17
4.3. Discussion of Results	19
4.4. Model Selection and Practical Implications	20
4.5. Chapter Conclusion	20
Chapter 5. Development of a Modular Architecture for Real-Time ML-Driven Intrusion Detection in IoT/IIoT with Privacy-Aware Cloud SIEM Integration	21
5.1. System Architecture Design	21
5.2. System Platforms Integration and Communication	22

5.2.1	Sensor-Level Data Acquisition	22
5.2.2	Node-RED-Based Middleware and Integration Layer.....	22
5.2.3	Privacy-Preserving Cloud-Integration.....	23
5.2.4	Threats Model and Attack Scenarios.....	23
5.3.	Machine Learning Inference	24
5.3.1	Runtime Model Switching Mechanism for Adaptive IDS Inference	24
5.4.	Results, Performance Evaluation, and System-Level Analysis.....	25
5.4.1	Detection Performance and Accuracy Analysis.....	25
5.4.2	Latency and Throughput Performance.....	25
5.4.3	Runtime Adaptation and Model Switching Overhead.....	26
5.4.4	Cloud Integration, Operational Validation, and Cost Efficiency	27
5.5.	Chapter Conclusion	28
Chapter 6.	Resilient Hybrid Edge–Fog Architectures for ML-Driven Intrusion Detection in IoT/IIoT Remote Monitoring and Control Systems.....	29
6.1.	System Architecture and Experimental Testbed	29
6.2.	Experimental Scenarios and Evaluation Methodology	30
6.3.	Results and Discussions	31
6.3.1	Control Latency and Reliability Performance	31
6.3.2	Resource Utilization Analysis.....	31
6.3.3	Inference Latency and Throughput Performance	31
6.4.	Chapter Conclusion.....	32
Chapter 7.	Final Conclusions, Original Contributions, Dissemination of Results, and Further Research	34
7.1.	Conclusions	34
7.2.	Original Contributions of Research.....	34
7.3.	Dissemination of Research Results	35
7.4.	Directions for Further Research.....	36
References		37

ABSTRACT

Remote engineering systems enable interaction with physical processes and engineering infrastructures over distributed networks and have become increasingly prominent in both engineering education and industrial practice. Advances in networking, cloud computing, and IoT/IIoT technologies, accelerated by Industry 4.0 initiatives, have expanded their flexibility and scalability while simultaneously increasing exposure to cyber threats. As previously isolated systems become network-connected, conventional perimeter-based access control models and resource-intensive intrusion detection mechanisms struggle to meet contemporary security, performance, and resilience requirements.

This thesis addresses these challenges by designing, implementing, and experimentally evaluating complementary cybersecurity solutions for IoT- and IIoT-based remote engineering systems. The research is structured around three interconnected directions: identity-centric access control, resource-efficient intrusion detection, and resilience-oriented deployment architectures.

The work begins with a systematic analysis of remote engineering environments across educational and industrial domains, identifying key technological trends, cybersecurity challenges, and research gaps. To strengthen access control, a Zero Trust architecture is designed and validated within an online laboratory environment, demonstrating the applicability of identity-centric, policy-enforced access control in remote engineering systems.

To improve attack detection under resource constraints, a lightweight machine learning–driven intrusion detection approach is proposed. The approach introduces a statistical feature-selection methodology that reduces feature dimensionality while maintaining competitive detection performance for MQTT-based IoT security attacks. Building on this foundation, the research investigates deployment-aware intrusion detection architectures through Edge–Cloud and hybrid Edge–Fog implementations, enabling real-time monitoring while preserving privacy and operational efficiency.

Experimental results demonstrate that the proposed approaches achieve effective intrusion detection with substantially reduced computational overhead while supporting real-time deployment in resource-constrained environments. The findings provide an experimentally validated foundation for enhancing the cybersecurity and operational resilience of IoT/IIoT-based remote engineering systems and establish directions for future research in adaptive security, distributed intelligence, and resilient cyber–physical systems.

Chapter 1. Introduction

1.1. Background and Research Motivation

Remote engineering systems enable users to interact with physical processes and engineering infrastructures over distributed networks and have become increasingly significant in both engineering education and industrial practice. In educational settings, remote laboratories extend access to physical experimentation beyond traditional temporal and geographical constraints, facilitating distance learning and resource sharing [1], [2]. In industrial contexts, remote monitoring and control (RMC) systems support supervision, maintenance, and automation of geographically dispersed assets [3], [4].

The rapid adoption of remote engineering has been driven by advances in networking technologies, cloud computing, the Internet of Things (IoT), and Industrial IoT (IIoT) [5], and has been further accelerated by Industry 4.0 initiatives [6]. From smart household devices to industrial smart factories, assets and facilities are now digitally interconnected across vast geographical areas [6]. This transformation was particularly underscored during the COVID-19 pandemic, which compelled industries and educational institutions to adopt remote operations and hybrid modalities at unprecedented speed [7], [8]. Consequently, remote accessibility has transitioned from convenience to an operational necessity [9].

However, these advances also introduce substantial cybersecurity risks. Distributed remote engineering platforms operate across heterogeneous, multi-layered communication infrastructures that integrate industrial assets, resource-constrained embedded devices, open networking protocols, and cloud-based services [10], [11]. While this architectural composition enables scalable monitoring, experimentation, maintenance, and remote interaction, it simultaneously expands the attack surface across sensing, communication, and application layers [12]. The convergence of legacy operational technology (OT) with modern connectivity mechanisms further complicates security enforcement and limits the applicability of conventional enterprise security models [13].

Traditional cybersecurity approaches, largely developed for conventional enterprise IT infrastructures, do not always align well with the operational constraints of remote engineering systems. Many existing remote access control mechanisms remain perimeter-based and tend to rely on implicit trust once network access is established, often via Virtual Private Networks (VPNs). The perimeter-based approach faces growing challenges in modern enterprise contexts that increasingly span remote users, bring-your-own-device (BYOD) practices, and cloud-based assets [14]. In addition, these perimeter-based models offer limited protection against insider and insider-enabled threats, where misuse of legitimate access rather than external intrusion becomes a primary risk factor [15].

At the same time, computational constraints inherent to remote engineering systems, particularly IoT/IIoT-based RMC infrastructures, limit the feasibility of deploying resource-intensive security mechanisms, especially at the edge where sensors, actuators, and embedded controllers operate. Consequently, security mechanisms must be designed to preserve real-time operational integrity under resource constraints [16], [17].

Against this background, the central research problem addressed in this thesis concerns how the cybersecurity of remote engineering systems, particularly IoT/IIoT-based RMC infrastructures, can be strengthened through the design and deployment-aware security mechanisms that address identity-centric access control, anomaly detection, and the distribution of detection capabilities across system layers, while preserving real-time operational requirements and system reliability.

1.2. Aim and Objectives of Research

The primary aim of this thesis is to design, implement, and experimentally validate cybersecurity mechanisms and deployment strategies that enhance the security and resilience of IoT/IIoT-based remote engineering systems.

To achieve this aim, the thesis pursues the following objectives:

- O1. To analyze and synthesize the architectural characteristics, enabling technologies, and security challenges of remote engineering systems.

- O2. To investigate and operationalize Zero Trust as a foundational access control model for remote engineering environments.
- O3. To design, evaluate, and deploy resource-efficient machine learning–based intrusion detection mechanisms for IoT/IIoT-based remote engineering systems, considering both detection performance and deployment constraints.

1.3. Structure of the Thesis

The thesis is structured as follows:

- Chapter 1 introduces the necessity and motivation for the thesis, defines the research aim and objectives, and outlines the overall structure of the thesis.
- Chapter 2 presents the conceptual and technological foundations of remote engineering systems, examining their architecture, enabling technologies, and cybersecurity challenges across both educational and industrial domains.
- Chapter 3 develops and validates an identity-centric Zero Trust access control architecture for remote engineering environments through a practical online laboratory case study.
- Chapter 4 proposes a feature selection–driven, resource-efficient machine learning methodology for the classification of MQTT-based IoT security attacks.
- Chapter 5 develops a modular and privacy-aware infrastructure for deploying real-time, machine learning–based intrusion detection with integrated cloud-based SIEM monitoring.
- Chapter 6 investigates the operational resilience of IoT/IIoT-based remote monitoring and control systems under cyberattacks and proposes a hybrid IDS deployment validated through experimental evaluation.
- Chapter 7 concludes the thesis by synthesizing the main findings and original contributions, summarizing the dissemination of results, and outlining directions for future research.

Chapter 2. Remote Engineering Systems and Cybersecurity Challenges

2.1. Remote Engineering in Education

2.1.1 Background and Laboratory Paradigms

Engineering education has traditionally relied on physical laboratories (face-to-face), where students interact directly with experimental equipment in a co-located environment to reinforce conceptual understanding and gain hands-on experience. These traditional hands-on laboratories offer pedagogical benefits but are limited by factors such as cost, equipment availability, space, maintenance, and safety risks [18], [19].

To overcome these limitations, alternative laboratory paradigms have been introduced, including virtual laboratories (VLs), which rely on software-based simulations, and remote laboratories (RLs), which enable interaction with real physical equipment over networked infrastructures. More broadly, these environments are often referred to as online laboratories, encompassing both remotely accessible simulated experiments (VLs) and remotely operated plants (Remote Labs), representing network-accessible platforms that support experimentation beyond temporal and spatial constraints [20].

Laboratory environments are commonly classified based on two dimensions: the location of the user (local or remote) and the nature of the experimental resources (physical or virtual) [21], [22]. This leads to four primary categories:

1. Physical hands-on laboratories, involving direct interaction with physical equipment.
2. Local virtual laboratories, based on simulations executed on local machines.
3. Remote virtual laboratories, where simulated environments are accessed via networks.
4. Remote laboratories, which provide network-based access to real physical equipment.

Among these paradigms, RLs have gained particular importance due to their ability to preserve the realism of physical experimentation while improving accessibility and scalability. RLs allow users to interact with geographically distributed physical devices through web-based interfaces, thereby extending laboratory access to a broader audience [1], [23], [24].

Over time, these laboratory paradigms have evolved into fully integrated components of engineering education. They support large-scale deployment, cross-institutional sharing, and continuous access, addressing key challenges related to scalability and pedagogical flexibility. Their significance was further highlighted during the COVID-19 pandemic, which accelerated the adoption of online and hybrid learning modalities [25], [26].

Following their maturation and widespread adoption, virtual and remote laboratories have increasingly been implemented as organized, large-scale infrastructures rather than isolated instructional tools. A substantial body of work reports the development of online laboratory platforms that support learning and experimentation across various domains, including electronics, power and microelectronics, electrical drives, renewable energy, chemistry, physics, and control and automation [1], [27]–[33]. Early implementations, however, were predominantly institution-specific and custom-built, resulting in isolated systems with limited interoperability, reuse, and cross-institutional accessibility [34]. These constraints hindered broader adoption and inter-institutional sharing.

To address these limitations, research has shifted toward federated architectures that enable integration of geographically distributed laboratories within shared frameworks [34]. In this context, Remote Laboratory Management Systems (RLMS) emerged as a key enabling layer, providing unified access, scheduling, and experiment management for heterogeneous remote laboratory resources [35].

2.1.2 Cybersecurity Challenges and Mitigation Strategies in RL Systems

Given their internet-facing and cyber-physical nature, RL systems introduce a complex and expanded attack surface, exposing both digital assets and physical equipment to cybersecurity risks. These risks

affect not only the traditional information security properties of confidentiality, integrity, and availability (CIA), but also extend to non-repudiation, authentication, privacy, and safety (CIANA+PS) [36], [37]. Unlike conventional laboratory environments, RL systems are persistently accessible, highly interconnected, and often deployed within broader institutional networks, thereby increasing their susceptibility to both active and passive cyber-attacks [38].

From a foundational perspective, cybersecurity threats in RL systems should be understood within the broader definition of cyber threats articulated by the National Institute of Standards and Technology (NIST), which defines a threat as any circumstance or event capable of adversely impacting organizational operations, assets, or individuals through unauthorized access, modification, disclosure, or denial of service [39]. In educational RL environments, such threats directly undermine the reliability of experiments, the integrity of generated data, and, in some cases, the physical safety of laboratory equipment and users.

Several studies indicate that the security mechanisms of web-based applications supporting remote experiments remain insufficient [40], [41]. This deficiency has largely been attributed to development practices that prioritize educational content, experimentation, and rapid deployment over the systematic integration of security considerations [37]. As a result, remote RL systems frequently inherit vulnerabilities stemming from weak authentication mechanisms, insecure web services, and insufficiently monitored infrastructure.

The vulnerabilities manifest in multiple attack vectors that threaten the CIANA+PS properties of RL systems.

- Web Application Attacks: SQL Injection and XPath Injection attacks exploit insufficient input validation and application hardening [42], [43]. RL platforms have been shown to exhibit such weaknesses [37], enabling unauthorized data access, manipulation, and arbitrary code execution, potentially compromising experimental integrity.
- Communication-Layer Attacks: Man-in-the-Middle (MITM) attacks exploit insecure communication channels [37]. Beyond credential interception and privacy violations, such attacks may enable adversarial manipulation of experimental parameters, thereby introducing safety and equipment-integrity risks.
- Availability Attacks: Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks can disrupt RL operations by overwhelming servers or network infrastructure [37]. Additionally, abuse of laboratory reservation mechanisms may create application-layer denial-of-service conditions.
- Device-Based and Cyber-Physical Attacks: RL systems are exposed to firmware exploitation, hardware tampering, and environmental manipulation, which may compromise system availability, data integrity, and safe operation.

While documented attacks explicitly targeting RL systems remain limited, broader evidence highlights substantial risk exposure. Multiple reports consistently rank the education sector among the most targeted industries for malware, phishing, and ransomware campaigns [86]–[89], and indicate comparatively low levels of cyber preparedness [90], [91]. Recent reports indicate a significant rise in ransomware attacks against universities, frequently exploiting remote-access vulnerabilities, with further escalation expected as remote infrastructure expands [92].

Consequences include data loss, intellectual property compromise, regulatory violations, reputational damage, and erosion of trust in remote experimentation [26], [44]. Importantly, the cyber-physical nature of RL systems enables cross-domain risk propagation, where digital intrusions may trigger physical damage or safety hazards.

Despite these risks, security considerations remain secondary in much of the literature [37]. Reviews in [45] and [26] identified relatively few works explicitly addressing safety and security in federated RL and OLMS environments.

- Authentication and Access Control: Early mitigation strategies emphasized username–password authentication schemes and alternative approaches such as image-based passwords. More recent

research explored blockchain-assisted authentication and decentralized verification mechanisms to enhance resilience against credential compromise in distributed RL infrastructures [46]. However, authentication alone does not ensure comprehensive system security.

- Tiered Security Models: Krbeček et al. [47] proposed a security model scaling protection mechanism according to experiment complexity. While useful in illustrating differentiated security requirements, this approach risks underestimating low-complexity experiments as potential entry points into institutional networks.
- Architectural and Virtualization-Based Controls: Client–server separation, virtualization, sandboxing, read-only images, and network segmentation have been proposed to reduce attack surfaces and contain malware propagation [48].
- Governance and Standards Alignment: Security zoning, dynamic trust management, and regulatory compliance, including GDPR adherence, have been proposed as governance mechanisms [45]. Alignment with international standards such as ISO/IEC 27000, ISO/IEC 15048, and IEC 62443 supports systematic risk management.
- Network-Level Defenses: VLAN segmentation, encryption, IDS/IPS deployment, web application firewalls, honeypots, VPN usage, and continuous traffic monitoring are recommended to strengthen network resilience [37]. Protocol-specific hardening, including secure MQTT integration in IoT-based RL systems, has also been explored [49].

2.2. Remote Engineering in Industry

2.2.1 Background and Evolution

Industrial systems were traditionally designed as localized and largely isolated environments, with monitoring and control activities confined to on-site facilities [50]. Early forms of remote operation relied on short-range wired solutions intended to extend local control rooms rather than enable true geographic separation between operators and industrial processes [51]. This architectural isolation was motivated not only by technological constraints but also by safety, reliability, and security considerations inherent to industrial environments.

The evolution toward industrial RMC began in the mid-20th century with advances in digital computing and telecommunications [52]. The introduction of programmable logic controllers (PLCs), supervisory control and data acquisition (SCADA) systems, and distributed control systems (DCS) marked a major shift, enabling more flexible, scalable, and geographically distributed control of industrial processes [5], [53]. These technologies laid the foundation for modern RMC systems by allowing operators to monitor system states and issue control commands from increasingly remote locations [50].

Subsequent integration of digital communication protocols, Internet connectivity, and wireless technologies further expanded the reach of RMC systems, facilitating centralized supervision of geographically distributed assets [51]. As a result, RMC solutions became widely adopted across critical infrastructure sectors, including energy, transportation, healthcare, water treatment, and manufacturing, driven by shared objectives of operational efficiency, reliability, and safety [10], [54].

In modern industrial settings, RMC solutions have transitioned from optional capabilities into operational necessities. Many sectors now depend on RMC systems to maintain productivity, ensure personnel safety, and reduce operational costs [55], [56]. This dependency was particularly evident during the COVID-19 pandemic, when remote access to industrial systems became crucial for maintaining business continuity and protecting the workforce [57]–[59].

Industrial RMC systems support diverse applications, including teleoperation in hazardous environments [60], [61], robotic and automated process control, and real-time process monitoring [62]. Additionally, RMC technologies enable remote maintenance, diagnostics, training, and simulation, enhancing operational resilience and reducing downtime [51], [63].

2.2.2 Cybersecurity Vulnerabilities, Threat Actors, and Attack Vectors in Industrial RMC Systems

Industrial RMC systems present a significantly expanded attack surface resulting from the convergence of legacy operational technologies, insecure design practices, and increased integration with IT networks and Internet-facing services [13], [64]. Vulnerabilities stem from outdated or unsupported systems and protocols lacking built-in security, such as Modbus and DNP3, which often transmit data in plaintext and are susceptible to interception and manipulation [65], [66].

These weaknesses are further exacerbated by inadequate network segmentation between IT and OT environments, which enables lateral movement and the compromise of critical assets. Additional risks arise from weak authentication and access control mechanisms, including default credentials, poor privilege management, and insecure remote access technologies such as Telnet and unencrypted RDP [12], as well as vulnerabilities introduced by third-party components and supply chain dependencies [65]. Operational constraints, particularly delayed patch management due to downtime and compatibility concerns, leave known vulnerabilities unaddressed [67], while insufficient physical security enables direct tampering and reinforces the convergence of cyber-physical risks.

These systemic weaknesses have made industrial environments prime targets for sophisticated threat actors exploiting both known and zero-day vulnerabilities [68], [69], as demonstrated by high-impact incidents such as Stuxnet (2010), which enabled the physical destruction of industrial equipment [70], [71], the German steel mill attack (2014) and Ukrainian power grid disruptions (2015–2016), and large-scale ransomware and supply chain attacks, including NotPetya, WannaCry, Colonial Pipeline (2021), and SolarWinds (2020) [72], alongside more recent disruptions affecting critical infrastructure and healthcare systems [73], [74]. Industrial systems are exposed to diverse attack vectors, including malware, ransomware, phishing, SQL injection, removable media attacks, and, in remote access scenarios, credential compromise, impersonation, Man-in-the-Middle (MITM), replay, false data injection, and denial-of-service attacks targeting communication and authentication infrastructures [71], [72]

The threat landscape is further intensified by the diversity of adversaries, ranging from insiders and cybercriminal groups to industrial spies and nation-state actors, whose motivations span financial gain, espionage, and geopolitical objectives, thereby increasing attack sophistication and complicating detection, attribution, and response [53], [71].

2.2.3 Cybersecurity Mitigation Strategies and Best Practices for Industrial Systems

Table 2.1 summarizes the main cybersecurity mitigation strategies for industrial systems, organized into key functional categories addressing prevention, detection, resilience, and governance.

Table 2.1. Summary of cybersecurity mitigation strategies and best practices for industrial systems.

Category	Key Measures	Purpose	References
Prevention & Detection	Firewalls, IDS/IPS, monitoring	Network segmentation and layered threat detection to mitigate insider threats and zero-days.	[53], [65], [75], [65], [76]
	TLS encryption	Ensures data confidentiality and integrity in transit.	[77], [78]
Secure Communication & Authentication	Passive Feature Learning (PFL)	Enables threat detection in encrypted traffic.	[78]
	Cryptographic authentication (AES, SHA-2)	Prevents spoofing, replay, and unauthorized access	[53], [77]

	MFA, Zero Trust	Enforce strong identity verification and continuous access control.	[79],[80], [81]
Security Testbeds	Physical / Virtual / Hybrid testbeds	Enable safe evaluation of attacks and defenses under controlled conditions.	[69], [82], [75]
	VICSORT and similar frameworks	Support scalable, resource-efficient experimentation.	[83]
Resilience & Recovery	Redundancy, failover, segmentation	Maintain operation and limit impact during disruptions.	[84], [85]
	D2R2+DR lifecycle	Structured resilience: defend, detect, remediate, recover, refine	[86], [87]
Standards & Frameworks	ISA/IEC 62443, NIST SP 800-82, ISO/IEC 27000	Provide risk-based security governance and compliance.	[88], [89] [90], [91]
	MITRE CREF, ENISA	Guide resilience engineering and secure remote access.	[92],[93]

2.3. Conclusion

This chapter established the conceptual foundations of remote engineering systems across both educational remote laboratories and industrial RMC environments. Despite differing objectives, both domains rely on interconnected IoT/IIoT-based architectures that incorporate cloud platforms, web services, industrial protocols, and remote access technologies, significantly expanding the attack surface.

The analysis showed that educational RLs, while improving accessibility, often suffer from weak authentication, insecure communication, and limited monitoring. In contrast, industrial systems face similar issues compounded by legacy technologies, insecure protocols, IT–OT convergence, and availability-driven design priorities. Across both domains, recurring vulnerabilities include weak identity management, inadequate network segmentation, insecure communication channels, and limited intrusion detection capabilities, with industrial incidents demonstrating the potential for severe operational and physical consequences.

A cross-domain synthesis reveals key gaps: fragmented and domain-specific security approaches, limited consideration of cross-layer cyber–physical interactions, and insufficient support for resource-constrained environments. In particular, the lack of lightweight, deployment-ready security mechanisms at the edge and gateway layers remains a critical limitation.

These findings highlight the need for a unified, deployment-aware cybersecurity approach that integrates identity-centric access control, efficient intrusion detection, and layered protection mechanisms across system architectures. Accordingly, this thesis addresses these challenges by combining an investigation of Zero Trust access control, resource-efficient ML-based IDS, and adaptive deployment strategies, thereby fulfilling objective O1 and establishing the foundation for subsequent contributions.

Chapter 3. Design and Implementation of Zero Trust Access Control Architecture for Remote Engineering Systems

Remote engineering systems invalidate traditional perimeter-based security assumptions by requiring continuous remote access to physical resources over Internet-facing infrastructures. The conventional “castle-and-moat” model, which assumes trusted internal networks, is no longer adequate in modern environments characterized by distributed users, cloud integration, and increased exposure to insider and lateral movement threats [94], [95].

VPN-based remote access, while widely adopted, extends network trust to authenticated users and introduces significant security risks due to vulnerabilities, misconfigurations, and implicit trust models. The growing number of VPN-related exploits, including ransomware and credential attacks, highlights these limitations [80].

In response, the Zero Trust (ZT) paradigm has emerged, replacing implicit trust with continuous verification based on identity, context, and policy under the principle of “never trust, always verify” [96]. ZT emphasizes identity-based access control, least-privilege principles, and continuous monitoring [97], it has gained widespread adoption as a response to evolving cyber threats [81].

Despite its maturity, the application of Zero Trust, particularly Zero Trust Network Access (ZTNA), to remote engineering systems remains underexplored in academic research. This chapter addresses this gap by proposing and implementing a ZTNA-based access-control architecture for remote laboratory environments, demonstrating that identity-centric, policy-driven security can replace traditional perimeter-based models.

3.1. Implementation in a Remote Laboratory Environment

3.1.1 System Architecture and Deployment Context

The ZT access control architecture was implemented and evaluated in an educational remote laboratory environment that hosts Internet-accessible experiments. The remote laboratory infrastructure employed in this chapter is based on an online serial laboratory platform designed to support remote experimentation with serial communication technologies, including RS232, RS422, and RS485 [98]. In this thesis, the laboratory serves as an operational remote-engineering system, providing a realistic, constrained environment for evaluating ZT-based access-control mechanisms.

As illustrated in Figure 3.1, the proposed ZT security architecture is implemented using a ZTNA framework composed of the following logical components:

- Identity Provider (IdP): responsible for user authentication.
- PDP: evaluates access requests against defined policies.
- PEP: mediates and enforces authorized traffic.

In this implementation, a commercial ZTNA platform [99] is used to realize these functions. The controller component acts as the PDP, handling policy evaluation and issuing access tokens. At the same time, connectors function as PEPs, enforcing access decisions by controlling traffic flow between users and laboratory resources. The primary security objectives of the proposed architecture are to prevent unauthorized access to laboratory resources, mitigate credential misuse and lateral movement, protect experimental integrity and availability, and reduce exposure to network-based attacks.

Accordingly, to enforce clear trust boundaries, the architecture is logically partitioned into three primary trust zones:

- User zone: user devices and client applications operating outside institutional control.
- Control zone: the IdP, PDP, and PEP responsible for identity verification and access enforcement.
- Remote resource zone: experiment servers and physical laboratory devices.

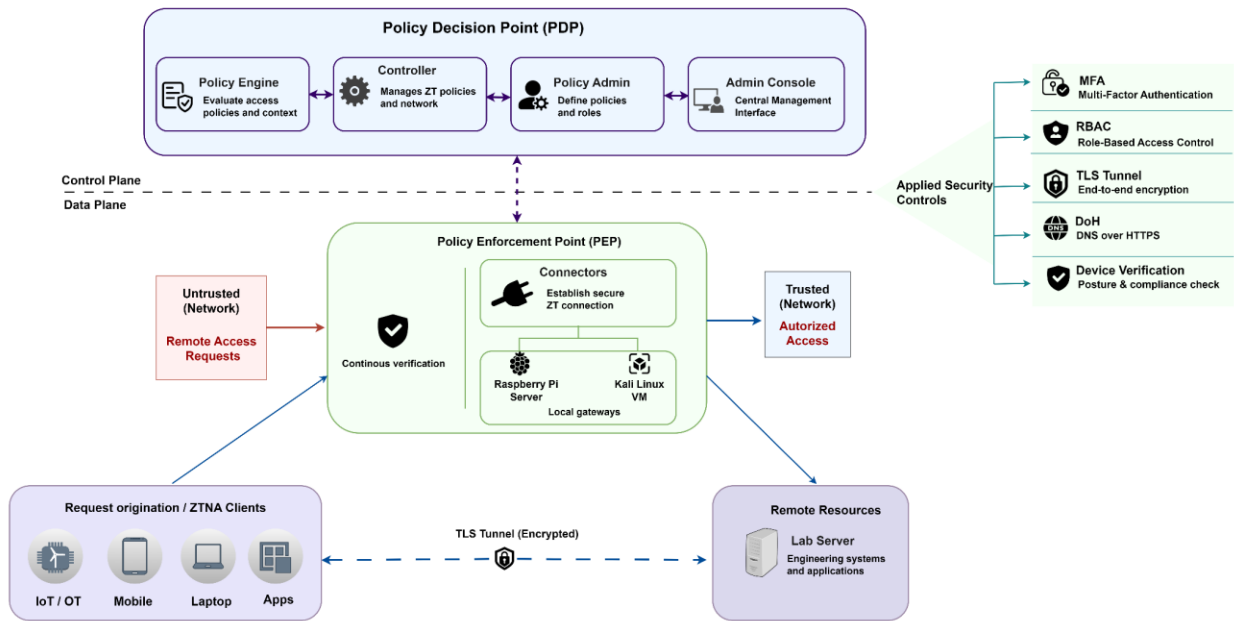


Figure 3.1. Logical architecture of the proposed Zero Trust access control framework.

3.1.2 ZTNA Architecture, Access Control, and Device Security Policy

In the proposed Zero Trust Network Access (ZTNA) architecture, protected resources are organized into Remote Networks, which abstract the underlying infrastructure (e.g., on-premises or VPC environments) and enable secure, policy-controlled access via the Twingate platform. Access is centrally managed through the controller, while Connectors serve as policy enforcement points, establishing encrypted outbound tunnels to internal resources without exposing the network perimeter. A multi-connector setup (Raspberry Pi and VM-based) enhances availability, load balancing, and fault tolerance.

Resources are defined using IP-based addressing or Classless Inter-Domain Routing (CIDR) ranges and remain inaccessible directly from end-user devices. Users access them through authenticated sessions initiated via a client application, following invitation-based onboarding and identity verification.

As illustrated in Figure 3.2, access control is enforced through identity-centric authentication and authorization mechanisms, including integration with external identity providers (Google SSO), Role-Based Access Control (RBAC), and Multi-Factor Authentication using TOTP for privileged roles. Access decisions follow the least privilege principle, ensuring users interact only with authorized resources under defined conditions, thereby minimizing lateral movement and attack surface.

Additionally, device-level security policies are enforced before access, requiring compliance with predefined security standards (encryption, firewalls, antivirus, and system integrity checks). Communication between clients and resources is secured through certificate-pinned TLS tunnels and DNS-over-HTTPS (DoH), ensuring confidentiality, integrity, and protection against interception or spoofing. Continuous monitoring and logging of access activity further support threat detection and response.

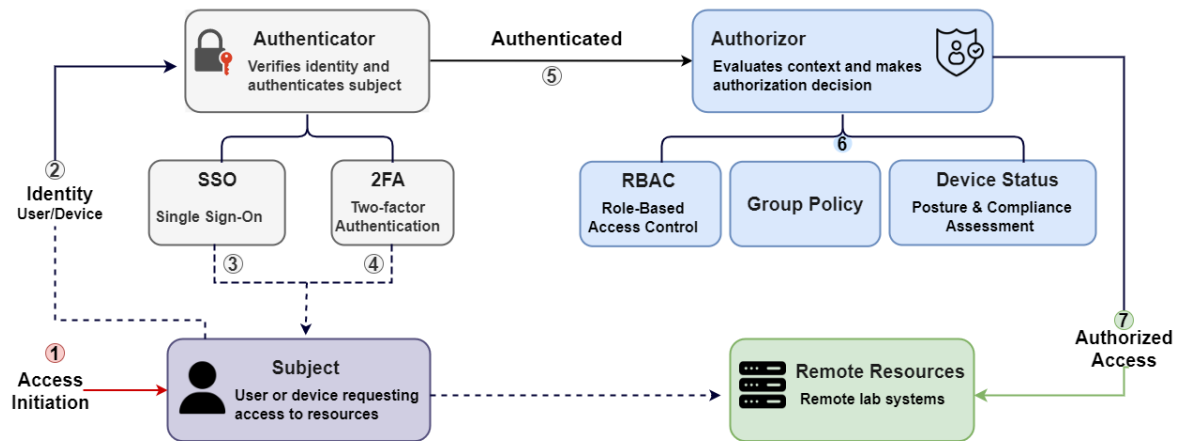


Figure 3.2. High-level access control workflow, from client initiation to resource authorization.

3.2. Discussions

The access control architecture presented in this chapter demonstrates a fundamental shift from perimeter-based remote access to a resource-centric ZT model. Unlike VPNs, which grant broad network access after authentication and enable lateral movement, ZTNA applies continuous verification, identity-aware policies, and least-privilege principles, eliminating implicit trust and restricting access to explicitly authorized resources.

The security implications of this shift are illustrated in Figure 3.3, which presents a stolen-credentials attack scenario. In VPN-based architectures, compromised credentials typically allow unrestricted network access and lateral movement. In contrast, the ZT model enforces resource-level access decisions based on identity, device posture, and policy, denying access to unauthorized devices and limiting the scope of access, thereby containing the attack and reducing its impact.

Beyond security, ZTNA improves operational scalability and manageability by replacing network-centric configurations with centralized, policy-driven access control. This reduces administrative overhead, simplifies user provisioning, and enhances adaptability in dynamic environments such as remote laboratories. Additionally, by limiting lateral movement, ZTNA reduces the potential impact and cost of security incidents.

However, ZTNA does not inherently detect malicious activity within authorized sessions. Insider threats, compromised identities, or misuse of legitimate privileges may remain undetected, particularly in cyber-physical systems where authorized actions can have physical consequences. Therefore, while ZT provides a strong access-control foundation, it must be complemented by monitoring and intrusion-detection mechanisms, which motivates the subsequent chapters.

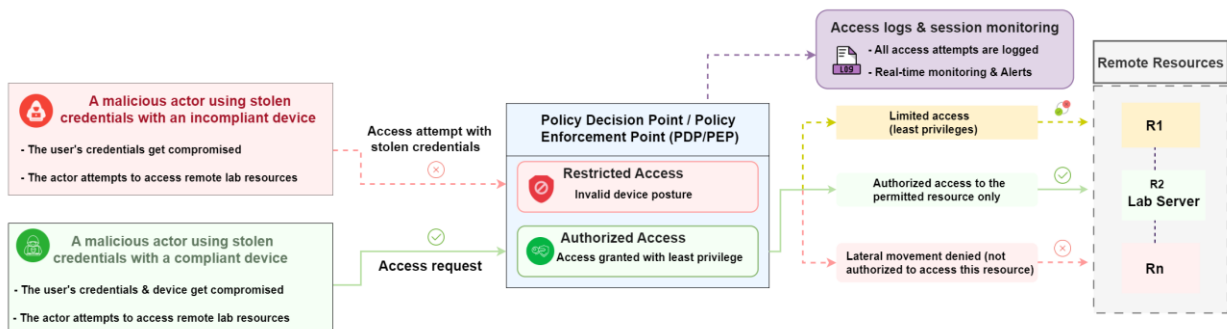


Figure 3.3. Containment of a stolen-credentials attack under a Zero Trust architecture, illustrating how policy-based enforcement (PDP/PEP) and device posture validation restrict access and prevent lateral movement within remote engineering resources.

3.3. Chapter Conclusion

This chapter investigated the problem of secure remote access in remote engineering systems, with particular emphasis on educational online laboratories as representative cyber-physical environments. The analysis demonstrated that traditional perimeter-based security mechanisms, most notably VPN-centric remote access models, introduce structural limitations when applied to shared remote engineering infrastructures.

Building on NIST's ZT framework [154], the chapter proposed and implemented an identity-centric, fine-grained resource-level access control architecture that eliminates implicit network trust, enforces least-privilege access, and constrains lateral movement within the laboratory infrastructure. The practical deployment in an online serial laboratory environment confirmed that ZT access control can be integrated into real-world remote engineering platforms without exposing internal network topology or granting excessive privileges, while preserving usability and operational continuity.

A comparative analysis highlighted the advantages of the ZT approach over traditional VPN-based access in terms of access granularity, reduction of lateral movement, and improved auditability. At the same time, the chapter acknowledged the limitations of access-control-only solutions, particularly their inability to detect malicious or anomalous behavior originating from authorized users or from compromised identities.

Therefore, this chapter fulfills objective O2 by designing, implementing, and validating an identity-centric, fine-grained, resource-level access-control architecture tailored to online laboratory environments. Accordingly, it establishes ZT access control as a foundational security layer for remote engineering systems, demonstrating its effectiveness in mitigating risks associated with perimeter-based access models. However, access control alone cannot address post-compromise threats or anomalous behavior within trusted sessions. These limitations motivate the integration of complementary detection and monitoring mechanisms to identify abnormal activity in cyber-physical environments. The subsequent chapters, therefore, extend the security framework by developing resource-efficient intrusion and anomaly-detection techniques tailored to IoT/IIoT-based remote engineering systems.

Chapter 4. Feature Selection–Driven Resource-Efficient Machine Learning-Based Traffic Classification for MQTT-IoT Security Attacks

This chapter proposes a lightweight FS method, Statistical Moments Difference Thresholding (SMDT), that selects discriminative features based on differences in statistical moments (mean, standard deviation, skewness, kurtosis). Evaluated on the MQTTset dataset [100], the method demonstrates that significant feature reduction can be achieved while maintaining high detection performance. Although limited to offline evaluation, this work establishes the methodological foundation for subsequent real-time deployment and integration in IoT/IIoT systems.

4.1. Methodology and Experimental Design

This section describes the experimental design and methodological pipeline used to investigate resource-efficient malicious-traffic classification in MQTT-based IoT environments. The adopted methodology follows a structured process encompassing dataset selection, data preprocessing and exploratory analysis, feature selection, and ML-based classification. The overall feature selection and classification framework adopted in this chapter is illustrated in Figure 4.1.

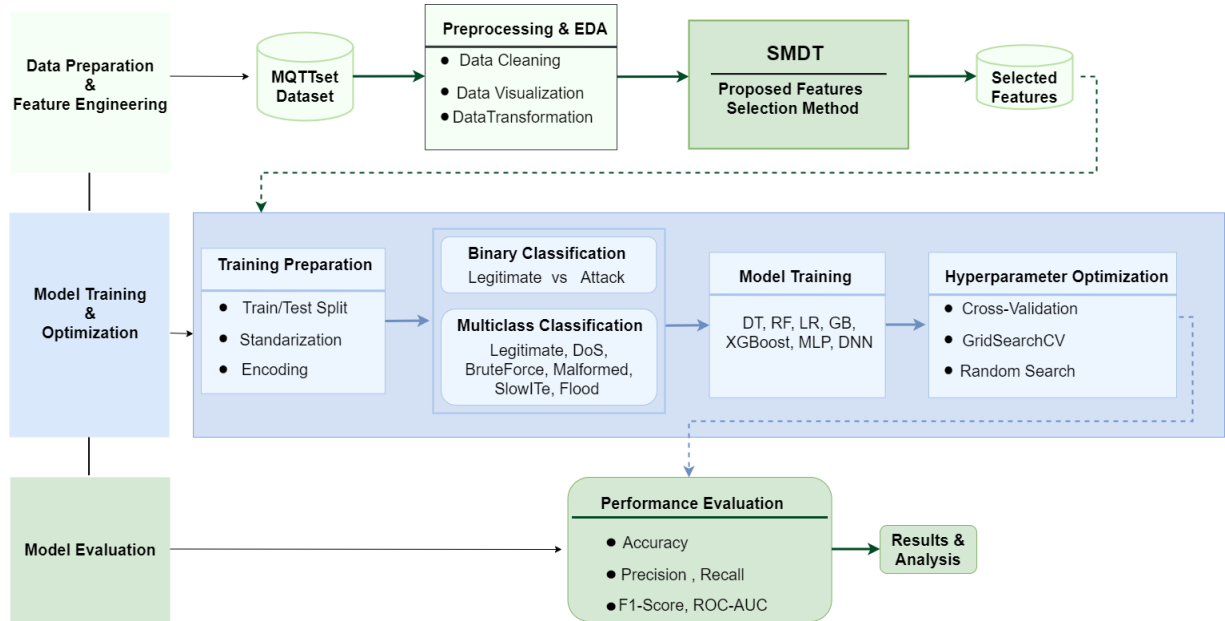


Figure 4.1. Conceptual overview of the traffic classification framework adopted in this chapter, highlighting SMDT-based dimensionality reduction and subsequent binary and multiclass classification pipeline.

4.1.1 Dataset Description

This study employs MQTTset, a publicly available IoT dataset specifically designed to capture MQTT communications, as described in [101] and made available online in [100]. MQTTset was generated using an IoT sensor network intended to replicate key characteristics of real-world MQTT-based deployments. The infrastructure includes an MQTT broker implemented using Eclipse Mosquitto v1.6.2 and a network of eight interconnected sensors, namely temperature, light intensity, humidity, motion, CO-gas, smoke, fan-speed controller, and door-lock sensors, coordinated through the broker. The resulting dataset contains both legitimate (benign) traffic and malicious (attack) traffic, enabling supervised learning for intrusion detection in MQTT-enabled IoT environments. MQTTset incorporates multiple attack scenarios targeting MQTT networks [101]. The attack classes and traffic distribution are depicted in Figure 4.2.

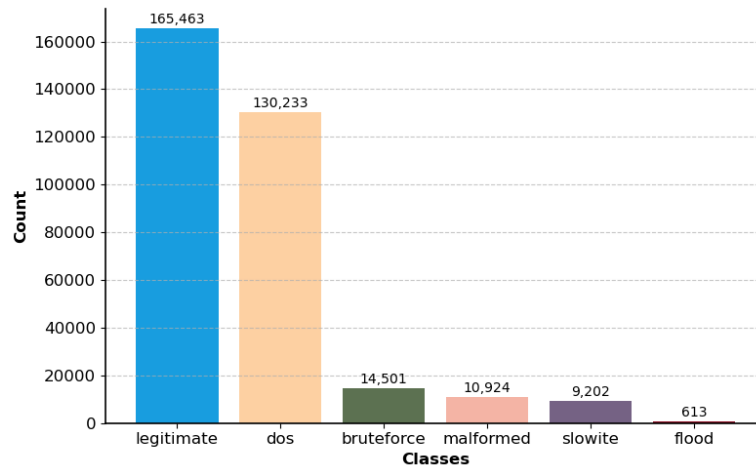


Figure 4.2. Class distribution of the balanced reduced MQTTset dataset—benign vs attack types.

4.1.2 Feature Selection

FS constitutes a critical stage in the ML pipeline, aiming to identify a subset of informative features that contribute most effectively to predictive performance while reducing redundancy and noise [102]. By eliminating irrelevant or weakly informative attributes, FS mitigates [103] overfitting, reduces computational complexity, and addresses generalization challenges that are particularly pronounced in high-dimensional datasets, commonly described as the “curse of dimensionality.”[104] These considerations are especially relevant in IoT and IIoT environments, where resource constraints necessitate lightweight yet discriminative detection mechanisms.

To address these challenges, this study introduces the SMDT method. The SMDT approach leverages the first four statistical central moments [105], mean, standard deviation, skewness, and kurtosis to quantify distributional differences between legitimate and malicious traffic. These moments were selected due to their complementary ability to capture central tendency, dispersion, asymmetry, and tail behavior of feature distributions [106]. Such multidimensional statistical characterization is particularly appropriate for IoT traffic, which frequently exhibits heterogeneous, non-Gaussian, and highly skewed distributions.

4.2. Experimental Design and Evaluation

4.2.1 Binary Classification Setup and Results

4.2.1.1 Model Training and Selection

The binary classification task was defined to distinguish MQTT traffic as either legitimate (0) or an attack (1), using the balanced reduced MQTTset dataset to address class imbalance. Features were normalized using Z-score standardization (StandardScaler) to improve model convergence and performance [107], [108], and the dataset was split into training and testing sets (70/30) using stratified sampling to preserve class distribution [109].

Model optimization was performed using GridSearchCV with five-fold cross-validation ($k = 5$) to systematically tune hyperparameters and reduce overfitting. After identifying optimal configurations, models were retrained on the full training set.

For the DNN model, hyperparameter tuning was conducted using Keras Tuner with Random Search, optimizing parameters such as dropout and learning rate [110]. The network architecture consisted of three hidden layers (64, 32, 16 neurons) with ReLU activation and dropout for regularization, and a sigmoid output layer for binary classification [111]. Training was performed using the Adam optimizer and binary cross-entropy loss [112], followed by retraining with the optimal configuration for 10 epochs.

4.2.1.2 Performance Evaluation

The performance of the trained models was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. To gain deeper insight into prediction behavior, confusion matrices were analyzed for all models. The confusion matrix reports the distribution of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The consolidated performance metrics for all evaluated models are summarized in Table 4.1. The searching time (s) reflects the duration of the hyperparameter optimization process, including GridSearchCV for conventional models and Random Search via Keras Tuner for the DNN. Additionally, Figure 4.3 presents the confusion matrices for all models.

Table 4.1. Binary classification performance metrics and computational times.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Searching Time (s)	Training Time (s)	Testing Time (s)
RF	0.95	0.95	0.95	0.95	0.99	2280.81	6.96	0.45
LR	0.95	0.95	0.95	0.95	0.99	221.58	11.27	0.99
DT	0.95	0.95	0.95	0.95	0.99	74.98	0.12	0.01
GB	0.95	0.95	0.95	0.95	0.99	3051.20	43.08	0.45
XGBOOST	0.95	0.95	0.95	0.95	0.99	1252.83	12.38	0.04
MLP	0.89	0.91	0.89	0.89	0.95	3551.45	264.61	0.20
DNN	0.81	0.85	0.81	0.80	0.92	5602.99	722.57	6.44

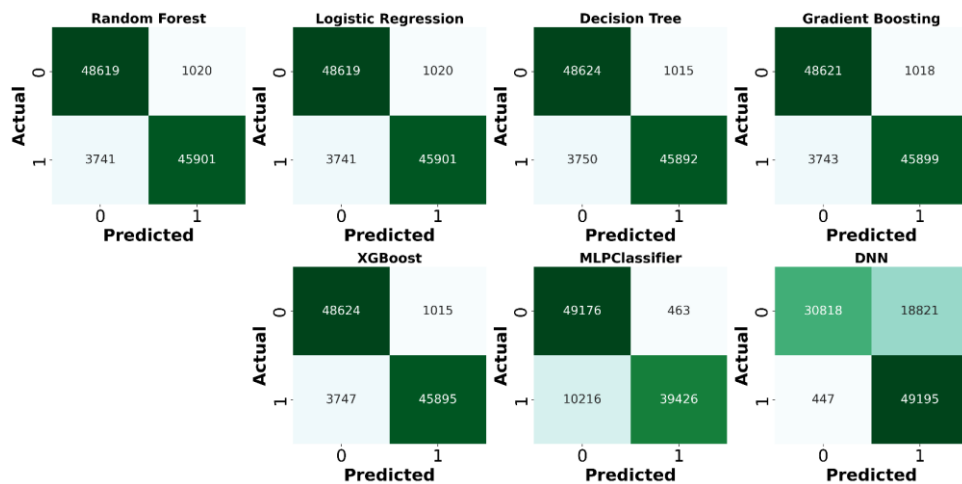


Figure 4.3. Confusion matrices for all evaluated models (binary classification).

4.2.2 Multiclass Classification Setup and Results

Following the binary classification experiments, which focused on distinguishing legitimate from malicious MQTT traffic, the analysis was extended to a multiclass classification setting. This extension enables the identification of specific attack categories: brute force, DoS, flooding, malformed data, and SlowITe. This granularity supports both precise threat characterization and targeted mitigation strategies. In this setting, the classification task involves six distinct classes, compared to the two-class formulation adopted in the binary scenario.

Model performance was evaluated using metrics adapted from the binary classification case. Specifically, the evaluation comprised the following components:

- a) Classification report: Accuracy, precision, recall, and F1-score were computed for each model. The aggregated results are reported in Table 4.2. The searching time presented in the table denotes the time required for hyperparameter optimization before final model training.

Table 4.2. Model Performance: Evaluation Metrics and Searching Times (Multiclass).

Model	Accuracy	Precision	Recall	F1-Score	Searching Time (s)	Training Time (s)	Testing Time (s)
RF	0.91	0.90	0.91	0.90	1547.43	16.21	1.15
LR	0.79	0.79	0.79	0.75	1834.67	8.00	0.01
DT	0.91	0.91	0.91	0.90	9.24	0.12	0.01
GB	0.91	0.91	0.91	0.90	24489.46	505.96	4.80
XGBOOST	0.91	0.91	0.91	0.90	49323.92	15.75	0.06
MLP	0.82	0.85	0.82	0.80	3573.60	141.93	0.11
DNN	0.78	0.77	0.78	0.74	5081.92	702.20	6.03

- b) Confusion matrix: To analyze prediction behavior across all attack categories, confusion matrices were generated for each model. In the multiclass setting, the confusion matrix expands to an $N \times N$ grid, where N corresponds to the number of classes. Each cell represents the number of instances associated with a given actual/predicted class pair, allowing for detailed inspection of classification patterns. For conciseness, Figure 4.4 presents the confusion matrix for the RF model only.

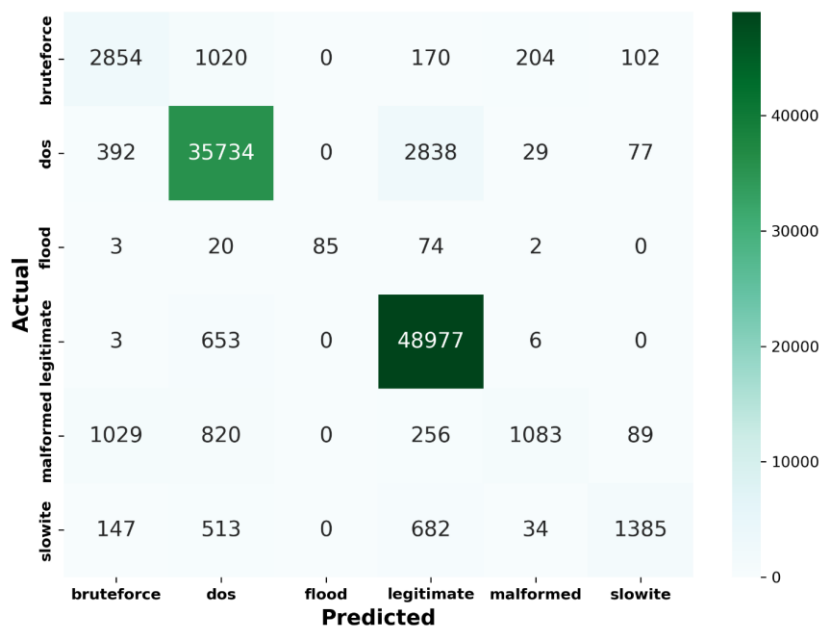


Figure 4.4. Confusion matrix—Random Forest (multiclass).

4.3. Discussion of Results

For binary classification, classical ML models (DT, RF, GB, XGBoost) exhibited strong and consistent performance, achieving accuracies, precision, recall, and F1 Scores of approximately 0.95, with ROC AUC values approaching 0.99 (see Table 4.1). In contrast, neural network-based models demonstrated comparatively lower performance. The MLP achieved moderate accuracy (0.89), while the Deep Neural DNN achieved an accuracy of 0.81 and the ROC-AUC of 0.92.

Additionally, the confusion matrices (see Figure 4.3) provide additional insight into the error characteristics of the evaluated models. Although the DNN exhibited lower overall accuracy, it demonstrated high sensitivity, correctly identifying a large proportion of attack instances (49,195) and resulting in fewer false negatives (447). This behavior, however, was accompanied by a substantially elevated number of false positives (18,821), indicating limited specificity and a tendency to generate excessive false alarms, as further reflected in Figure 4.7. Such behavior may be undesirable in real-time operational environments, where frequent false alerts can lead to inefficient resource utilization and alert fatigue. In contrast, the MLP displayed the opposite error profile, producing fewer false positives (463) but more false negatives (10,216). While this reduces unnecessary alerts, it increases the risk of missed attacks. These results highlight the importance of balancing detection sensitivity and operational reliability in real-time deployments.

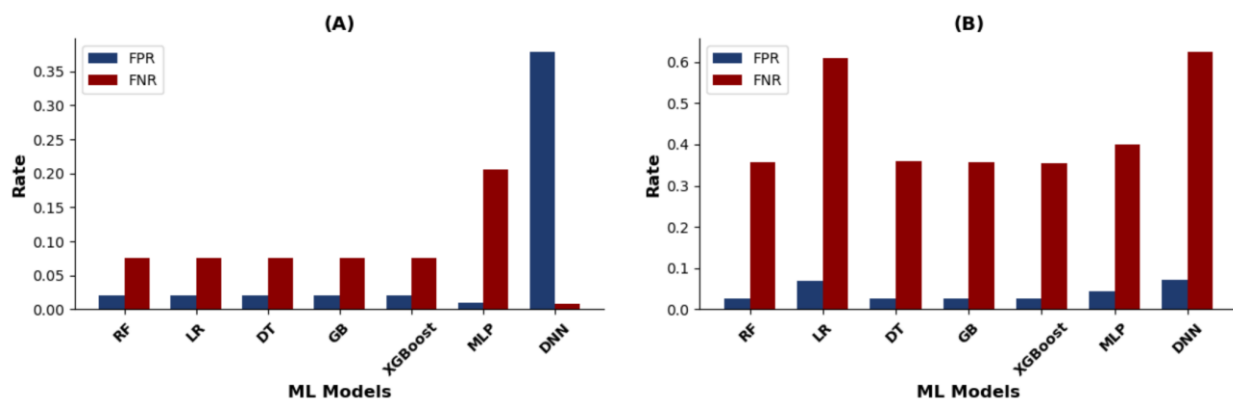


Figure 4.5. FPR and FNR across ML models: (A) Binary classification case and (B) multiclass classification case.

In the multiclass scenario, performance degraded across all models due to class imbalance and increased task complexity. Ensemble and tree-based models (RF, DT, GB, and XGBoost) maintained relatively stable performance, with accuracy and F1 Scores around 0.90 (see Table 4.2), whereas LR, MLP, and DNN showed lower effectiveness. Confusion matrices (see Figure 4.5) indicate that most misclassifications occurred in minority attack classes, such as malformed, SlowITe, and flooding, suggesting that rare-class discrimination remains a challenge even after feature reduction.

Training and optimization time varied substantially across models (see Figure 4.6). Tree-based methods were consistently the most efficient, while MLP and DNN incurred significantly higher processing overhead.

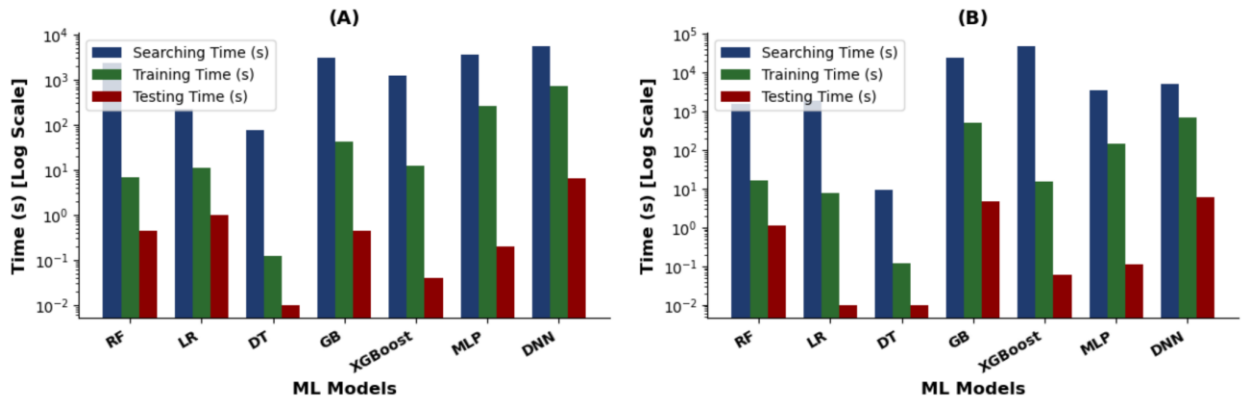


Figure 4.6. Computational time across ML models: (A) Binary classification case and (B) multiclass.

4.4. Model Selection and Practical Implications

From a deployment perspective, the results indicate that the DT and RF models offer the best trade-off between detection performance and computational efficiency in resource-constrained IoT-based remote engineering environments. DT, in particular, offers low training and inference latency, making it suitable for real-time monitoring. While ensemble approaches such as GB and XGBoost achieve strong detection performance, their higher computational requirements make them more appropriate for less constrained environments. In contrast, MLP and DNN models present lower performance-to-cost efficiency for online deployment. Overall, model selection should consider not only accuracy but also latency, resource usage, and error trade-offs, with tree-based models demonstrating a balanced profile suitable for practical IoT intrusion detection.

4.5. Chapter Conclusion

This chapter addressed malicious traffic detection in MQTT-based IoT/IIoT remote engineering systems, highlighting the limitations of access-control-only defenses and the need for complementary intrusion detection mechanisms. The chapter contributes to the objective O3 by developing and validating a feature-selection-driven traffic classification framework tailored for constrained MQTT-based environments. A resource-efficient ML framework was proposed, centered on the SMDT feature selection method, which reduces feature dimensionality while preserving detection performance.

Experimental results on the MQTTset dataset showed that reducing the feature set from 33 to 5 maintains competitive performance for both binary and multiclass classification, particularly with lightweight models such as Decision Trees and Random Forests. The analysis also revealed key trade-offs, indicating that simpler models combined with efficient feature selection provide a better balance between accuracy and computational cost than more complex approaches in resource-constrained environments.

However, the evaluation was limited to offline conditions and did not consider real-time constraints or system integration. Therefore, this chapter establishes the foundation for Chapter 5, which extends the work toward real-time deployment and operational evaluation in IoT/IIoT environments.

Chapter 5. Development of a Modular Architecture for Real-Time ML-Driven Intrusion Detection in IoT/IloT with Privacy-Aware Cloud SIEM Integration

This chapter proposes a lightweight, privacy-aware, ML-driven IDS framework tailored for IoT/IloT remote monitoring and control environments. The architecture emphasizes edge-based inference for real-time detection, strict data minimization by retaining raw traffic locally, and selective transmission of structured alerts to a cloud-based SIEM. Additionally, it introduces a dynamic runtime model-switching mechanism that enables flexible adaptation to evolving threats without interrupting system operation. The framework is validated through simulated MQTT-based attack scenarios, demonstrating its feasibility for real-time deployment.

5.1. System Architecture Design

This section presents the architecture and implementation of the proposed ML-driven intrusion detection system for IoT/IloT remote monitoring and control environments. The system adopts a modular, layered design that supports sensor data acquisition, real-time edge machine learning inference, and secure forwarding of detection results to a cloud-based SIEM platform, namely Microsoft Sentinel. As illustrated in Figure 5.1, the deployed system comprises multiple functional layers, each responsible for a distinct set of operations.

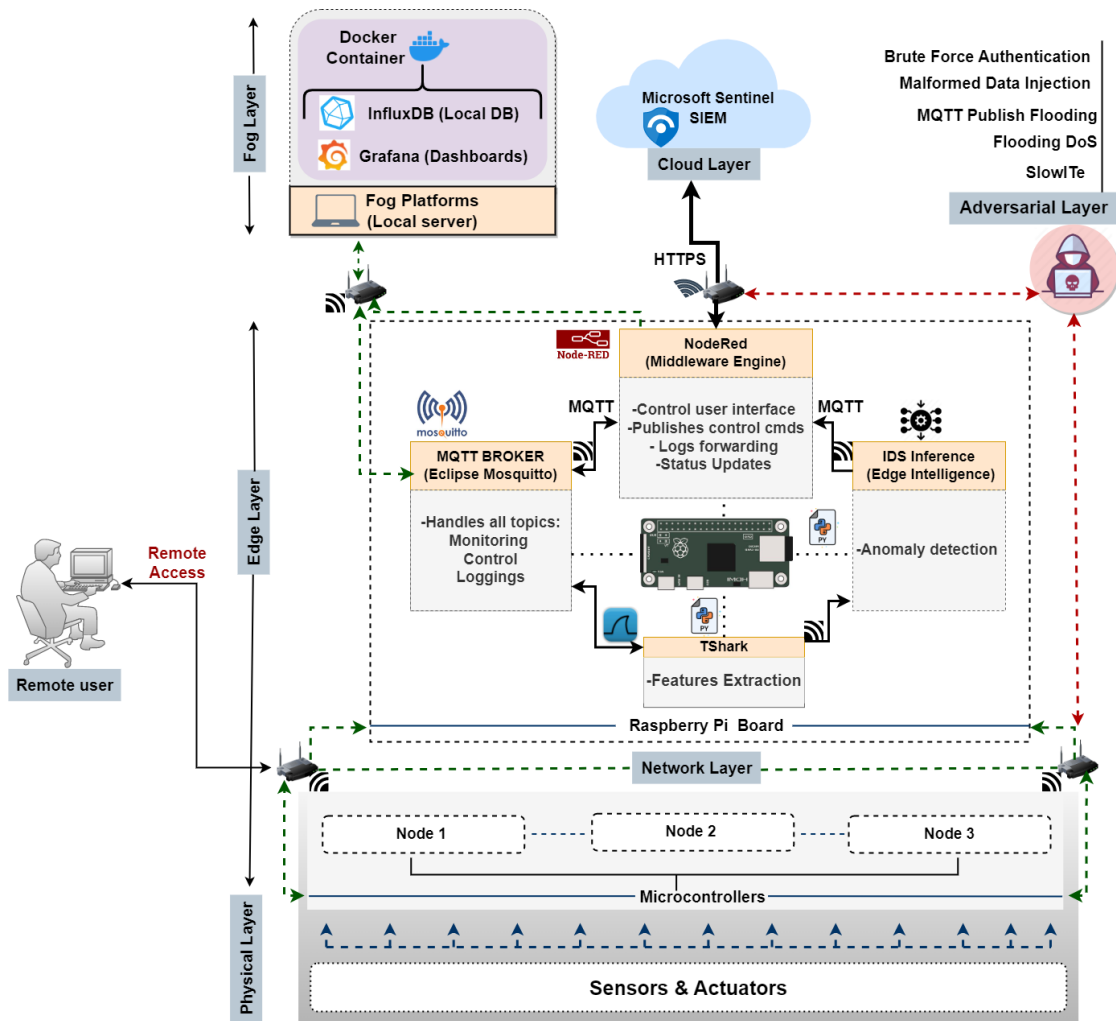


Figure 5.1. Overall architecture of the proposed Edge-Cloud IDS testbed.

5.2. System Platforms Integration and Communication

5.2.1 Sensor-Level Data Acquisition

The physical layer consists of distributed sensors across multiple microcontroller nodes to support environmental and security monitoring. As shown in Figure 5.2, an Arduino-based node collects data from ultrasonic, vibration, and relay components and forwards it via serial communication to an ESP32 node, which aggregates and publishes the data to MQTT topics. A second ESP32 node independently handles additional sensors (flame, gas, temperature/humidity) and actuators, enabling bidirectional communication through control topics for closed-loop operation. All ESP32 nodes use a standardized MQTT configuration, transmitting data periodically (every 10 seconds) and through event-driven updates.

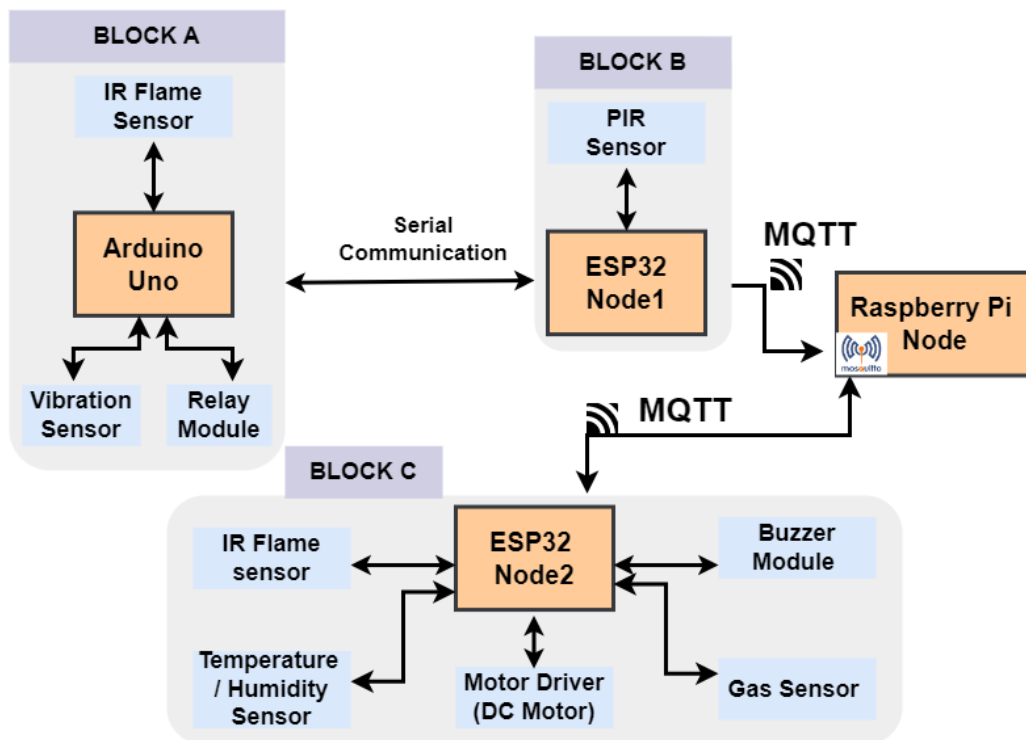


Figure 5.2. Sensor-level data acquisition and communication architecture.

5.2.2 Node-RED-Based Middleware and Integration Layer

Node-RED is used as the middleware layer to orchestrate communication, data processing, and integration within the system and is deployed on a Raspberry Pi edge node. It enables real-time, event-driven workflows for data acquisition, transformation, and interaction with external services.

The platform performs two main functions: (i) aggregating and monitoring sensor data, and (ii) integrating ML inference outputs and security telemetry. It subscribes to MQTT topics, processes and formats incoming data. It forwards them to InfluxDB for storage and to a cloud-based SIEM for analysis, with visualization provided through Grafana dashboards.

Additionally, Node-RED supports bidirectional device control, enabling both automated actions based on sensor inputs and manual control via a web interface. A feedback mechanism using MQTT status topics ensures synchronization between commands and actuator states, improving system reliability and real-time operational awareness.

5.2.3 Privacy-Preserving Cloud-Integration

The system employs a hybrid privacy-preserving logging strategy in which all raw traffic data (e.g., packet payloads, device identifiers, and operational parameters) remain at the edge. At the same time, only structured alerts and minimal metadata are transmitted to the cloud SIEM. This approach reduces data exposure risks, supports confidentiality, and minimizes bandwidth usage, aligning with data minimization principles such as GDPR [113] and NIST guidelines [114].

Node-RED processes ML inference outputs from the “model/prediction” MQTT topic and forwards only compact JSON-formatted metadata, including timestamp, predicted attack class, model version, and selected contextual attributes. No raw packets, features, or sensor data are transmitted, ensuring that sensitive operational data remains local.

As illustrated in Figure 5.3, telemetry is securely transmitted to Microsoft Sentinel via the Azure Monitor HTTP Data Collector API using TLS-encrypted HTTPS and HMAC-SHA256 authentication. Once ingested, data are stored in Azure Log Analytics, enabling KQL-based querying, visualization, and automated incident response.

This privacy-preserving integration strategy provides several practical advantages. The system remains fully operational at the edge during cloud connectivity disruptions, ensuring uninterrupted detection capabilities. At the same time, compliance with data minimization principles supports regulatory requirements, while reduced log volume lowers cloud storage and processing costs.

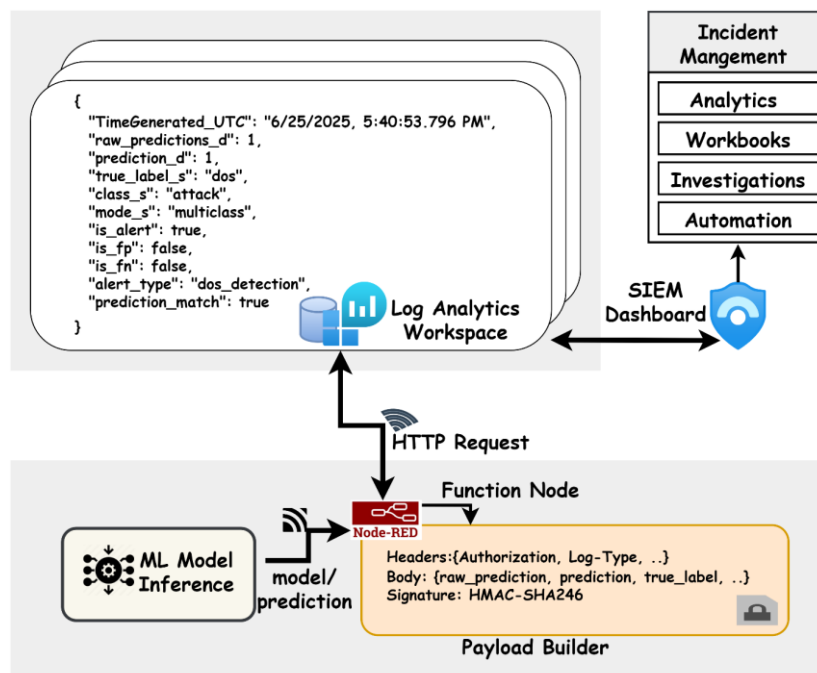


Figure 5.3. Architecture for forwarding edge-based detection results to the Cloud SIEM.

5.2.4 Threats Model and Attack Scenarios

This chapter defines a threat model targeting MQTT-based IoT/IIoT systems deployed in insecure or weakly segmented network environments. In the adversarial model considered, an attacker gains access to the same network segment as legitimate devices and exploits vulnerabilities in MQTT communication mechanisms to degrade system performance, disrupt data delivery, or compromise confidentiality and availability. This threat model reflects realistic attack vectors commonly observed in industrial and IoT deployments that lack strong access control, network segmentation, or continuous monitoring. Figure 5.4 depicts simulated attacks and the utilized tools.

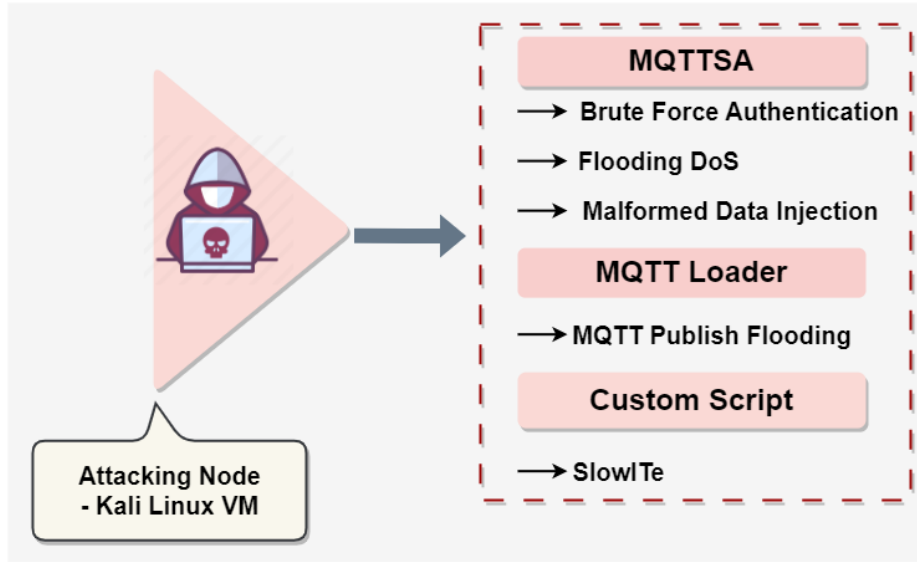


Figure 5.4. Simulated attacks and utilized tools.

5.3. Machine Learning Inference

The inference represents the stage in the machine learning lifecycle in which a trained model is deployed to generate predictions on new, unseen data [115]. In this chapter, models trained offline on the MQTTset dataset (Chapter 4) are deployed within a real-time edge-based detection pipeline.

To balance accuracy and computational efficiency, lightweight DT classifiers were selected for deployment on resource-constrained edge devices. At the same time, more complex models (e.g., ensemble and neural approaches) are reserved for higher-capacity environments. The inference engine is implemented as a lightweight script running on a Raspberry Pi Zero 2 W (512 MB of RAM), in which pre-trained models are serialized with Python’s pickle and loaded at runtime.

As depicted in Figure 5.5, the system subscribes to the “traffic/analysis” MQTT topic, processes incoming feature vectors aligned with the training schema, and performs real-time classification. Predictions are published to the “model/prediction” topic, stored locally in InfluxDB for logging and forensic analysis, and formatted for transmission to the cloud SIEM, enabling centralized monitoring while maintaining edge-based processing.

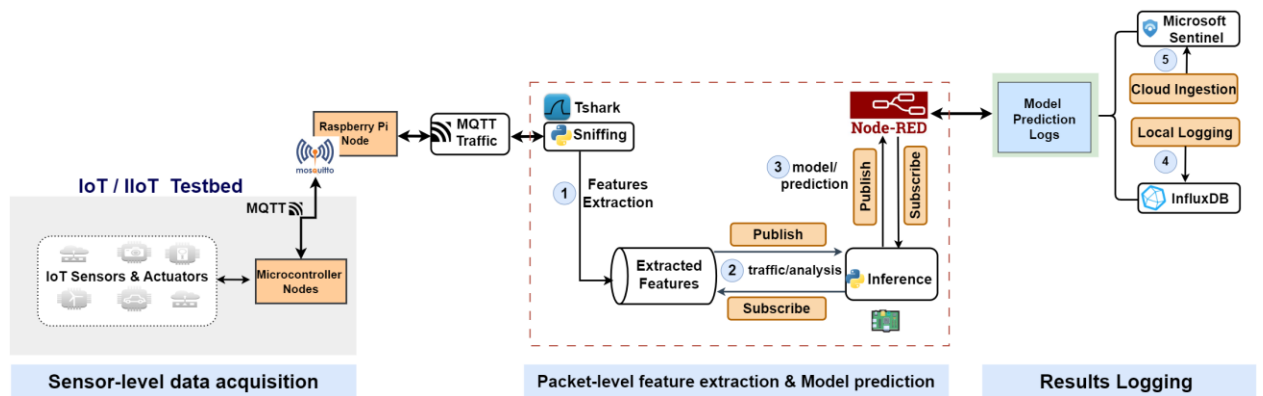


Figure 5.5. End-to-end real-time inference and alert forwarding workflow.

5.3.1 Runtime Model Switching Mechanism for Adaptive IDS Inference

Operationalizing ML models in production requires mechanisms for monitoring, updating, and replacing models without disrupting system operation[116]. To address this, the proposed IDS incorporates a lightweight runtime model-switching mechanism, aligned with adaptive ML and intrusion-detection paradigms in which models evolve in response to changing data and threat patterns [116], [117].

The mechanism enables dynamic model updates via MQTT-based control messages. As illustrated in Figure 5.6, an authorized operator can trigger a model switch by sending a structured JSON command specifying the target model and configuration parameters. Upon validation, the inference engine loads the new model at runtime and updates relevant settings, including classification mode (binary or multiclass), label mappings, and logging configurations.

This approach provides operational flexibility by enabling seamless transitions between detection modes, the activation of specialized models for specific attack types, and the deployment of newly retrained models without interrupting monitoring. It also supports rollback to previously validated models in case of performance issues. Overall, the design decouples the model lifecycle from the monitoring pipeline, enabling continuous adaptation while maintaining system stability and availability.

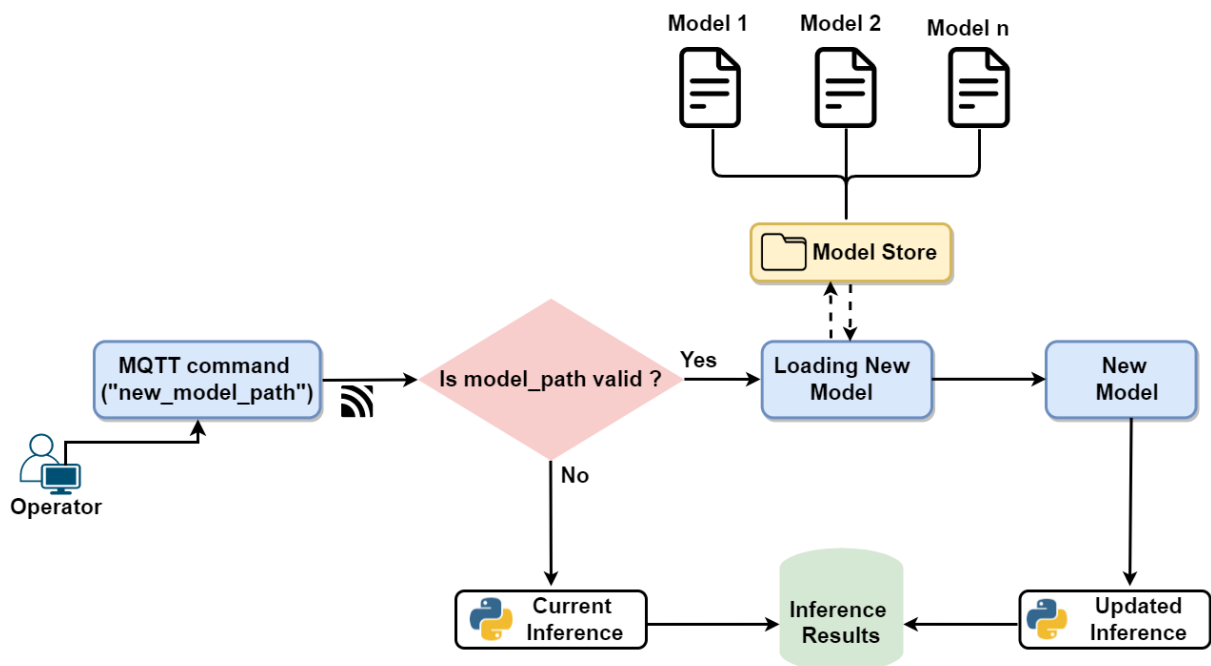


Figure 5.6. Runtime model-switching workflow enabling dynamic deployment of alternative IDS inference models through MQTT-based control commands.

5.4. Results, Performance Evaluation, and System-Level Analysis

5.4.1 Detection Performance and Accuracy Analysis

The edge-based inference system achieved an average classification accuracy of 74.04% over a 30-day observation period in multi-attack classification. This represents a notable reduction compared to the 91% accuracy obtained during offline training.

This performance gap reflects the distributional differences between training and deployment environments and highlights the impact of concept drift and evolving traffic patterns in real-world IoT/IIoT systems. Limited representativeness of the training dataset with respect to operational variability may also contribute to this degradation, reducing the transferability of learned decision boundaries [118].

As results demonstrate, the offline evaluation cannot fully capture the complexity of live deployments. Sustaining long-term detection accuracy in live environments requires adaptive learning and periodic model retraining, which are considered directions for future work.

5.4.2 Latency and Throughput Performance

Latency measurements confirmed the feasibility of deploying machine-learning inference at the edge. On average, the system achieved an inference latency of 9.32 ms, while the overall end-to-end latency, including cloud offloading, was 22.62 ms.

As illustrated in Figure 5.7, a maximum latency of 50.9 ms was observed during periods of high traffic intensity. These peaks were primarily attributed to CPU contention on the Raspberry Pi Zero 2 W.

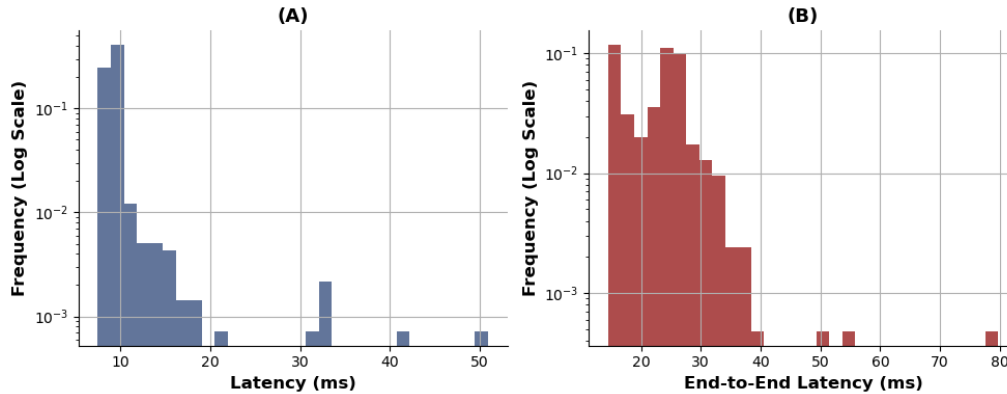


Figure 5.7. Distribution of latency (A) and End-to-End latency (B) for the deployed inference system.

Throughput analysis showed that the system sustained an average processing rate of 4.95 predictions per second. Figure 5.8 illustrates throughput variability, with occasional drops to 1.24 predictions per second and peaks of 13.43 predictions per second during intensive attack scenarios. Flooding and malformed packet attacks generated the highest processing loads, leading to transient reductions in responsiveness.

The latency and throughput results demonstrate that lightweight edge-based inference is viable for moderate traffic volumes. However, high-throughput industrial environments might need horizontal scaling or hardware acceleration to maintain consistent performance.

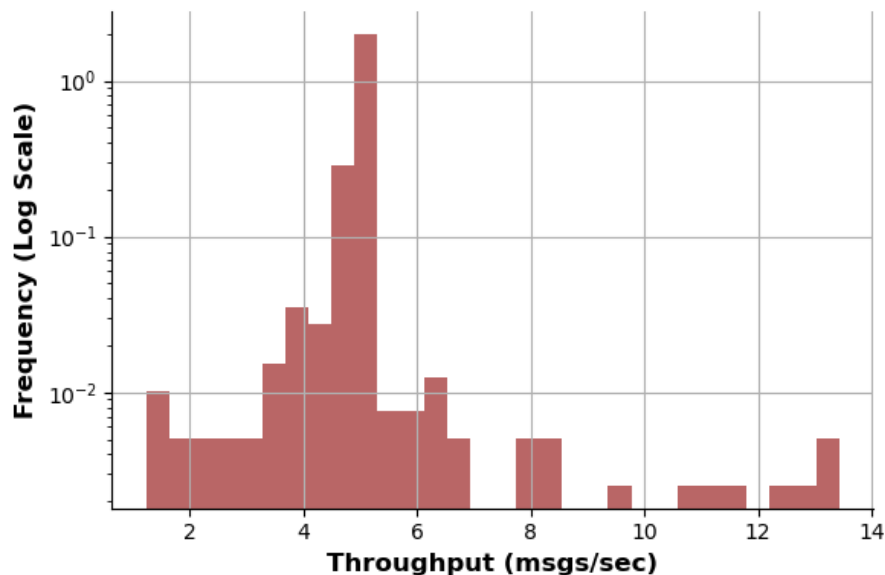


Figure 5.8. Distribution of throughput for the deployed inference system.

5.4.3 Runtime Adaptation and Model Switching Overhead

To evaluate the overhead introduced by the runtime model switching mechanism, the latency required to load and activate new models during execution was measured on the edge node.

Swap latency ranged from 2.28 ms to approximately 200 ms, reflecting variations in model size and resource availability on the Raspberry Pi platform. Importantly, the switching process remained non-blocking. Once the new model was loaded, inference resumed without requiring service restarts or sustained downtime.

5.4.4 Cloud Integration, Operational Validation, and Cost Efficiency

A. Cloud Integration and Operational Validation

Classification outcomes produced by the inference engine were published to a local MQTT topic and forwarded by Node-RED to Azure Log Analytics for ingestion into Microsoft Sentinel. This hybrid integration preserves the benefits of cloud-centric SIEM-based monitoring while remaining compatible with the operational constraints of resource-constrained and privacy-sensitive IoT/IloT environments.

Within the SIEM platform, ingested logs were stored in custom tables, enabling analysts to define detection rules, correlate alerts with external threat intelligence, and visualize security events through interactive dashboards. As depicted in Figure 5.9, Sentinel aggregates prediction logs over time and automatically generates incidents for high-confidence attack scenarios. The incident management interface serves as the primary entry point for analysts, providing consolidated visibility into alerts and affected entities. This interface facilitates rapid investigation and supports efficient assessment of potential security breaches. Furthermore, Microsoft Sentinel workbooks provide quantitative and interactive visualizations that enhance operational visibility and support security analysis. These workbooks enable analysts to examine historical trends, attack distributions, and temporal variations in detection outcomes.

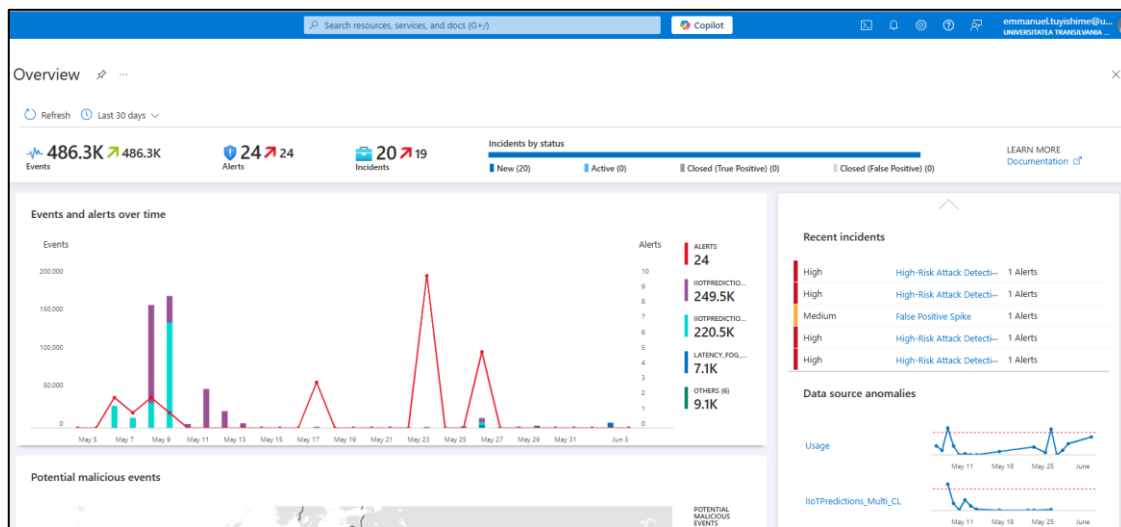


Figure 5.9. Incident overview and alert aggregation in Microsoft Sentinel based on edge-level inference logs.

B. Cost Efficiency and Cloud Resource Utilization

The proposed framework was evaluated for cloud resource consumption under Microsoft Sentinel's volume-based billing model, in which costs scale with data ingestion, processing, and retention. To reduce operational expenditure, the architecture adopts a selective telemetry approach, transmitting only compact prediction metadata to the cloud while retaining detailed logs locally in InfluxDB.

During a one-month evaluation, the system processed approximately 4.95 predictions per second, with each record averaging approximately 500 bytes. This resulted in an estimated daily ingestion volume of approximately 204 MB (6.1 GB per month). Based on Sentinel pricing (€4.47/GB) [119], the corresponding monthly cost is approximately €27.

In contrast, conventional cloud-centric IDS solutions transmitting flow- or packet-level data (2–5 KB per event) would generate 24–61 GB per month, incurring costs of approximately €109–€272. Thus, the

proposed approach achieves a 75–90% reduction in ingestion volume, significantly improving cost efficiency. This reduction enables more scalable and economically sustainable deployments, as cloud costs grow more slowly with system expansion because only inference results are transmitted, rather than raw or high-volume telemetry data.

5.5. Chapter Conclusion

This chapter demonstrated the practical deployment of ML-driven intrusion detection within a privacy-aware edge–cloud architecture for IoT/IIoT remote monitoring and control systems. Building on the offline models developed in Chapter 4, the work validated real-time detection at resource-constrained edge nodes while leveraging the cloud for aggregation, analytics, and incident management.

The proposed architecture enables low-latency detection, data minimization, and reduced bandwidth usage, while maintaining integration with monitoring and response workflows. The introduction of a runtime model-switching mechanism further enhances adaptability, allowing dynamic updates of detection models without interrupting system operation.

Beyond performance, the chapter highlights the importance of architecture-driven security design, showing how embedding IDS within system infrastructure improves manageability and reduces long-term cloud costs. However, limitations related to offline-trained models, edge resource constraints, and limited validation scope indicate the need for improved adaptability, scalability, and real-world evaluation.

Thus, this chapter further advances objective O3 by bridging the gap between offline ML analysis and operational deployment, laying the foundation for the next chapter, which extends the framework toward hybrid, resilience-oriented architectures.

Chapter 6. Resilient Hybrid Edge–Fog Architectures for ML-Driven Intrusion Detection in IoT/IIoT Remote Monitoring and Control Systems

This chapter extends the previous edge-based framework toward a resilience-oriented hybrid Edge–Fog architecture. The study aims to (i) quantify the impact of network-layer attacks (MQTT flooding and malformed packet injection) on system latency and reliability, and (ii) evaluate whether distributing IDS inference across edge and fog layers improves resilience. By deploying identical ML models across both layers, the architecture leverages low-latency edge processing and higher computational capacity at the fog layer, enabling redundant detection and cooperative alerting while maintaining real-time performance and enhancing system resilience under adversarial conditions [120].

6.1. System Architecture and Experimental Testbed

This section presents the extended IoT/IIoT RMC testbed used to evaluate the proposed hybrid Edge–Fog IDS architecture. Building on the infrastructure from Chapter 5, the system is enhanced to support distributed inference and resilience evaluation under cyberattack conditions.

As shown in Figure 6.1, the architecture introduces a fog-based IDS component operating alongside the edge node, forming a hybrid deployment. The fog node runs an identical ML inference module, providing redundancy and additional computational capacity during high workload or disruption. Both edge and fog instances share the same feature extraction pipeline, preprocessing steps, and trained models to ensure consistency across experiments.

To ensure portability and reproducibility, the fog-based inference engine is implemented using containerization (Docker). The deployment includes the inference script, pre-trained models, and configuration files, with Dockerfile and Docker Compose managing dependencies, execution, networking, and integration with the MQTT broker and logging systems.

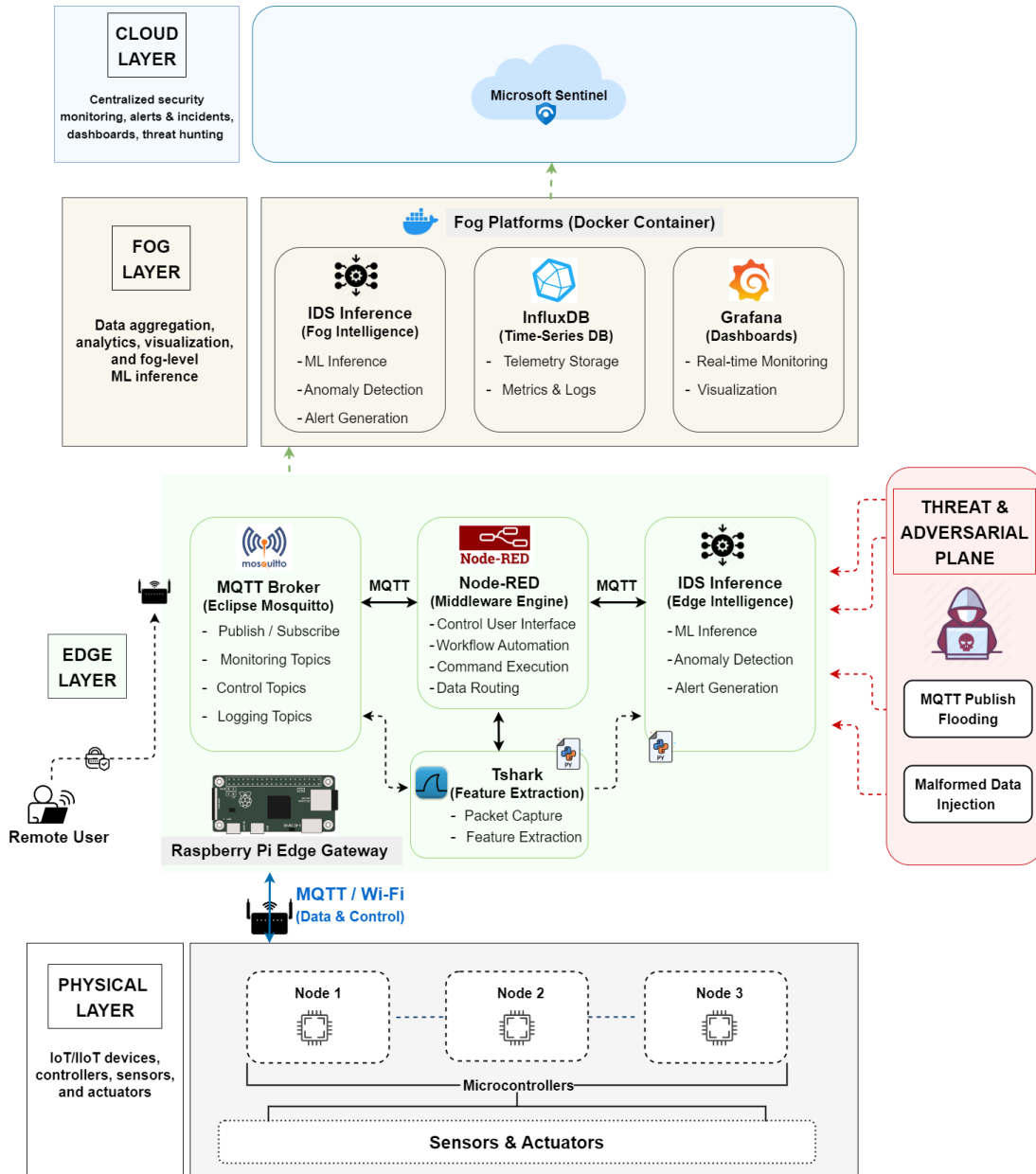


Figure 6.1. Hybrid Edge-Fog ML-driven IDS architecture for IoT/IIoT Remote Monitoring and Control systems.

6.2. Experimental Scenarios and Evaluation Methodology

The resilience of the hybrid Edge-Fog IDS architecture was evaluated through controlled experiments under two cyberattack scenarios: MQTT flooding and malformed-packet injection. Each scenario was tested across three deployment configurations, edge-based, fog-based, and hybrid IDS, under a three-phase workflow: baseline (normal operation), attack phase, and post-attack recovery. Approximately 1,000 automated control commands were generated per phase to ensure consistent workload conditions.

The actuation layer emulated representative cyber-physical processes using a relay-based door lock (discrete control) and a DC motor fan (continuous control), enabling evaluation across different control dynamics typical of IoT/IIoT RMC systems.

Performance was assessed using key metrics, including median and 95th percentile control latency, command success rate, and CPU/memory utilization, providing insight into responsiveness, reliability, and resource efficiency [121]. Additional metrics—such as inference latency, end-to-end latency, and IDS throughput (messages per second)—were used to evaluate processing efficiency across the detection

pipeline. All data were collected and analyzed using Python-based tools within the Anaconda environment.

6.3. Results and Discussions

6.3.1 Control Latency and Reliability Performance

As illustrated in Figure 6.2, under normal operating conditions (baseline), the system maintained median control latencies ranging between 80 ms and 190 ms across all deployment modes, confirming its suitability for real-time operation. When subjected to MQTT flooding and malformed-packet injection attacks, latency increased moderately, with the degree of degradation varying by IDS deployment layer. In this context, operational reliability is reflected in the command success rate, which measures the proportion of control commands successfully delivered and executed by the actuators without loss or excessive delay. High success rates, therefore, indicate stable and dependable system behavior under both normal and adversarial conditions.

During the MQTT flooding phase, the door-lock actuator recorded a median latency of 194 ms (p95 = 253 ms) under the edge-based IDS and 173 ms (p95 = 222 ms) under the fog-based configuration. In contrast, the hybrid Edge-Fog IDS achieved the best balance, maintaining a median latency of 172.5 ms (p95 = 233 ms) and a command success rate of 98.6%, nearly identical to baseline performance. Similar behavior was observed for the cooling-fan actuator, where the hybrid deployment again achieved the lowest latency (84 ms) and the highest reliability (99.2%). Under malformed-packet injection attacks, the hybrid IDS maintained near-baseline performance, with median latencies of 166.5 ms for the door actuator and 77 ms for the fan, and success rates exceeding 99.6%.

Although the observed control latency deviations remain moderate, the evaluated attacks have a more pronounced impact on intrusion-detection inference latency and throughput stability than on end-to-end control latency, as demonstrated in the subsequent section.

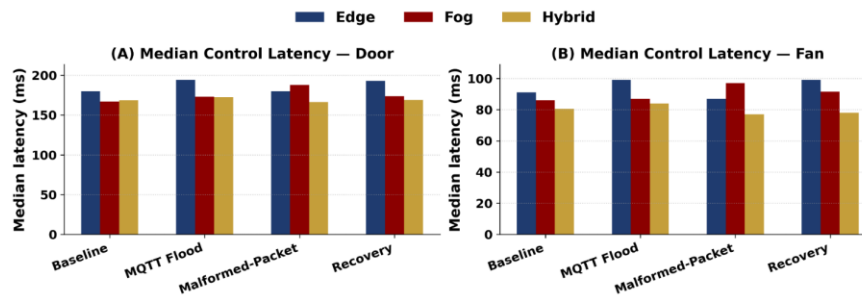


Figure 6.2. Median end-to-end control latency across baseline, attack, and recovery phases for (A) Door Lock Actuator and (B) Cooling Fan Actuator.

6.3.2 Resource Utilization Analysis

Resource utilization statistics indicate that computational overhead remained modest across all configurations. For the edge-based IDS, mean CPU utilization increased slightly from 5.4% to 6.2% during MQTT flooding, with peak values approaching 51%. The fog-based deployment exhibited the lowest resource consumption, averaging 2% CPU usage and 37% memory utilization. The hybrid configuration recorded average CPU and memory utilization of 6.4% and 50%, respectively, remaining well within acceptable limits for embedded and industrial systems. Memory utilization, which peaks at below 65%, further indicates that the distributed architecture remains scalable and resource efficient. Reduced computational overhead also improves energy efficiency, which is particularly advantageous for power-constrained IoT deployments.

6.3.3 Inference Latency and Throughput Performance

The fog-based deployment demonstrated superior inference performance, achieving lower average inference latency (5.06 ms vs. 9.32 ms at the edge) and end-to-end latency (12.96 ms vs. 22.63 ms), with reduced maximum delays (see Figure 6.3). These results reflect the fog node's higher computational capacity, while the resource-constrained edge exhibited greater variability under heavy traffic.

Throughput analysis (see Figure 6.4) showed similar average rates (~ 4.95 messages/s) for both deployments; however, the fog provided more stable performance, whereas the edge experienced fluctuations and drops under resource contention.

Despite these improvements, the evaluation was limited to two attack scenarios and a single-node setup. Future work will extend to additional attack types, multi-node scalability, and integration with zero-trust and adaptive security mechanisms.

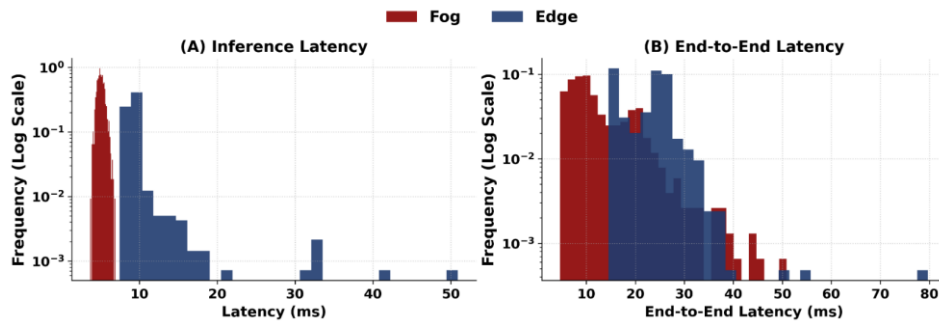


Figure 6.3. Distribution of latency (A) and end-to-end latency (B) for the deployed inference system.

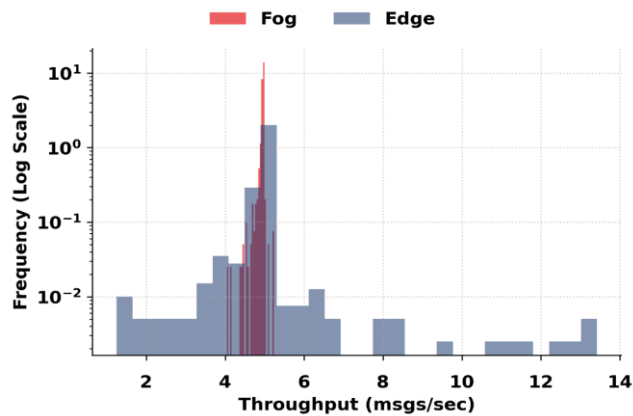


Figure 6.4. Distribution of Throughput for the deployed inference system.

6.4. Chapter Conclusion

This chapter evaluated the resilience of IoT/IIoT-based remote monitoring and control (RMC) systems under cyberattacks, focusing on the influence of ML-driven IDS inference placement. Using the testbed introduced in Chapter 5, the study assessed the impact of MQTT Flood and Malformed-Packet attacks and investigated the effectiveness of a hybrid Edge–Fog deployment. It therefore advances and completes objective O3.

Experimental results showed that both attacks produced only moderate increases in control latency (4–20 ms) without disrupting system operation. While end-to-end control performance remained largely stable, the IDS layer experienced more noticeable effects, particularly in inference latency and throughput stability. The hybrid Edge–Fog architecture mitigated these impacts by distributing inference across edge and fog nodes, leveraging redundancy and the superior processing stability of the fog layer. Future work will address scalability, adaptive learning, and Zero Trust integration.

Chapter 7. Final Conclusions, Original Contributions, Dissemination of Results, and Further Research

7.1. Conclusions

This doctoral thesis addresses the cybersecurity challenges of IoT/IIoT-based remote engineering systems, with particular emphasis on secure access control, resource-efficient intrusion detection, and resilience-oriented deployment architectures. The research was motivated by the increasing convergence of distributed connectivity, heterogeneous system components, and real-time operational requirements, which collectively expand the attack surface while limiting the applicability of conventional IT-centric security approaches.

The thesis objectives (O1–O3), as defined in Chapter 1, are addressed through a progressive investigation spanning system characterization, identity-centric access control, machine learning–driven intrusion detection, and deployment-aware IDS architectures across Chapters 2–6. The initial cross-domain analysis established that remote engineering systems exhibit structural convergence between educational and industrial environments, driven by IoT/IIoT integration and cloud connectivity, while simultaneously inheriting vulnerabilities from both domains.

Building on this foundation, the thesis demonstrated that identity-centric access control, implemented through Zero Trust principles, effectively mitigates risks associated with implicit trust and exposure to excessive privilege. However, the findings also confirmed that access control alone is insufficient to detect behavioral or post-compromise threats, thereby necessitating complementary detection mechanisms.

From a detection perspective, the thesis introduced a feature-selection–driven machine-learning methodology for classifying MQTT-based traffic. The results showed that intrusion detection can be adapted to resource-constrained IoT/IIoT environments without prohibitive computational overhead, enabling practical deployment at the edge and embedded platforms.

At the system level, this thesis investigates deployment strategies for intrusion detection, addressing trade-offs between detection effectiveness, latency, resource utilization, and data privacy. A modular, privacy-aware IDS architecture is designed and experimentally validated, combining resource-efficient edge-level inference with centralized cloud-based monitoring. Furthermore, the experimental results indicate that distributing inference across the edge–fog–cloud continuum improves resilience to cyberattacks. In particular, the hybrid Edge–Fog deployment exhibited reduced latency degradation, consistently high actuation reliability, and near-baseline system performance, while maintaining minimal resource overhead.

The thesis adopts a layered, system-level perspective that aligns security design with the operational realities of distributed, real-time IoT/IIoT environments. The findings show that improving the security of remote engineering systems requires complementary, coordinated mechanisms that combine identity-centric access control, resource-efficient intrusion detection, and deployment-aware strategies across edge, fog, and cloud layers.

7.2. Original Contributions of Research

This thesis presents a set of original and complementary contributions to the cybersecurity of IoT/IIoT-based remote engineering systems. These contributions advance a system-level perspective spanning access control, intrusion detection, and deployment-aware security design.

The principal contributions are summarized as follows.

1. Conceptual and architectural contributions

This thesis provides a structured, security-oriented characterization of remote engineering systems across educational and industrial domains, identifying common vulnerabilities arising from IT/OT convergence, distributed connectivity, and legacy integration. It further operationalizes Zero Trust principles by designing and validating an identity-centric, resource-level access-control architecture, demonstrating its effectiveness at mitigating lateral movement and excessive privilege exposure in remote environments.

2. Methodological contributions in machine learning–driven intrusion detection

A feature-selection–driven methodology for MQTT traffic classification is proposed and validated, enabling resource-efficient intrusion detection with reduced feature dimensionality. The approach maintains high detection performance while supporting deployment on constrained edge devices, improving the practicality of ML-based IDS in IoT/IIoT systems.

3. System-level contributions to deployable IDS Architectures

The thesis designs and validates a modular, privacy-aware Edge–Cloud IDS architecture that integrates edge-level inference with selective cloud-based SIEM. Building on this, a hybrid Edge–Fog deployment is introduced, demonstrating improved resilience under cyberattacks by reducing latency degradation, maintaining high reliability, and preserving system performance with minimal resource overhead.

7.3. Dissemination of Research Results

The scientific contributions presented in this thesis have been disseminated through peer-reviewed journals, international conference proceedings, and an edited book chapter. The publications listed below include both works that directly support the research contributions presented in the thesis and complementary studies published during the doctoral period.

A. Journal Publications (ISI-Rated)

1. **Tuyishime, E., Balan, T. C., Cotfas, P. A., Cotfas, D. T., & Rekeraho, A. (2023). Enhancing cloud security—proactive threat monitoring and detection using a SIEM-based approach.** *Applied Sciences*, 13(22), 12359. <https://doi.org/10.3390/app132212359>, (IF: 2.9, Q2, JCR 2025).
2. Rekeraho, A., Cotfas, D. T., Cotfas, P. A., Bălan, T. C., **Tuyishime, E., & Acheampong, R. (2024). Cybersecurity challenges in IoT-based smart renewable energy.** *International Journal of Information Security*, 23(1), 101-117. <https://doi.org/10.1007/s10207-023-00732-9>, (IF: 5, Q1, JCR 2025).
3. Rekeraho, A., Cotfas, D. T., Cotfas, P. A., **Tuyishime, E., Balan, T. C., & Acheampong, R. (2024). Enhancing security for IoT-based smart renewable energy remote monitoring systems.** *Electronics*, 13(4), 756. <https://doi.org/10.3390/electronics13040756>, (IF: 2.9, Q2, JCR 2025).
4. **Tuyishime, E., Martalò, M., Cotfas, P. A., Popescu, V., Cotfas, D. T., & Rekeraho, A. (2025). Resource-Efficient Traffic Classification Using Feature Selection for Message Queuing Telemetry Transport-Internet of Things Network-Based Security Attacks.** *Applied Sciences*, 15(8), 4252. <https://doi.org/10.3390/app15084252>, (IF: 2.9, Q2, JCR 2025).
5. Rekeraho, A., Cotfas, D. T., Balan, T. C., Cotfas, P. A., Acheampong, R., & **Tuyishime, E. (2025). Cybersecurity Threat Modeling for IoT-Integrated Smart Solar Energy Systems: Strengthening Resilience for Global Energy Sustainability.** *Sustainability*, 17(6), 2386. <https://doi.org/10.3390/su17062386>, (IF: 4.1, Q2, JCR 2025).
6. Acheampong, R., Balan, T. C., Popovici, D. M., **Tuyishime, E., Rekeraho, A., & Voinea, G. D. (2025). Balancing usability, user experience, security and privacy in XR systems: A multidimensional approach.** *International Journal of Information Security*, 24(3), 1-18. <https://doi.org/10.1007/s10207-025-01025-z>, (IF: 5, Q1, JCR 2025).

B. International Conference Proceedings (ISI and BDI-Rated)

7. Manzi, F., **Tuyishime, E., Hitayezu, A., Muhawenayo, G., Nsengiyumva, P., and Jayavel, K. (2023, November). Leveraging Machine Learning and Raspberry Pi for Enhanced Wildlife Remote Monitoring and Localization.** In 2023 International Conference on Modeling & E-Information Research, Artificial Learning and Digital Applications (ICMERALDA) (pp. 301-306). IEEE. <https://doi.org/10.1109/ICMERALDA60125.2023.10458213>

8. Hitayezu, A., Umutoni, R. M., **Tuyishime, E.**, Jayavel, K., & Kabiri, C. (2023, December). **Enhancing Road Safety with Cloud-Integrated IoT Road Signal Detection System**. In 2023 IEEE 16th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc) (pp. 117-122). IEEE. <https://doi.org/10.1109/MCSoc60832.2023.00025>, WOS:001190988000017
9. **Tuyishime, E.**, Radu, F., Cotfas, P., Cotfas, D., Balan, T., & Rekeraho, A. (2024, June). **Online laboratory access control with zero trust approach: Twingate use case**. In 2024, the 16th International Conference on Electronics, Computers and Artificial Intelligence (ECAI) (pp. 1-7). IEEE. <https://doi.org/10.1109/ECAI61503.2024.10607562>
10. Radu, F., Cotfas, P. A., **Tuyishime, E.**, Cotfas, D. T., Constantin, A., Alexandru, M., & Popescu, V. (2025, October). **Secure Testing and Simulation System Based on Raspberry Pi**. In 2025, IEEE 31st International Symposium for Design and Technology in Electronic Packaging (SIITME) (pp. 98-102). IEEE. <https://doi.org/10.1109/SIITME67657.2025.11293695>
11. **Tuyishime, E.**, Cotfas, P. A., Balan, T. C., Rekeraho, A., Cotfas, D. T., & Popescu, V. (2026). **ML-Based Intrusion Detection in IoT/IIoT with Privacy-Aware Offloading to Cloud SIEM**. In *Proceedings of the 12th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP*; ISBN 978-989-758-800-6; ISSN 2184-4356, SciTePress, pages 489-496.
<https://doi.org/10.5220/0014213600004061>
12. **Tuyishime, E.**, Cotfas, P. A., Cotfas, D. T., Balan, T. C., & Popescu, V. (2026). **Hybrid Edge-Fog ML-Driven IDS for Resilient IoT/IIoT Remote Monitoring and Control**. In: *Proceedings of the 23rd International Conference on Smart Technologies & Education (STE2026)*. [Accepted for publication].
- C. Book Chapter (LNNS, Springer)**
13. Radu, F., **Tuyishime, E.**, Cotfas, P., Cotfas, D., Balan, T., Rekeraho, A. (2024). **Online Serial Laboratories**. In: May, D., Auer, M.E., Kist, A. (eds) *Online Laboratories in Engineering and Technology Education*. Lecture Notes in Networks and Systems, vol 1135. Springer, Cham.
https://doi.org/10.1007/978-3-031-70771-1_19

7.4. Directions for Further Research

While this thesis has made several contributions to addressing cybersecurity challenges in remote engineering systems, several limitations and open research directions remain to be addressed in future work, including:

1. Adaptive learning mechanisms for ML-driven IDS: The current system relies on offline-trained models. Extending research toward adaptive and federated learning approaches would allow IDS models to dynamically adapt to data drift and evolving attack patterns in IoT/IIoT environments.
2. Scalability analysis in multi-node edge-fog deployments: Large-scale industrial environments typically involve multiple distributed clusters and gateways. Future work should evaluate performance, detection accuracy, and resource consumption under multi-cluster and high-concurrency scenarios.
3. Self-healing cyber-physical security architecture: Beyond detection, future systems should integrate automated response, containment, and recovery mechanisms, enabling self-adaptive IoT/IIoT-based RMC infrastructures.
4. Extension of WebSocket over QUIC [122] and MQTT (WSQM) to remote engineering systems: Further investigation may explore the extension of the WSQM approach [123] to remote engineering environments, with particular emphasis on low-latency encrypted transport, session persistence, and integration with policy-driven security controls.

References

- [1] R. Heradio, L. de la Torre, and S. Dormido, "Virtual and remote labs in control education: A survey," *Annu. Rev. Control*, vol. 42, pp. 1–10, Jan. 2016.
- [2] J. Miranda, C. Navarrete, J. Noguez, J.-M. Molina-Espinosa, M.-S. Ramírez-Montoya, S. A. Navarro-Tuch, M.-R. Bustamante-Bello, J.-B. Rosas-Fernández, and A. Molina, "The core components of education 4.0 in higher education: Three case studies in engineering education," *Comput. Electr. Eng.*, vol. 93, p. 107278, July 2021.
- [3] J.-Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, "Industrial Internet: A Survey on the Enabling Technologies, Applications, and Challenges," vol. 19, no. 3, pp. 1504–1526, 2017.
- [4] P. Helo, Y. Hao, R. Toshev, and V. Boldosova, "Cloud manufacturing ecosystem analysis and design," *Robot. Comput.-Integr. Manuf.*, vol. 67, p. 102050, Feb. 2021.
- [5] F. J. Folgado, D. Calderón, I. González, and A. J. Calderón, "Review of Industry 4.0 from the Perspective of Automation and Supervision Systems: Definitions, Architectures and Recent Trends," *Electronics*, vol. 13, no. 4, p. 782, Jan. 2024.
- [6] D. Miller, "Remote Access Grows Even as Cybersecurity Fears Surge," *Automation World*, 16-Sept-2021. [Online]. Available: <https://www.automationworld.com/cybersecurity/article/21722664/remote-access-grows-even-as-cybersecurity-fears-surge>. [Accessed: 14-Feb-2026].
- [7] A. V. den Beemt, S. Groothuijsen, L. Ozkan, and W. Hendrix, "Remote labs in higher engineering education: engaging students with active learning pedagogy," vol. 35, no. 2, pp. 320–340, Aug. 2023.
- [8] T. Chamunorwa, H. A. Modran, D. Ursuțiu, C. Samoilă, and H. Hedeșiu, "Cloud-Based, Expandable—Reconfigurable Remote Laboratory for Electronic Engineering Experiments," *Electronics*, vol. 11, no. 20, p. 3292, Jan. 2022.
- [9] J. Amankwah-Amoah, Z. Khan, G. Wood, and G. Knight, "COVID-19 and digitalization: The great acceleration," *J. Bus. Res.*, vol. 136, pp. 602–611, Nov. 2021.
- [10] B. Galloway and G. P. Hancke, "Introduction to Industrial Control Networks," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 2, pp. 860–880, 2013.
- [11] J. Cecílio and P. Furtado, "Industrial Control Systems: Concepts, Components, and Architectures," in *Wireless Sensors in Industrial Time-Critical Environments*, J. Cecílio and P. Furtado, Eds. Cham: Springer International Publishing, 2014, pp. 7–18.
- [12] S. Vitla, "Unsecured Remote Desktop Protocol (RDP) Access: A Gateway for Ransomware Attacks and Corporate Extortion," *J. Comput. Sci. Technol. Stud.*, vol. 6, no. 2, pp. 150–165, May 2024.
- [13] E. Byres, D. Leversage, and N. Kube, "Security incidents and trends in SCADA and process industries," *Ind. Ethernet Book*, vol. 39, no. 2, pp. 12–20, 2007.
- [14] Y. Chen, H. Hu, and G. Cheng, "Design and implementation of a novel enterprise network defense system by maneuvering multi-dimensional network properties," *Front. Inf. Technol. Electron. Eng.*, vol. 20, no. 2, pp. 238–252, Feb. 2019.
- [15] J. Kindervag, B. Stephanie, M. Kelley, and B. Josh, "No More Chewy Centers: The Zero Trust Model Of Information Security," Forrester Research, Inc., Cambridge, MA 02139 USA, Mar. 2016.
- [16] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A Survey on Security and Privacy Issues in Edge-Computing-Assisted Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4004–4022, Mar. 2021.
- [17] J. Arshad, M. A. Azad, R. Amad, K. Salah, M. Alazab, and R. Iqbal, "A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT," *Electronics*, vol. 9, no. 4, p. 629, Apr. 2020.
- [18] L. Gomes and S. Bogosyan, "Current Trends in Remote Laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4744–4756, Dec. 2009.
- [19] B. Aktan, C. A. Bohus, L. A. Crawl, and M. H. Shor, "Distance learning applied to control engineering laboratories," *IEEE Trans. Educ.*, vol. 39, no. 3, pp. 320–326, Aug. 1996.
- [20] S. Frerich, D. Kruse, M. Petermann, and A. Kilzer, "Virtual Labs and Remote Labs: Practical experience for everyone," in *2014 IEEE Global Engineering Education Conference (EDUCON)*, 2014, pp. 312–314.

- [21] H. Tjahyadi, K. Prasetya, and I. M. Murwantara, "Digital Twin Based Laboratory for Control Engineering Education," *Int. J. Inf. Educ. Technol.*, vol. 13, no. 4, pp. 704–711, 2023.
- [22] J. Ma and J. V. Nickerson, "Hands-on, simulated, and remote laboratories: A comparative literature review," *ACM Comput. Surv.*, vol. 38, no. 3, pp. 7-es, Sept. 2006.
- [23] J. Grodotzki, T. R. Ortelt, and A. E. Tekkaya, "Remote and Virtual Labs for Engineering Education 4.0: Achievements of the ELLI project at the TU Dortmund University," *Procedia Manuf.*, vol. 26, pp. 1349–1360, Jan. 2018.
- [24] A. Van den Beemt, S. Groothuijsen, L. Ozkan, and W. Hendrix, "Remote labs in higher engineering education: engaging students with active learning pedagogy," *J. Comput. High. Educ.*, vol. 35, no. 2, pp. 320–340, Aug. 2023.
- [25] T. Chamunorwa, H. A. Modran, D. Ursuțiu, C. Samoilă, and H. Hedeșiu, "Cloud-Based, Expandable—Reconfigurable Remote Laboratory for Electronic Engineering Experiments," *Electronics*, vol. 11, no. 20, p. 3292, Jan. 2022.
- [26] A. E. Pena-Molina and M. M. Larrondo-Petrie, "Safety and Security Considerations for Online Laboratory Management Systems," *J. Cybersecurity Priv.*, vol. 5, no. 2, p. 24, June 2025.
- [27] J. Machotka, Z. Nedic, and Ö. Göl, "Collaborative Learning in the Remote Laboratory NetLab," *J. Syst. Cybern. Inform.*, vol. 6, June 2008.
- [28] C. Lazar and S. Carari, "A remote-control engineering laboratory," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2368–2375, 2008.
- [29] P. A. Cotfas, D. T. Cotfas, D. Ursutiu, and C. Samoila, "Remote Laboratory in Photovoltaics," *Int. J. Online Eng.*, vol. 5, no. 3, 2009.
- [30] D. Pop, D. G. Zutin, M. E. Auer, K. Henke, and H. D. Wuttke, "An online lab to support a master program in remote engineering," in *2011 Frontiers in Education Conference (FIE)*, 2011, pp. 2–1.
- [31] C. A. Jara, F. A. Candelas, S. T. Puente, and F. Torres, "Hands-on experiences of undergraduate students in Automatics and Robotics using a virtual and remote laboratory," *Comput. Educ.*, vol. 57, pp. 2451–2461, 2011.
- [32] A. Rojko, D. Hercog, and K. Jezernik, "Power Engineering and Motion Control Web Laboratory: Design, Implementation, and Evaluation of Mechatronics Course," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3343–3354, Oct. 2010.
- [33] M. A. Vivar and A. R. Magna, "Design, implementation and use of a remote network lab as an aid to support teaching computer network," in *2008 Third International Conference on Digital Information Management*, 2008, pp. 905–909.
- [34] M. Tawfik, D. Lowe, S. Murray, M. De La Villefromoy, M. Diponio, E. Sancristobal, M. J. Albert, G. Diaz, and M. Castro, "Grid Remote Laboratory Management System," in *2013 10th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2013, pp. 1–9.
- [35] M. Kalúz, P. Orduña, J. García-Zubia, M. Fikar, and L. Ćirka, "Sharing Control Laboratories by Remote Laboratory Management System WebLab-Deusto," *IFAC Proc. Vol.*, vol. 46, no. 17, pp. 345–350, Jan. 2013.
- [36] E. Werner, J. C. Matias, M. D. Berejuck, and H. Saliah-Hassane, "Main Attacks and Ways to Keep Security Guidelines Updated in Remote Laboratories," in *Online Engineering and Society 4.0*, vol. 298, M. E. Auer, K. R. Bhimavaram, and X.-G. Yue, Eds. Cham: Springer International Publishing, 2022, pp. 115–121.
- [37] Y. Larbaoui, A. Naddami, and A. Fahli, "Security, Control and Management of Smart Remote Laboratory for Remote Experiments in Electricity and Electronics," *Int. J.*, vol. 9, no. 3, 2020.
- [38] E. Lisova, I. Šljivo, and A. Čaušević, "Safety and security co-analyses: a systematic literature review," *IEEE Syst J*, vol. 13, no. 3, pp. 2189–2200, 2018.
- [39] Joint Task Force Transformation Initiative, "Guide for conducting risk assessments," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-30r1, 2012.
- [40] R. K. Alqurashi, M. A. AlZain, B. Soh, M. Masud, and J. Al-Amri, "Cyber Attacks and Impacts: A Case Study in Saudi Arabia," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1, pp. 217–224, 2020.
- [41] S. Christey and R. A. Martin, "CWE - Vulnerability Type Distributions in CVE," in *Cwe.mitre.org*, p. .
- [42] M. Bach-Nutman, "Understanding The Top 10 OWASP Vulnerabilities." arXiv, 17-Dec-2020.

- [43] M. A. Al-Shareeda, S. Manickam, and S. A. Sari, "A Survey of SQL Injection Attacks, Their Methods, and Prevention Techniques," in *2022 International Conference on Data Science and Intelligent Computing (ICDSIC)*, 2022, pp. 31–35.
- [44] "Top 6 Industries Targeted By Threat Actors In 2024 | Cyble," 03-Dec-2024. .
- [45] D. Uckelmann, D. Mezzogori, G. Esposito, M. Neroni, D. Reverberi, and M. Ustenko, "Safety and Security in Federated Remote Labs—A Requirement Analysis," in *Cross Reality and Data Science in Engineering: Proceedings of the 17th International Conference on Remote Engineering and Virtual Instrumentation 17*, 2021, pp. 21–36.
- [46] E. Werner, J. C. Matias, M. D. Berejuck, and H. Saliah-Hassane, "Evaluation of blockchain techniques to ensure secure access on remote FPGA laboratories," in *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*, 2021, pp. 1–6.
- [47] M. Krbeček, F. Schauer, and R. Jasek, "Security aspects of remote e-laboratories," *Int J Online Eng*, vol. 9, no. 3, pp. 34–39, 2013.
- [48] Th. Richter, R. Watson, S. Kassavetis, M. Kraft, P. Grube, D. Boehringer, P. de Vries, E. Hatzikraniotis, and S. Logothetidis, "The WebLabs of the University of Cambridge: A study of securing remote instrumentation," in *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2012, pp. 1–5.
- [49] J. A. Sanchez-Viloria, L. F. Zapata-Rivera, C. Aranzazu-Suescun, A. E. Molina-Pena, and M. M. Larrondo-Petrie, "Online Laboratory Communication Using MQTT IoT Standard," in *2021 World Engineering Education Forum/Global Engineering Deans Council (WEEF/GEDC)*, 2021, pp. 550–555.
- [50] D. Dumo, "Cyber security concerns of Supervisory Control and Data Acquisition (SCADA) systems," in *2010 IEEE International Conference on Technologies for Homeland Security (HST)*, 2010, pp. 473–475.
- [51] A. Aijaz and M. Sooriyabandara, "The Tactile Internet for Industries: A Review," *Proc. IEEE*, vol. 107, no. 2, pp. 414–435, Feb. 2019.
- [52] K. J. Åström and P. R. Kumar, "Control: A perspective," *Automatica*, vol. 50, no. 1, pp. 3–43, Jan. 2014.
- [53] U. P. D. Ani, H. (Mary) He, and A. Tiwari, "Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective," *J. Cyber Secur. Technol.*, vol. 1, no. 1, pp. 32–74, Jan. 2017.
- [54] "Critical Infrastructure Sectors | CISA." [Online]. Available: <https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors>. [Accessed: 29-Dec-2025].
- [55] "The Ultimate Guide to Physical Security." [Online]. Available: <https://buildings.honeywell.com/us/en/brands/our-brands/lenels2/news/insights/the-ultimate-guide-to-physical-security>. [Accessed: 29-Dec-2025].
- [56] M. P. Coole and D. J. Brooks, "Physical Security: Best Practices," in *Encyclopedia of Security and Emergency Management*, L. R. Shapiro and M.-H. Maras, Eds. Cham: Springer International Publishing, 2021, pp. 730–737.
- [57] M. Rapaccini, N. Saccani, C. Kowalkowski, M. Paiola, and F. Adrodegari, "Navigating disruptive crises through service-led growth: The impact of COVID-19 on Italian manufacturing firms," *Ind. Mark. Manag.*, vol. 88, pp. 225–237, July 2020.
- [58] A. Margherita and M. Heikkilä, "Business continuity in the COVID-19 emergency: A framework of actions undertaken by world-leading companies," *Bus. Horiz.*, vol. 64, no. 5, pp. 683–695, Sept. 2021.
- [59] "REMOTE CONTROL, the ultimate convenience technology to address COVID-19 pandemic concerns and demographic workforce pressures," *Condition Monitoring*. July-2020.
- [60] T. Johan, D. Kristofer, M. György, S. Björn, M. Gunnar, M. Tomas, S. John, N. Jan, N. Jonas, M. Carlos, Z. Biao, and W. Jianjun, "Industrial remote operation: 5G rises to the challenge - Ericsson Technology Review," Nov. 2015.
- [61] G. M. Achilli, F. Chinello, C. Fang, P. G. Hernandez, S. Logozzo, and M. C. Valigi, "Preliminary Study on a Handle with Haptic Devices for Collaborative Robotics in a Remote Maintenance Environment," in *Proceedings of I4SDG Workshop 2023: IFToMM for Sustainable Development Goals*, 2023, pp. 484–491.

- [62] L. Seiffert, J. Sczodrok, J. Ghofrani, and K. Wieczorek, *Remote Technol. Common Pract. Ind. Maint. What Experts Say Telecom*, vol. 3, no. 4, pp. 548–563, Sept. 2022.
- [63] D. Mourtzis, V. Siatras, and J. Angelopoulos, “Real-time remote maintenance support based on augmented reality (AR,” *Appl. Sci.*, vol. 10, no. 5, 2020.
- [64] “Configuring and Managing Remote Access for Industrial Control Systems,” Centre for the Protection of National Infrastructure, 2010.
- [65] M. R. Asghar, Q. Hu, and S. Zeadally, “Cybersecurity in industrial control systems: Issues, technologies, and challenges,” *Comput. Netw.*, vol. 165, p. 106946, Dec. 2019.
- [66] C. Alcaraz, G. Fernandez, and F. Carvajal, “Security Aspects of SCADA and DCS Environments,” in *Critical Infrastructure Protection: Information Infrastructure Models, Analysis, and Defense*, J. Lopez, R. Setola, and S. D. Wolthusen, Eds. Berlin, Heidelberg: Springer, 2012, pp. 120–149.
- [67] T. Menze, “The state of industrial cybersecurity in the era of digitalization,” *Online Httpsics Kaspersky Commed. ARC ICS-2020-Trend-Rep. Pdf*, 2020.
- [68] M. Marali, S. D. Sudarsan, and A. Gogioneni, “Cyber security threats in industrial control systems and protection,” in *2019 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2019, pp. 1–7.
- [69] M. Conti, D. Donadel, and F. Turrin, “A Survey on Industrial Control System Testbeds and Datasets for Security Research,” *IEEE Commun. Surv. Tutor.*, vol. 23, no. 4, pp. 2248–2294, 2021.
- [70] J. P. Farwell and R. Rohozinski, “Stuxnet and the Future of Cyber War,” *Survival*, vol. 53, no. 1, pp. 23–40, Feb. 2011.
- [71] T. Miller, A. Staves, S. Maesschalck, M. Sturdee, and B. Green, “Looking back to look forward: Lessons learnt from cyber-attacks on Industrial Control Systems,” *Int. J. Crit. Infrastruct. Prot.*, vol. 35, p. 100464, Dec. 2021.
- [72] T. Alladi, V. Chamola, and S. Zeadally, “Industrial Control Systems: Cyberattack trends and countermeasures,” *Comput. Commun.*, vol. 155, pp. 1–8, Apr. 2020.
- [73] K. Pawel and K. Nikita, “The Russia-Ukraine Cyber War Part 3: Attacks on Telecom and Critical Infrastructure,” *LevelBlue*, 05-Mar-2025. .
- [74] S. Gatlan, “Ransomware attack forces 100 Romanian hospitals to go offline,” *BleepingComputer*, 12-Feb-2024. [Online]. Available: <https://www.bleepingcomputer.com/news/security/ransomware-attack-forces-100-romanian-hospitals-to-go-offline/>. [Accessed: 15-Mar-2026].
- [75] R. Sangkhro and A. K. Agrawal, “Cybersecurity in Industrial Control Systems: A Review of the Current Trends and Challenges,” in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2023, pp. 355–359.
- [76] D. Wu, A. Ren, W. Zhang, F. Fan, P. Liu, X. Fu, and J. Terpenney, “Cybersecurity for digital manufacturing,” *J. Manuf. Syst.*, vol. 48, pp. 3–12, July 2018.
- [77] A. Shahzad, M. Lee, H. D. Kim, S. Woo, and N. Xiong, “New Security Development and Trends to Secure the SCADA Sensors Automated Transmission during Critical Sessions,” *Symmetry*, vol. 7, no. 4, pp. 1945–1980, Dec. 2015.
- [78] S. Ndichu, S. McOyowo, H. Okoyo, and C. Wekesa, “A Remote Access Security Model based on Vulnerability Management,” *Int. J. Inf. Technol. Comput. Sci.*, vol. 12, no. 5, pp. 38–51, Oct. 2020.
- [79] C. Zanas, S. Russo, and M. Colajanni, “Flexible zero trust architecture for the cybersecurity of industrial IoT infrastructures,” *Ad Hoc Netw.*, vol. 156, p. 103414, Apr. 2024.
- [80] “2024 VPN Risk Report [HPE],” *Cybersecurity Insiders*. .
- [81] S. W. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero Trust Architecture,” *NIST*, Aug. 2020.
- [82] H. Holm, M. Karresand, A. Vidström, and E. Westring, “A Survey of Industrial Control System Testbeds,” in *Secure IT Systems*, 2015, pp. 11–26.
- [83] C. Ekisa, D. Ó. Briain, and Y. Kavanagh, “VICSORT - A Virtualised ICS Open-source Research Testbed,” in *2022 Cyber Research Conference - Ireland (Cyber-RCI)*, 2022, pp. 1–8.
- [84] F. Björck, M. Henkel, J. Stirna, and J. Zdravkovic, “Cyber Resilience – Fundamentals for a Definition,” in *New Contributions in Information Systems and Technologies*, 2015, pp. 311–316.
- [85] D. Galinec and W. Steingartner, “Combining cybersecurity and cyber defense to achieve cyber resilience,” in *2017 IEEE 14th International Scientific Conference on Informatics*, 2017, pp. 87–93.

- [86] T. N. I. Alrumaih, M. J. F. Alenazi, N. A. AlSowaygh, A. A. Humayed, and I. A. Alablani, "Cyber resilience in industrial networks: A state of the art, challenges, and future directions," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 9, p. 101781, Oct. 2023.
- [87] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, no. 8, pp. 1245–1265, June 2010.
- [88] K. Stouffer, "Guide to Operational Technology (OT) Security," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-82r3, 2023.
- [89] "ISA/IEC 62443 Series of Standards - ISA," *isa.org*. [Online]. Available: <https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards>. [Accessed: 04-Jan-2026].
- [90] "ISO - ISO/IEC 27000 family — Information security management," *ISO*. [Online]. Available: <https://www.iso.org/standard/iso-iec-27000-family>. [Accessed: 04-Jan-2026].
- [91] *VDI/VDE 2182 Blatt 1 - IT-security for industrial automation - General model*. 2020.
- [92] D. Bodeau and R. Graubart, "Cyber Resiliency Design Principles Selective Use Throughout the Lifecycle and in Conjunction with Related Disciplines," MITRE CORP, Technical Report MTR170001, Jan. 2017.
- [93] D. Liveri, A. Sarri, and E. Darra, "ENISA's Contribution to National Cyber Security Strategies," in *Cybersecurity Best Practices: Lösungen zur Erhöhung der Cyberresilienz für Unternehmen und Behörden*, M. Bartsch and S. Frey, Eds. Wiesbaden: Springer Fachmedien, 2018, pp. 43–64.
- [94] I.-A. Dumitru, "Zero Trust Security," in *Proceedings of the International Conference on Cybersecurity and Cybercrime - 2022*, vol. IX, Romania: Asociatia Romana pentru Asigurarea Securitatii Informatiei, 2022, pp. 99–104.
- [95] C. Buck, C. Olenberger, A. Schweizer, F. Völter, and T. Eymann, "Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust," *Comput. Secur.*, vol. 110, p. 102436, Nov. 2021.
- [96] J. Kindervag, B. Stephanie, and C. Lindsey, "No More Chewy Centers: Introducing The Zero Trust Model Of Information Security," Forrester Research, Inc., Cambridge, MA 02139 USA, Sept. 2010.
- [97] J. Kindervag, "Build Security Into Your Network's DNA: The Zero Trust Network Architecture," 2010.
- [98] F. Radu, E. Tuyishime, P. Cotfas, D. Cotfas, T. Balan, and A. Rekeraho, "Online Serial Laboratories," in *Online Laboratories in Engineering and Technology Education: State of the Art and Trends for the Future*, D. May, M. E. Auer, and A. Kist, Eds. Cham: Springer, Cham, 2024, pp. 375–390.
- [99] "Twingate: It's time to ditch your VPN." [Online]. Available: <https://www.twingate.com/>. [Accessed: 14-Jan-2026].
- [100] "MQTTset," *CNR IEIIT*, 2021. [Online]. Available: <https://www.kaggle.com/datasets/cnriiit/mqttset>. [Accessed: 16-Jan-2026].
- [101] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a New Dataset for Machine Learning Techniques on MQTT," *Sensors*, vol. 20, no. 22, p. 6578, Jan. 2020.
- [102] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.
- [103] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Anal.*, vol. 1, no. 1, p. 9, Nov. 2016.
- [104] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J. Big Data*, vol. 7, no. 1, p. 52, July 2020.
- [105] T. J. Ulrych, D. R. Velis, A. D. Woodbury, and M. D. Sacchi, "L-moments and C-moments," *Stoch. Environ. Res. Risk Assess.*, vol. 14, no. 1, pp. 50–68, Mar. 2000.
- [106] T. Li, "Robust estimations from distribution structures: II. Central Moments." arXiv, June-2024.
- [107] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 729–735.
- [108] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, p. 105524, Dec. 2020.
- [109] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning." arXiv, Nov-2020.

- [110] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, and L. Invernizzi, "KerasTuner," 2019. [Online]. Available: <https://github.com/keras-team/keras-tuner>. [Accessed: 21-Jan-2026].
- [111] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J Mach Learn Res*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
- [112] D. P. Kingma and J. Lei Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations (ICLR)*, 2015, vol. 5.
- [113] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 (General Data Protection Regulation)," Regulation L 119, Apr. 2016.
- [114] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, "NIST Cloud Computing Reference Architecture. Recommendations of the National Institute of Standards and Technology." Special Publication 500-292; NIST: Gaithersburg, MD, USA, 2011.
- [115] J. Bian, A. A. Arafat, H. Xiong, J. Li, L. Li, H. Chen, J. Wang, D. Dou, and Z. Guo, "Machine Learning in Real-Time Internet of Things (IoT) Systems: A Survey," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8364–8386, June 2022.
- [116] D. Panchal, P. Verma, I. Baran, D. Musgrove, and D. Lu, "Reusable MLOps: Reusable Deployment, Reusable Infrastructure and Hot-Swappable Machine Learning models and services." arXiv, 19-Feb-2024.
- [117] A. Marda, S. Kulkarni, and K. Vaidhyanathan, "SWITCH: An Exemplar for Evaluating Self-Adaptive ML-Enabled Systems," in *Proceedings of the 19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2024, pp. 143–149.
- [118] G. Andresini, F. Pendlebury, F. Pierazzi, C. Loglisci, A. Appice, and L. Cavallaro, "INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection," in *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, 2021, pp. 111–122.
- [119] "Pricing Calculator | Microsoft Azure." [Online]. Available: <https://azure.microsoft.com/en-us/pricing/calculator/>. [Accessed: 30-Jan-2026].
- [120] O. F. Olaitan, O. N. Akatakpo, C. O. Victor, C. J. Emejulu, T. M. Ayoola, D. E. Olayiwola, and A. A. Ajibola, "Secure and Resilient Industrial IoT Architectures for Smart Manufacturing: A Comprehensive Review," *J. Eng. Res. Rep.*, vol. 27, no. 6, pp. 331–344, June 2025.
- [121] "On the Latency, Rate, and Reliability Tradeoff in Wireless Networked Control Systems for IIoT | IEEE Journals & Magazine | IEEE Xplore." [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9134368>. [Accessed: 17-Mar-2026].
- [122] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," Internet Engineering Task Force, Request for Comments RFC 9000, May 2021.
- [123] G. Pettorru and M. Martalò, "A Persistent and Secure Publish-Subscriber Architecture for Low-Latency IoT Communications," *IEEE Trans. Netw. Serv. Manag.*, vol. 23, pp. 729–739, 2026.