People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mustapha Stambouli University of Mascara

Faculty of Exact Sciences

Department of Computer Science

DOCTORAL THESIS

Option: Computer Science

Improving IoT Network Security Using Deep Learning and Hybrid Feature Selection Techniques

Presented by: **Fouad Khatemi**

Defended on **14/04/2026**

In front of the Examination Committee

President	Mrs. Yahyaoui Khadidja	Professor	University of Mascara
Examiner	Mr. Hamdani Mostefa	MCA	University of Tissemsilt
Examiner	Mr. Maaskri Moustafa	MCA	University of Tiaret
Examiner	Mr. Hamri Mohamed Mehdi	MCA	University of Sidi Bel Abbes
Examiner	Mrs. Abid Sabrina	MCA	University of Mascara
Supervisor	Mrs. Meddeber Meriem	MCA	University of Mascara

Academic Year: 2025 - 2026

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Allah, the Almighty, for granting me the strength, patience, and perseverance to complete this work.

I would like to extend my sincere appreciation to my supervisor, Dr. Meriem Meddeber, for her invaluable guidance, continuous support, and insightful advice throughout this research. Her expertise and encouragement have been essential to the completion of this thesis.

My sincere thanks also go to the President of the defense committee and to all the members of the examination committee for accepting to evaluate this work and for their time, effort, and constructive feedback.

A very special thanks to Dr. Soufiene Djahel, from the University of Coventry, UK, for his continuous support, guidance, and encouragement. His advice and assistance have been greatly appreciated.

I am profoundly grateful to my parents for their unconditional love, sacrifices, and constant encouragement. My heartfelt thanks also go to my wife and my children for their patience, understanding, and unwavering support throughout this journey.

Finally, to be sure not to forget anyone, I would like to thank all those, near or far, who contributed in any way to this work through their advice, support, or encouragement.

Abstract

The rapid expansion of the Internet of Things (IoT) has introduced significant security challenges due to heterogeneous devices, high-dimensional data, and evolving cyberattacks. Traditional intrusion detection systems (IDSs) often struggle to achieve high accuracy while maintaining computational efficiency in such environments. This thesis proposes an enhanced deep learning-based IDS that integrates a dual statistical feature selection framework to improve detection performance and reduce computational complexity. The proposed approach combines Pearson correlation and Mutual Information in a two-stage feature selection process to identify the most relevant features from IoT network traffic. This framework is integrated with a fully connected deep neural network designed to learn complex intrusion patterns. Two IDS architectures are evaluated using the NSL-KDD and RT-IoT2022 datasets. Experimental results show that the proposed system achieves high detection accuracy, exceeding 99% in most scenarios, while reducing training time. Comparative analysis with existing machine learning and deep learning approaches confirms the effectiveness and robustness of the proposed solution for IoT intrusion detection.

Keywords: IoT Security, Intrusion Detection, Deep Learning, Feature Selection, Network Security

Résumé

L'expansion rapide de l'Internet des objets (IoT) a entraîné d'importants défis en matière de sécurité en raison de l'hétérogénéité des appareils, des données à haute dimensionnalité et de l'évolution des cyberattaques. Les systèmes traditionnels de détection d'intrusion (IDS) ont souvent du mal à atteindre une grande précision tout en conservant une efficacité computationnelle dans de tels environnements. Cette thèse propose un IDS amélioré basé sur l'apprentissage profond qui intègre un double cadre de sélection des caractéristiques statistiques afin d'améliorer les performances de détection et de réduire la complexité computationnelle. L'approche proposée combine la corrélation de Pearson et l'information mutuelle dans un processus de sélection des caractéristiques en deux étapes afin d'identifier les caractéristiques les plus pertinentes du trafic réseau IoT. Ce cadre est intégré à un réseau neuronal profond entièrement connecté conçu pour apprendre des modèles d'intrusion complexes. Deux architectures IDS sont évaluées à l'aide des ensembles de données NSL-KDD et RT-IoT2022. Les résultats expérimentaux montrent que le système proposé atteint une précision de détection élevée, supérieure à 99 % dans la plupart des scénarios, tout en réduisant le temps de formation. Une analyse comparative avec les approches existantes d'apprentissage automatique et d'apprentissage profond confirme l'efficacité et la robustesse de la solution proposée pour la détection des intrusions dans l'IoT.

Mots clés: sécurité IoT, détection d'intrusion, apprentissage profond, sélection de caractéristiques, sécurité réseau

الملخص

أدى التوسع السريع في إنترنت الأشياء (IoT) إلى ظهور تحديات أمنية كبيرة بسبب الأجهزة غير المتجانسة والبيانات عالية الأبعاد والهجمات الإلكترونية المتطورة. غالبًا ما تواجه أنظمة كشف التسلل التقليدية (IDS) صعوبة في تحقيق دقة عالية مع الحفاظ على الكفاءة الحسابية في مثل هذه البيئات. تقترح هذه الأطروحة نظام IDS محسّنًا قائمًا على التعلم العميق يدمج إطار عمل مزدوج لاختيار الميزات الإحصائية لتحسين أداء الكشف وتقليل التعقيد الحسابي. تجمع الطريقة المقترحة بين ارتباط بيرسون والمعلومات المتبادلة في عملية اختيار الميزات على مرحلتين لتحديد الميزات الأكثر صلة من حركة مرور شبكة إنترنت الأشياء. يتم دمج هذا الإطار مع شبكة عصبية عميقة متصلة بالكامل مصممة لتعلم أنماط التسلل المعقدة. تم تقييم بنية IDS باستخدام مجموعتي بيانات NSL-KDD و RT-IoT2022. أظهرت نتائج التجارب أن النظام المقترح يحقق دقة كشف عالية، تتجاوز 99٪ في معظم السيناريوهات، مع تقليل وقت التدريب. يؤكد التحليل المقارن مع نهج التعلم الآلي والتعلم العميق الحالية فعالية ومتانة الحل المقترح لاكتشاف التسلل إلى إنترنت الأشياء.

الكلمات المفتاحية: أمن إنترنت الأشياء، كشف التسلل، التعلم العميق، اختيار الميزات، أمن الشبكات

Contents

List of Figures

List of Tables

List of Abbreviations

3G/4G	Third/Fourth Generation Mobile Network
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ARFF	Attribute-Relation File Format
Bi-LSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSV	Comma-Separated Values
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCNN	Deep Convolutional Neural Network
DDoS	Distributed Denial-of-Service
DL	Deep Learning
DNN	Deep Neural Network
DoS	Denial-of-Service
DSL	Digital Subscriber Line
DSSAE	Deep Stacked Sequence-to-Sequence Autoencoder
DT	Decision Tree
EO-FS	Equilibrium Optimizer-based Feature Selection
ETSI	European Telecommunications Standards Institute
F1	F1-Score (Harmonic Mean of Precision and Recall)
FC-DNN	Fully Connected Deep Neural Network
FN	False Negative
FP	False Positive
FS	Feature Selection
GPRS	General Packet Radio Service
GRU	Gated Recurrent Unit
IDS	Intrusion Detection System
IDSs	Intrusion Detection Systems
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IMFO	Improved Mayfly Optimization Algorithm
IoE	Internet of Everything
IoT	Internet of Things
KDD	Knowledge Discovery in Databases

k-NN	k-Nearest Neighbors
KNN	K-Nearest Neighbors
L-SVM	Linear Support Vector Machine
LDA	Linear Discriminant Analysis
LoRaWAN	Long Range Wide Area Network
LR	Logistic Regression
LSS	Logical Space Subdivision
LSTM	Long Short-Term Memory
LTE	Long-Term Evolution
M2M	Machine-to-Machine
MI	Mutual Information
ML	Machine Learning
MLP	Multi-Layer Perceptron
MQTT	Message Queuing Telemetry Transport
NB	Naive Bayes
NFC	Near Field Communication
NIDS	Network-based Intrusion Detection System
NSL-KDD	Network Security Laboratory – Knowledge Discovery in Databases
PCA	Principal Component Analysis
PS	Proposed Solution
Q-SVM	Quadratic Support Vector Machine
QDA	Quadratic Discriminant Analysis
R2L	Remote to Local (Attack Class)
ReLU	Rectified Linear Unit
RF	Random Forest
RFID	Radio Frequency Identification
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RPL	Routing Protocol for Low-Power and Lossy Networks
RT-IoT2022	Real-Time Internet of Things 2022 Dataset
SMOTE	Synthetic Minority Over-sampling Technique
SOA	Service-Oriented Architecture
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
U2R	User to Root (Attack Class)
UCI	University of California Irvine (Machine Learning Repository)

UNSW-NB15	University of New South Wales Network Benchmark 2015 Dataset
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

1.1. Background and Context

The Internet of Things (IoT) has emerged as one of the most trans-formative technological paradigms of the 21st century, fundamentally reshaping how we interact with our physical environment. The number of connected IoT devices is projected to continue its rapid expansion, with an estimated growth rate of approximately 14% in 2025. Current forecasts suggest the global IoT ecosystem will encompass nearly 39 billion devices by 2030, exceeding 50 billion by 2035[1]. This unprecedented connectivity promises enhanced efficiency, automation, and data-driven decision-making across virtually every sector of human activity. However, the rapid proliferation of IoT devices has introduced significant cybersecurity challenges that traditional security frameworks struggle to address. Unlike conventional computing systems, IoT devices are characterized by severe resource constraints, heterogeneous communication protocols, limited computational capabilities, and often inadequate built-in security mechanisms. These inherent limitations create a vast attack surface that malicious actors increasingly exploit to launch sophisticated cyber attacks, including distributed denial-of-service (DDoS) attacks, data breaches, and device hijacking.

The consequences of IoT security breaches extend far beyond individual privacy concerns. Critical infrastructure systems, healthcare networks, and smart city deployments rely heavily on IoT technologies, making them potential targets for nation-state actors and cybercriminal organizations. The 2016 Mirai botnet attack, which compromised hundreds of thousands of IoT devices to launch massive DDoS attacks, demonstrated the

[1] "Number of connected IoT devices growing 14\% to 21.1 billion." Accessed: Nov. 20, 2025. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/

catastrophic potential of IoT vulnerabilities. More recently, attacks on healthcare IoT systems during the COVID-19 pandemic highlighted the life-threatening implications of inadequate IoT security. Traditional network security approaches, designed for resource-rich environments with predictable traffic patterns, prove inadequate for the dynamic, heterogeneous, and resource-constrained IoT ecosystem. Conventional signature-based intrusion detection systems (IDS) struggle with the high volume of network traffic, the diversity of IoT protocols, and the rapidly evolving threat landscape. Similarly, rule-based security systems lack the adaptability required to detect novel attack patterns and zero-day exploits that frequently target IoT networks.

1.2. Problem Statement

The fundamental challenge in IoT intrusion detection lies in developing security mechanisms that can effectively identify malicious activities while operating within the severe constraints imposed by IoT environments. Existing intrusion detection systems face several critical limitations when applied to IoT networks:

- Scalability Issues: IoT networks produce massive volumes of varied data across diverse devices with varying communication patterns. Traditional IDS struggle to process this scale of information in real-time while maintaining acceptable detection accuracy.
- Feature Engineering Complexity: IoT network traffic exhibits unique characteristics that differ significantly from conventional network communications. The high dimensionality of IoT data, combined with irrelevant and redundant features, creates significant challenges for machine learning based detection systems.
- Resource Constraints: Many IoT devices operate with limited processing power, memory, and energy resources. Security solutions must be lightweight and efficient to be practically deployable in resource-constrained environments.

- Dynamic Threat Landscape: IoT attacks evolve rapidly, with new attack vectors and techniques emerging continuously. Static detection models quickly become obsolete, requiring adaptive approaches that can learn and evolve with changing threat patterns.
- False Positive Rates: High false positive rates in existing systems lead to alert fatigue and reduced trust in security mechanisms. This issue becomes especially challenging in IoT settings, where normal device operations may exhibit substantial variation.
- Real-time Detection Requirements: Multiple IoT applications, particularly in healthcare and industrial control systems, require real-time threat detection to prevent potentially catastrophic consequences. Existing systems often cannot meet these stringent latency requirements.

These challenges necessitate a paradigm shift toward intelligent, adaptive, and efficient intrusion detection mechanisms specifically designed for IoT environments. The integration of deep learning techniques with advanced feature selection methods presents a promising approach to address these limitations, but significant research gaps remain in optimizing such systems for IoT-specific requirements.

1.3 Research Objectives

The prime goal of this research is to develop a robust and efficient intrusion detection system (IDS) for IoT networks by combining advanced deep learning techniques with a novel hybrid feature selection framework. This study focuses on enhancing the accuracy, efficiency, and generalizability of IoT intrusion detection while addressing the computational challenges associated with resource-constrained IoT environments.

To achieve this goal, the research is structured around the following specific objectives:

1. Development of a Two Stage Filter-Based Feature Selection Framework: Design and implement a two-stage statistical feature selection methodology capable of identifying the most relevant and informative features from IoT network traffic. This framework aims to reduce the dimensionality of the input data, minimize computational overhead, and maintain feature interpretability, thereby facilitating more efficient and effective intrusion detection.
2. Design of an Optimized Deep Learning Architecture: Develop a tailored deep learning model optimized for IoT intrusion detection. The architecture is intended to learn complex patterns and relationships within network traffic data while ensuring computational efficiency, enabling its deployment in real-world IoT systems with limited resources.
3. Integration of Feature Selection with Deep Learning Models: Establish a seamless integration strategy that combines the dual statistical feature selection framework with the deep learning model. This integration is aimed at enhancing the system's detection performance, improving learning efficiency, and reducing the computational cost associated with processing high-dimensional network data.
4. Comprehensive Performance Evaluation: Conduct extensive experimental evaluations using multiple IoT datasets to rigorously assess the proposed IDS. The evaluation will focus on key performance indicators, including detection accuracy, false positive rates, computational efficiency, and scalability, providing evidence of the system's effectiveness and practical applicability in diverse IoT scenarios.

Through these objectives, this research seeks to provide a highly accurate, computationally efficient, and scalable IDS solution that can address the growing security challenges in IoT networks while contributing new methodologies for feature selection and deep learning integration in intrusion detection.

1.4 Research Questions

This research addresses the following fundamental questions:

• RQ1: How can feature selection techniques be optimized and combined to effectively reduce the dimensionality of IoT network data while preserving critical information necessary for accurate intrusion detection?

• RQ2: What deep learning architecture configurations are most suitable for processing IoT network traffic data and detecting various types of cyber attacks in resource-constrained environments?

• RQ3: How can hybrid feature selection be effectively integrated with deep learning models to achieve an optimal balance between detection accuracy and computational efficiency?

• RQ4: To what extent can the proposed dual filter-based feature selection combined with deep learning outperform existing state-of-the-art intrusion detection methods in terms of accuracy, precision, recall, and F1-score across different types of IoT attacks?

• RQ5: How does the proposed system perform across different network environments and attack scenarios, and what factors influence its adaptability and generalizability?

1.5. Scientific Contributions

The primary novel contributions of this research are:

1. A dual-stage hybrid feature selection framework that systematically combines Pearson correlation and Mutual Information in a complementary manner
2. An empirically-validated threshold selection strategy optimized for IoT traffic
3. Seamless integration architecture between the feature selection module and DNN

4. Comprehensive evaluation demonstrating considerable improvement over existing hybrid approaches

This research has produced two primary scientific contributions, which have been disseminated in peer-reviewed venues:

- A journal paper entitled *"Enhancing IoT Intrusion Detection: Deep Learning with Dual Statistical Feature Selection"*, published in the International Journal of Intelligent Engineering & Systems, vol. 18, no. 9, 2025.
- A conference paper entitled "*A Deep Neural Network-Based Intrusion Detection System for IoT Networks Using Feature Selection and Multi-Class Classification*", presented in The First IEEE/International Conference on Artificial Intelligence and Cyber Physical Systems AICPS'25, 02-03 November 2025, Mascara, Algeria

1.6. Thesis Organization

This dissertation is organized into five main chapters, each addressing specific aspects of the research work, as outlined below:

- Chapter 1: Introduction: This chapter introduces the research context, motivation, and significance of intrusion detection in IoT networks. It presents the research problem, objectives, and contributions, providing a clear overview of the study's scope and rationale.
- Chapter 2: Background and Theoretical Foundation: This chapter presents the essential background knowledge and theoretical concepts necessary for understanding the research. It covers IoT architectures, network security challenges, machine and deep learning principles, and feature selection techniques relevant to intrusion detection.

- Chapter 3: Related Work: A comprehensive review of existing research is presented in this chapter. It examines current machine learning and deep learning methodologies for intrusion detection systems, including techniques for selecting relevant features, highlighting their strengths, limitations, and gaps that motivate the proposed study.
- Chapter 4: Proposed Deep Learning-Based IDS Architecture and Methodology: This chapter details the design and development of the proposed IDS. It introduces the two proposed deep learning-based architecture, and the methodology for integrating feature selection with deep learning models to enhance detection performance.
- Chapter 5: Experimental Setup, Results and Discussion: The experimental evaluation of the proposed IDS is presented in this chapter. It includes the dataset description, implementation details, performance metrics, per-class evaluation, and comparative analysis with existing approaches. The discussion also highlights key insights from the results. Additionally, this chapter addresses the limitations of the proposed approach and outlines directions for future research, providing guidance for further improving IoT intrusion detection systems.
- General Conclusion: This final section summarizes the research contributions, discusses the limitations, and outlines directions for future work. It emphasizes the significance of the study's findings and their potential impact on IoT security.

Chapter 2

Background and Theoretical Foundation

2.1. The Internet of Things (IoT)

2.1.1. IoT Definition

The concept of the Internet of Things (IoT) was first introduced in 1999 by Kevin Ashton. At its core, the paradigm posits that ordinary objects —equipped with constrained computational capacity and energy resources— can be rendered "smart" through the integration of sensing, actuation, and communication capabilities. These heterogeneous devices are thereby enabled to interconnect autonomously, exchange data, collaborate toward shared objectives, and transmit collected environmental or operational information to broader network infrastructures. This interoperability is achieved through the use of diverse yet compatible wireless communication protocols, wherein computing and connectivity functions are invisibly and pervasively embedded within the physical objects themselves [1]. Consequently, a wide array of entities, including conventional electronic devices, mobile appliances, household items (e.g., refrigerators and dishwashers), environmental control systems, wearable textiles, foodstuffs, livestock, and even flora— may now be augmented with embedded sensing, computation, processing, and communication functionalities. By creating digital representations (counterparts) of virtually any physical object or phenomenon, the IoT enables these entities to interact and exchange information autonomously over diverse wireless technologies, such as RFID, ZigBee, wireless sensor networks (WSN), WLAN, NFC, DSL, GPRS, 3G/4G, LTE, and Bluetooth [2], [3]. Indeed, Numerous definitions and reference architectures for the Internet of Things (IoT) have been proposed by standardization bodies and industry consortia. Notably, per the Institute of Electrical and Electronics Engineers (IEEE), the IoT consists of objects with integrated sensors that communicate through the Internet [4].

According to the European Telecommunications Standards Institute (ETSI), the preferred terminology is "machine-to-machine (M2M)" communication instead of "Internet of Things." M2M is defined as a system where machines communicate, decide, and operate independently, without requiring direct human supervision [5]. The Cisco organization employs the term "Internet of Everything (IoE)," defining it as a networked ecosystem encompassing people, processes, data, and things. Within this framework, information is generated, analyzed, and acted upon as it flows across interconnected entities [6].

2.1.2. IoT Architecture

Over the past decade, a multitude of Internet of Things (IoT) architectures have been introduced in academic literature. For example, the middleware-based architecture, the Service Oriented Architecture (SOA-based), and the five-layer architecture. The latter categorizes the IoT framework into five distinct layers as shown in Figure 2.1: the perception, sensing, or device layer; the network, transmission, or communication layer; the middleware/service management layer; the application layer; and the business layer, as depicted. However, among the array of proposed architectures, the foundational (general) common model is recognized as the three-layer architecture, which comprises the perception layer, the network layer, and the application layer.

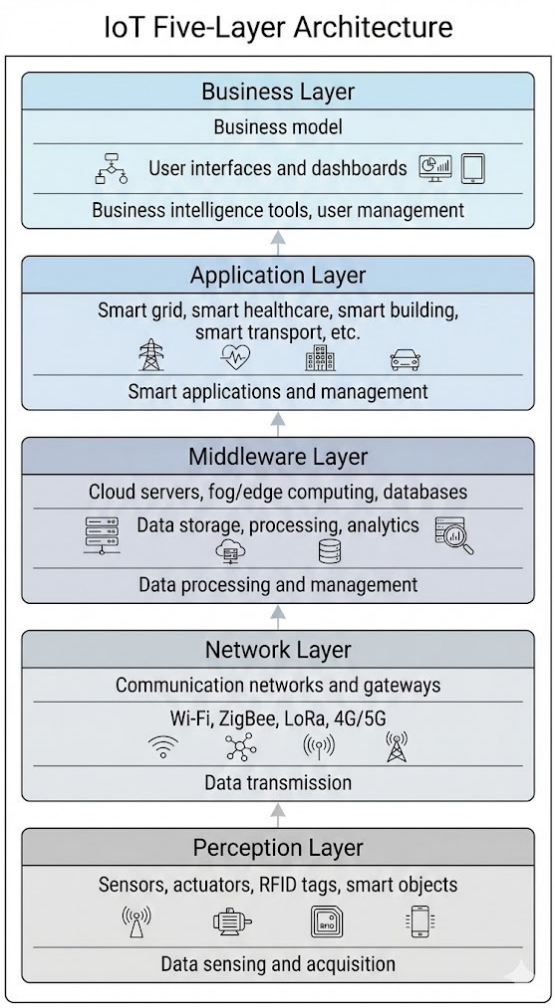


Figure 2.1. Five-Layer Architecture of the Internet of Things (IoT) [7]

2.1.3. IoT Characteristics

The Internet of Things (IoT) environments are characterized by several unique features that significantly influence their design and functionality. These characteristics include heterogeneity, scalability, resource constraints, mobility, and dynamic topologies, each presenting distinct challenges and opportunities for IoT applications. Understanding these aspects is crucial for developing effective IoT systems.

- Heterogeneity: IoT networks consist of a diverse array of devices, each differing in computational power, memory, operating systems, and communication protocols. This heterogeneity complicates interoperability and necessitates the development of strategies to manage communication across various device types effectively [8]. For instance, protocols like 6LoWPAN and RPL have been proposed to address these challenges, ensuring seamless data exchange among heterogeneous devices [9].
- Scalability: The scalability of IoT systems is another defining characteristic, as they can incorporate thousands to millions of devices. This vast scale requires robust architectures capable of managing numerous connections and data flows. The Logical Space Subdivision (LSS) system, for example, modularly configures devices to adapt to user requirements and environmental changes, enhancing scalability and flexibility [10].
- Resource Constraints: Many IoT devices operate on low-power microcontrollers, imposing strict energy limitations. This constraint necessitates energy-efficient protocols and mechanisms to optimize resource usage while maintaining performance. Research has highlighted the importance of balancing energy consumption with network efficiency to ensure the longevity of devices [11].
- Mobility: Mobility is a prevalent characteristic in IoT environments, as devices frequently change their location or network context. This dynamic nature requires adaptive protocols that can manage device mobility without compromising connectivity or data integrity. Solutions like LNR-PP and PMCAR have been explored to address these mobility challenges [11].
- Dynamic Topologies: The dynamic topologies of IoT networks, where connections evolve rapidly as devices join or leave, further complicate network management. This necessitates the development of adaptive algorithms that can quickly reconfigure network paths and maintain communication reliability [10].

While these characteristics present significant challenges, they also offer opportunities for innovation in IoT applications. For instance, the integration of cloud computing can enhance data analytics capabilities, allowing for more intelligent decision-making processes in real-time environments [11]. However, the complexity of managing such diverse and dynamic systems may lead to increased operational overhead and potential security vulnerabilities, which must be carefully addressed in future research.

2.1.4. Applications of IoT

The Internet of Things (IoT) encompasses a wide array of applications that significantly enhance various sectors by connecting devices and enabling data exchange. These applications range from healthcare, where IoT devices monitor patient vitals and facilitate remote consultations, to smart homes that automate tasks and improve energy efficiency. In agriculture, IoT supports precision farming through soil monitoring and automated irrigation systems. Transportation benefits from real-time tracking and predictive maintenance, while industrial IoT enhances manufacturing efficiency and safety [12], [13], [14]. Some of the most common applications of the IoT are:

Healthcare

- Remote Monitoring: Devices track vital signs and health metrics.
- Telemedicine: Facilitates remote consultations, improving access to care.

Smart Homes

- Automation: Controls lighting, heating, and security systems.
- Energy Management: Optimizes energy consumption through smart devices.

Agriculture

- Precision Farming: Sensors monitor soil conditions and automate irrigation.
- Crop Management: Data-driven insights enhance yield and resource use.

Transportation

- Fleet Management: Real-time tracking of vehicles for efficiency.
- Predictive Maintenance: Anticipates equipment failures to reduce downtime.

Industrial IoT

- Manufacturing E ciency: Streamlines processes and enhances safety.
- Supply Chain Optimization: Improves logistics through data analytics [15], [16].

While the benefits of IoT are substantial, challenges such as data security, interoperability, and privacy concerns remain critical issues that need addressing to fully realize its potential across these applications [14].

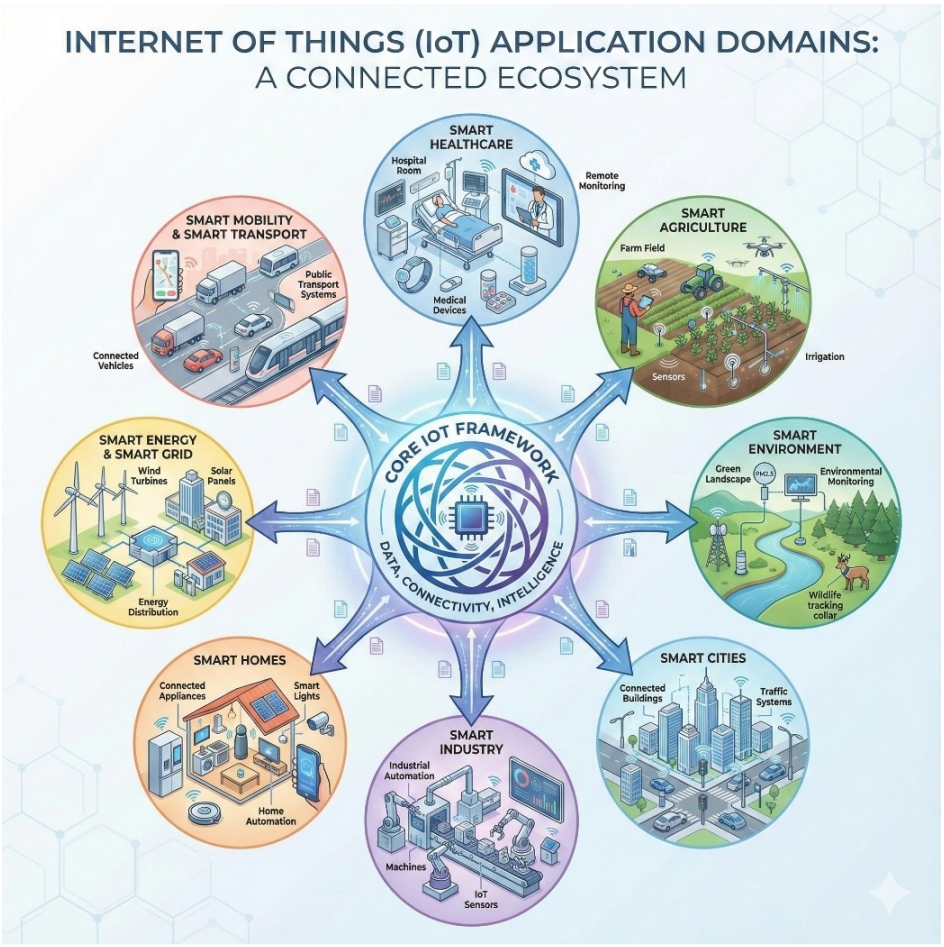


Figure 2.2. IoT application domains.

2.1.5. IoT Security Landscape and Challenges

Unlike traditional computing systems, IoT ecosystems are characterized by resource-constrained devices, heterogeneous communication protocols, and distributed architectures that complicate traditional security approaches [1].

2.1.5.1. Unique Characteristics of IoT Security

IoT security differs fundamentally from conventional network security due to several distinctive characteristics. First, the heterogeneity of IoT devices creates a complex ecosystem where devices with varying computational capabilities, operating systems, and

communication protocols must coexist securely [17]. Second, the resource constraints typical of IoT devices, including limited processing power, memory, and battery life, restrict the implementation of traditional security mechanisms [18]. Third, the scale and distributed nature of IoT deployments make centralized security management challenging, requiring novel approaches to threat detection and response [19].

The attack surface in IoT environments is significantly larger than traditional networks due to the sheer number of connected devices and their diverse communication pathways [20].Each IoT device represents a potential entry point for attackers, and the interconnected nature of these devices means that compromising a single device can provide access to the entire network [21]. This expanded attack surface is further complicated by the fact that many IoT devices are deployed in physically accessible locations, making them vulnerable to physical tampering and side-channel attacks [22].

2.1.5.2. IoT-Specific Vulnerabilities

Research has identified several categories of vulnerabilities that are particularly prevalent in IoT systems. Device-level vulnerabilities include weak authentication mechanisms, insufficient encryption, and inadequate firmware update processes [23]. Many IoT devices ship with default credentials that are never changed, creating easily exploitable entry points for attackers. Additionally, the use of lightweight cryptographic protocols, while necessary for resource-constrained devices, often introduces security weaknesses that can be exploited by sophisticated attackers [24].

Communication-level vulnerabilities arise from the diverse protocols used in IoT networks, including WiFi, Bluetooth, Zigbee, and LoRaWAN. Each protocol has its own security characteristics and potential weaknesses, and the interoperability requirements often force compromises in security implementations. Man-in-the-middle attacks, eavesdropping, and

protocol-specific exploits represent significant threats to IoT communication channels [25], [26].

Application and cloud-level vulnerabilities emerge from the integration of IoT systems with cloud services and mobile applications. Insecure APIs, inadequate data protection, and insufficient access controls in cloud platforms can expose IoT data and control functions to unauthorized access. The complexity of IoT ecosystems, which often involve multiple vendors and service providers, creates additional security challenges related to trust relationships and responsibility boundaries [27].

2.2. Intrusion Detection Systems (IDSs)

2.2.1. Evolution of IDS

Intrusion Detection Systems (IDS) have evolved significantly since their inception in the 1980s, progressing from simple signature-based systems to sophisticated machine learning-powered solutions. Traditional intrusion detection systems are typically classified into signature-based (misuse detection) and anomaly-based detection systems [28].

Signature-based systems rely on predefined patterns or signatures of known attacks, making them highly effective at detecting known threats but incapable of identifying novel or zero-day attacks [29].

Anomaly-based detection systems, in contrast, establish a baseline of normal behavior and flag deviations from this baseline as potential intrusions. While these systems can theoretically detect unknown attacks, they suffer from high false positive rates and the challenge of accurately defining normal behavior in dynamic environments [30], [31]. The evolution of IDS technology has been driven by the need to address these limitations while adapting to increasingly sophisticated attack techniques and changing network architectures.

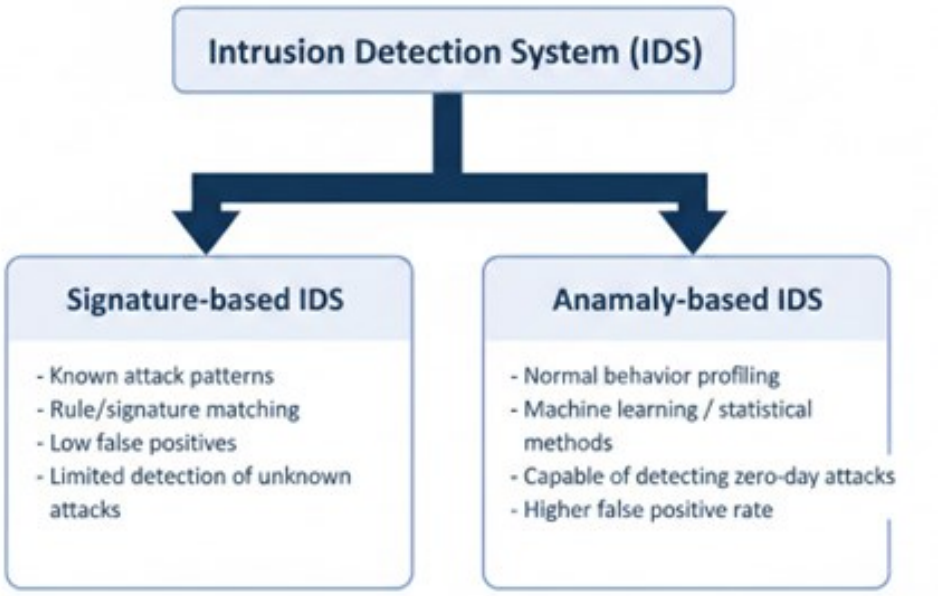


Figure 2.3. IDS Types.

2.2.2. Traditional IDS Limitations in IoT Contexts

Traditional IDS approaches face significant challenges when applied to IoT environments. The resource constraints of IoT devices make it impractical to deploy conventional IDS software directly on these devices [32]. The computational overhead of signature matching and statistical analysis can overwhelm the limited processing capabilities of typical IoT devices [33]. Additionally, the memory requirements for storing signature databases or maintaining behavioral baselines often exceed the available resources on IoT devices.

The heterogeneity of IoT networks presents another challenge for traditional IDS approaches. Signature-based systems require attack signatures specific to each device type and communication protocol, leading to an explosion in the number of signatures required [34]. Anomaly-based systems struggle to establish meaningful baselines across diverse device types with fundamentally different operational characteristics. The dynamic nature

of IoT networks, where devices frequently join and leave the network, further complicates baseline establishment and maintenance [35].

Network-based IDS (NIDS) approaches, while more feasible from a resource perspective, face challenges in monitoring the diverse communication protocols used in IoT networks. Many IoT protocols use encrypted communication, limiting the effectiveness of deep packet inspection techniques commonly used in traditional NIDS [36]. The distributed nature of IoT networks also means that no single monitoring point can observe all network traffic, requiring coordinated monitoring strategies [37].

2.2.3. Modern Approaches to IoT Intrusion Detection

Modern IoT intrusion detection approaches have emerged to address the limitations of traditional systems. Hybrid detection systems combine signature-based and anomaly-based approaches to leverage the strengths of both methodologies. These systems use signature-based detection for known threats while employing anomaly detection for unknown attacks, potentially reducing false positive rates while maintaining detection coverage [38].

Distributed and collaborative intrusion detection represents another significant advancement in IoT security. These approaches distribute detection responsibilities across multiple nodes in the network, enabling more comprehensive monitoring while distributing computational load [39]. Collaborative systems allow devices to share threat intelligence and coordinate responses, improving overall network security [40].

Edge computing approaches have gained prominence as a means of addressing the resource constraints of IoT devices while maintaining real-time detection capabilities. By deploying IDS functionality at edge nodes with greater computational resources, these systems can provide sophisticated detection capabilities without overwhelming individual IoT devices

[41]. This approach also reduces the latency associated with cloud-based detection systems and maintains functionality even when connectivity to central servers is compromised [42].

2.3. Machine Learning and Deep Learning in Cybersecurity

2.3.1. Machine Learning Foundations in Security

The application of machine learning (ML) to cybersecurity has transformed the field by enabling automated pattern recognition and adaptive threat detection [43]. Traditional rule-based security systems require manual creation and maintenance of detection rules, a process that becomes increasingly difficult as attack techniques evolve [44]. Machine learning approaches can automatically learn patterns from data, potentially identifying subtle indicators of malicious activity that might be missed by human analysts [45].

Supervised learning techniques, including decision trees, support vector machines, and ensemble methods, have been widely applied to intrusion detection problems. These approaches require labeled training data containing examples of both normal and malicious behavior, which can be challenging to obtain in sufficient quantity and quality [46]. The performance of supervised learning systems is heavily dependent on the representativeness of the training data and their ability to generalize to new, previously unseen attacks [47].

Unsupervised learning approaches, including clustering and outlier detection algorithms, address some limitations of supervised methods by not requiring labeled training data. These techniques can identify anomalous behavior patterns without prior knowledge of specific attack types, making them potentially effective against zero-day attacks. However, unsupervised methods often struggle with high false positive rates and the challenge of distinguishing between benign anomalies and genuine security threats [48], [49].

2.3.2. Deep Learning Revolution in Cybersecurity

Deep learning has emerged as a particularly promising approach for cybersecurity applications due to its ability to automatically learn complex, hierarchical feature representations from raw data [50]. Unlike traditional machine learning approaches that require manual feature engineering, deep learning models can discover relevant features automatically, potentially identifying subtle patterns that human experts might overlook [51].

Convolutional Neural Networks (CNNs) have shown effectiveness in analyzing network traffic patterns and detecting malicious activities. By treating network traffic as sequences or images, CNNs can identify spatial and temporal patterns indicative of attacks [52]. The hierarchical feature learning capability of CNNs enables them to capture both low-level packet characteristics and high-level behavioral patterns [53].

Recurrent Neural Networks (RNNs) and their variants, including Long Short-Term Memory (LSTM) networks, have proven particularly effective for sequence-based intrusion detection [54]. These architectures can model temporal dependencies in network traffic, enabling detection of attacks that unfold over time [55]. The ability to maintain memory of previous states makes RNNs well-suited for detecting multi-stage attacks and identifying patterns that span multiple time steps [56].

Autoencoders have gained attention for anomaly detection applications in cybersecurity. These neural networks learn to reconstruct normal behavior patterns and can identify anomalies as instances that cannot be accurately reconstructed. The unsupervised nature of autoencoder training makes them particularly attractive for scenarios where labeled attack data is scarce [57], [58].

2.3.3. Deep Learning Challenges in IoT Security

Deep learning techniques, while promising, present considerable difficulties when implemented for IoT security context. The computational requirements of deep neural networks often exceed the capabilities of resource-constrained IoT devices. Training DL models demands significant computational power and extensive datasets, which may not be available in IoT environments [59]. The memory footprint of trained models can also be prohibitive for deployment on IoT devices with limited storage capacity [60].

Adversarial attacks represent a significant concern for deep learning-based security systems. Attackers can potentially craft inputs designed to fool neural networks, causing them to misclassify malicious activities as benign. The vulnerability of deep learning models to adversarial examples raises questions about their robustness in adversarial environments where attackers actively attempt to evade detection [61], [62].

2.4. Feature Selection Techniques in Network Security

2.4.1. Importance of Feature Selection in Security Applications

Feature selection plays a crucial role in the effectiveness of machine learning-based security systems by identifying the most relevant attributes for threat detection while reducing computational complexity [63]. In network security applications, datasets often contain hundreds or thousands of features extracted from network traffic, system logs, and device behaviors [64]. Many of these features may be irrelevant or redundant, potentially degrading model performance and increasing computational requirements [65].

The interpretability of security models is also enhanced through effective feature selection. By focusing on a smaller set of relevant features, security analysts can better understand which characteristics of network traffic or device behavior are most indicative of threats

[66]. This interpretability is crucial for validating model decisions and developing appropriate response strategies [67].

2.4.2. Categories of Feature Selection Methods

Feature selection (FS) methods, as shown in Figure 2.3, can be broadly categorized into three main approaches: filter methods, wrapper methods, and embedded methods [68].

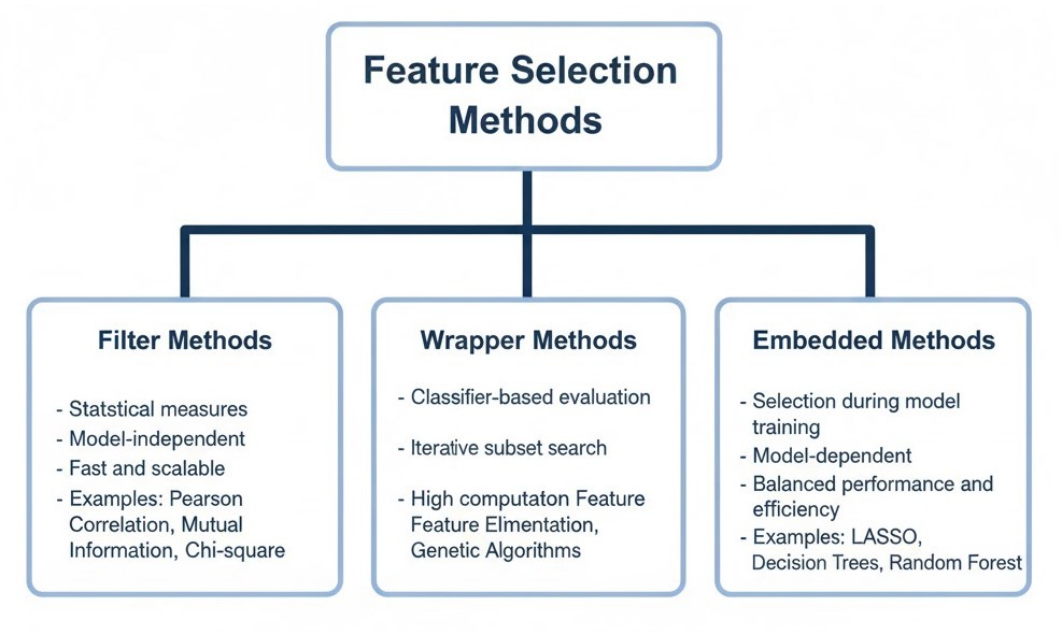


Figure 2.4. Categories of FS methods

Filter Methods: Filter methods evaluate features independently of the learning algorithm, using statistical measures to assess the relevance of each feature. These methods are computationally efficient and can be applied as a preprocessing step before model training [69]. Common filter methods include correlation-based feature selection, mutual information, chi-square tests, and information gain. Correlation-based methods identify features that are highly correlated with the target variable while having low correlation with other selected features [70]. Mutual information measures the amount of

information that one variable provides about another, making it effective for identifying non-linear relationships [71].

Wrapper methods: These methods evaluate feature subsets by training and testing the learning algorithm with different combinations of features. While computationally more expensive than filter methods, wrapper approaches can identify feature interactions and select features that work well together for the specific learning algorithm [72]. Common wrapper methods include forward selection, backward elimination, and genetic algorithms.

Embedded methods: integrate feature selection into the model training process, simultaneously learning the model parameters and identifying relevant features [73]. Regularization techniques such as L1 (Lasso) and L2 (Ridge) regression incorporate feature selection into the optimization objective [74]. Tree-based methods naturally perform feature selection by choosing the most informative features for splitting decisions [75].

2.4.3. Feature Selection Challenges in IoT Security

IoT security applications present unique challenges for feature selection due to the characteristics of IoT data and environments. The heterogeneity of IoT devices means that features relevant for one device type may be irrelevant or unavailable for others [76]. This heterogeneity complicates the development of universal feature sets that work across diverse IoT ecosystems [77].

Furthermore, The restricted computing capacity of IoT environments limits the sophistication of feature selection methods that can be utilized [78]. Sophisticated wrapper methods or genetic algorithms may be too computationally expensive for real-time application in IoT networks [79]. This constraint necessitates the development of

lightweight feature selection approaches that can operate within the resource limitations of IoT devices.

2.5. Summary

Chapter 2 provided the theoretical foundation for the development of the proposed intrusion detection system for IoT networks. It began with an overview of the Internet of Things (IoT), discussing its architecture, applications, and the unique security challenges posed by resource-constrained and heterogeneous devices. The chapter then introduced Intrusion Detection Systems (IDSs), highlighting different types, including signature-based, anomaly-based, and hybrid approaches, and their relevance for detecting malicious activities in IoT networks.

Next, the role of machine learning and deep learning in cybersecurity was explored, emphasizing their ability to automatically detect complex and previously unseen attacks. Popular algorithms and architectures, such as neural networks, CNNs, and RNNs, were presented in the context of IoT intrusion detection. Finally, feature selection techniques were discussed as essential tools for reducing data dimensionality, improving interpretability, and enhancing computational efficiency, which are critical in high-dimensional IoT datasets.

Overall, this chapter established the key concepts and background knowledge that underpin the proposed IDS, linking IoT characteristics, security challenges, deep learning approaches, and feature selection strategies, and setting the stage for the methodology presented in the following chapters.

Chapter 3

Related Work

3.1. Introduction

This chapter provides a critical review of the existing literature on intrusion detection systems for IoT networks. It examines both traditional machine learning and modern deep learning approaches, evaluating their effectiveness, limitations, and suitability for IoT environments. The chapter also analyzes feature selection methods applied in network security, highlighting how they impact detection accuracy and computational efficiency. By systematically assessing the strengths and weaknesses of prior studies, this chapter identifies gaps in the current research and motivates the development of the proposed deep learning-based IDS, ensuring that the methodology addresses the limitations observed in previous works.

3.2. Machine Learning-Based IDS for IoT

Machine learning techniques have risen to the forefront of IoT security solutions, distinguished by their ability to discover sophisticated patterns within data automatically, adapt to evolving threats, and handle the high-dimensional, heterogeneous nature of IoT network traffic. This section explores how different machine learning techniques are applied to detecting intrusions in IoT systems, organized according to their learning methodologies.

3.2.1. Supervised Learning Approaches

Supervised learning algorithms, which learn from labeled training data containing examples of both normal traffic and various attack types, have been extensively applied to IoT intrusion detection. Common supervised learning algorithms employed in this

context include Support Vector Machines (SVM), Decision Trees (DT), Random Forest (RF), k-Nearest Neighbors (k-NN), Naive Bayes (NB), Logistic Regression (LR), and various ensemble methods.

Abdulla et al. [80] delivered a detailed overview of IoT intrusion detection systems using supervised machine learning, identifying tree-based algorithms (Decision Trees, Random Forest, and boosting methods) as top performers in many studies. The review highlights that machine learning algorithm performance varies significantly with dataset characteristics, feature selection, and classification type [80].

Support Vector Machines have been widely employed due to their effectiveness in high-dimensional spaces and their ability to handle non-linear decision boundaries through kernel functions. However, SVMs can be computationally expensive for large-scale datasets and require careful hyperparameter tuning. Albulayhi et al. [81] evaluated four machine learning algorithms (Logistic Regression, SVM, Decision Tree, and Artificial Neural Network) for classification purposes, comparing their performance via Receiver Operating Characteristic (ROC) curves on IoT-specific datasets including Bot-IoT and IoTID20.

Random Forest and other ensemble methods have demonstrated particularly strong performance in IoT intrusion detection tasks [82]. These methods combine multiple decision trees to improve generalization and reduce overfitting, often achieving high accuracy with relatively modest computational requirements during inference. The interpretability of tree-based methods also offers advantages in understanding which features contribute most to detection decisions [83].

Despite their successes, supervised learning approaches face several challenges in IoT contexts [84]. The need for extensive labeled datasets for training poses a significant challenge, as obtaining representative labeled datasets covering diverse IoT devices, protocols, and attack types is resource-intensive and time-consuming [85]. The class

imbalance problem, where attack instances are vastly outnumbered by normal traffic, can lead to biased models that perform poorly on minority attack classes [86].

3.2.2. Unsupervised Learning Approaches

Unsupervised learning algorithms, which do not require labeled training data, offer potential advantages in IoT intrusion detection by reducing the labeling burden and enabling detection of previously unknown attack patterns [87]. Common unsupervised approaches include clustering algorithms such as k-Means, DBSCAN, and hierarchical clustering, as well as dimensionality reduction techniques like Principal Component Analysis (PCA) and autoencoders.

Clustering-based approaches operate by grouping similar traffic patterns together, with the assumption that attack traffic will form distinct clusters separate from normal traffic [88]. These methods can potentially detect novel attacks that deviate significantly from normal behavior patterns. However, the effectiveness of clustering-based detection depends critically on the choice of distance metrics, clustering algorithms, and the ability to distinguish between legitimate anomalies and actual attacks [89].

Despite their advantages, unsupervised learning approaches typically achieve lower detection accuracy compared to supervised methods and generate higher false positive rates. Alsoufi et al. [90] found that supervised deep learning techniques offer better performance compared to unsupervised and semi-supervised learning in IoT intrusion detection contexts. The lack of labeled data makes it difficult to evaluate and tune unsupervised models, and distinguishing between benign anomalies and actual attacks remains challenging.

3.2.3. Performance Limitations of ML-Based Approaches

While machine learning-based IDS have demonstrated promising results in experimental settings, several performance limitations and scalability issues constrain their practical deployment in real-world IoT environments. These limitations can be categorized in the following key points:

Computational Complexity: Many machine learning algorithms, particularly ensemble methods and kernel-based approaches, incur significant computational costs during both training and inference. For instance, Darley et al. [91] found that data preprocessing, feature extraction, model training, and deployment of ML-based IDS increase computational complexity and resource requirements, demanding greater CPU, memory, and energy resources. This is particularly problematic for resource-constrained IoT devices that cannot support complex models.

Scalability Challenges: The massive scale of IoT deployments poses significant challenges for machine learning-based detection systems. Training models on large-scale datasets can be time-consuming and resource-intensive. Ashraf et al. [92] note that model training can be time-consuming and that increasing network scale impacts performance. Real-time detection requirements further constrain the complexity of models that can be deployed.

Generalization and Overfitting: Machine learning models trained on specific datasets may not generalize well to different IoT environments, device types, or attack patterns. Haripriya [93] highlights overfitting as a noted issue, particularly with commonly used benchmark datasets. The diversity of IoT devices and protocols makes it challenging to develop universal models that perform well across varied deployment contexts.

Dataset Limitations: The effectiveness of machine learning-based IDS is fundamentally limited by the quality and representativeness of available training datasets. Abdulla et al.

[80] emphasize the lack of recent datasets collected from real IoT environments, noting that many existing datasets are imbalanced or not captured from actual IoT deployments. This gap between experimental datasets and real-world traffic patterns can lead to poor performance when models are deployed in production environments.

Adversarial Robustness: Machine learning models can be vulnerable to adversarial attacks where malicious actors deliberately craft inputs to evade detection. In this context, Darley et al. [91] note that ML can also be used for adversarial attacks, highlighting the arms race between attackers and defenders. Ensuring robustness against adversarial manipulation remains an open challenge.

These limitations have motivated research into more sophisticated approaches, particularly deep learning techniques that can automatically learn hierarchical feature representations and potentially achieve better generalization, as well as feature selection methods that can reduce computational complexity while maintaining detection accuracy [94] [95].

3.3. Deep Learning-Based IDS for IoT

Deep learning (DL), a subset of machine learning based on artificial neural networks with multiple layers, has emerged as a particularly promising approach for IoT intrusion detection. Deep learning approaches automatically derive layered feature abstractions from original data, which may reveal complex relationships that are difficult to capture through hand-crafted rules. The present section discusses how different deep learning approaches are applied to detect intrusions within IoT networks.

3.3.1. Deep Neural Networks (DNN)

Deep Neural Networks, consisting of multiple fully connected layers with non-linear activation functions, represent the foundational architecture of deep learning [96]. DNNs have been applied to IoT intrusion detection as classifiers that learn to map network traffic features to attack categories.

Several studies have demonstrated the effectiveness of DNNs for IoT intrusion detection. Haripriya [93] reports that deep learning models including DNNs achieved high accuracy rates, with some configurations reaching 99.996% accuracy on the KDD99 dataset. The ability of DNNs to learn non-linear decision boundaries and complex feature interactions makes them well-suited for distinguishing between normal traffic and various attack types.

However, DNNs face several challenges in IoT contexts. The large number of parameters in deep networks requires substantial computational resources for training and inference, which may exceed the capabilities of resource-constrained IoT devices [97]. DNNs are also prone to overfitting, particularly when training data is limited or imbalanced [98]. Furthermore, the "black box" nature of DNNs limits interpretability, making it difficult to understand why specific traffic is classified as malicious.

3.3.2. Convolutional Neural Networks (CNN)

Initially designed for analyzing images, CNNs have been repurposed for detecting network intrusions by interpreting traffic characteristics as spatial data representations. CNNs employ convolutional layers that can automatically learn local feature patterns and pooling layers that provide translation invariance and dimensionality reduction. The application of CNNs to IoT intrusion detection has shown promising results. Sulaiman et al. [99] identify CNNs as one of the key DL techniques for detecting anomalies in IoT security systems, highlighting their effectiveness in extracting complex patterns from high-

dimensional data. Haripriya [93] reports that Deep Convolutional Neural Networks (DCNN) achieved 99.996% accuracy on the KDD99 dataset, demonstrating the strong performance potential of CNN architectures.

However, CNNs also face challenges in IoT intrusion detection. The spatial structure assumption underlying CNNs may not always align well with the sequential or tabular nature of network traffic data. Careful design of input representations is required to effectively leverage CNN architectures. Additionally, deep CNN architectures still require substantial computational resources, particularly during training [100].

3.3.3. Recurrent Neural Networks and LSTM

Recurrent Neural Networks (RNN) and their variants, particularly Long Short-Term Memory (LSTM) networks, are specifically designed to handle sequential data by maintaining internal state information across time steps. This makes them particularly well-suited for analyzing network traffic, which inherently has temporal dependencies [101]. The vanishing gradient challenge encountered in conventional RNNs is mitigated through the use of LSTM architectures, enabling them to learn long-term dependencies in sequential data. Several research efforts have validated the efficac of LSTM-based approaches for IoT intrusion detection. Sulaiman et al. [99] highlight LSTM networks as a key technique for anomaly detection in IoT security systems. Haripriya [93] reports that LSTM achieved 96.3% accuracy for detecting MQTT-based attacks, demonstrating effectiveness in protocol-specific intrusion detection.

In another research, Vaiyapuri et al. [102] survey deep learning approaches for intrusion detection in Industrial IoT (IIoT) networks, reporting that LSTM and other deep learning models achieved accuracies up to 99.967% and 100% recall rates across various datasets including NSL-KDD and UNSW-NB15. The study emphasizes the potential of LSTM-based approaches to mitigate cyberattacks in IIoT environments.

Hybrid architectures combining LSTM with other neural network components have also been explored. Alruwaili [103] proposes a hybrid model combining Long Short Term Memory with Recurrent Neural Network (LSTM-RNN) for attack detection and classification in IoT environments, demonstrating superior performance over existing models. Aldaej et al. [104] develop an edge-cloud-based IoT IDS using Bidirectional LSTM (Bi-LSTM) combined with RNN, achieving 99.56% accuracy, 99.45% precision, 98.25% recall, and 99.12% F1-measure on the BoT-IoT dataset.

Despite their advantages, LSTM-based approaches face computational challenges. The sequential nature of RNN and LSTM processing makes parallelization difficult, leading to longer training times compared to feedforward architectures. The computational requirements of LSTM networks can be prohibitive for resource-constrained IoT devices, necessitating edge or cloud-based deployment strategies [105] [101].

3.3.4. Hybrid Deep Learning Architectures

Recognizing that different deep learning architectures have complementary strengths, researchers have increasingly explored hybrid architectures that combine multiple neural network components. These hybrid approaches aim to leverage the spatial feature extraction capabilities of CNNs, the temporal modeling capabilities of RNNs/LSTMs, and the classification power of fully connected layers [106].

In the literature, several hybrid architectures have been proposed. For instance, Sulaiman et al. [99] highlight hybrid models as an important category of deep learning techniques for IoT security, noting their effectiveness when combined with appropriate preprocessing techniques such as feature selection and PCA. The review emphasizes that hybrid approaches can address the challenges posed by data volume and diversity in IoT networks.

Duraibi et al. [107] present a hybrid approach combining feature selection with deep learning for cyberattack detection in IoT environments. The study develops an Improved Mayfly Optimization Algorithm with Hybrid Deep Learning based Intrusion Detection (IMFOHDL-ID) approach that integrates IMFO-based feature selection with Long Short Term Memory based Deep Stacked Sequence-to-Sequence Autoencoder (LSTM-DSSAE) for intrusion detection. The hybrid approach demonstrated improved performance over existing methods.

Hybrid architectures offer the potential to achieve superior performance by combining the strengths of different neural network components [170]. However, they also introduce additional complexity in terms of architecture design, hyperparameter tuning, and computational requirements [171]. The increased model complexity can exacerbate overfitting risks and make deployment on resource-constrained devices more challenging [172].

3.3.5. Advantages and Challenges of DL-Based Approaches

Deep learning approaches offer several significant advantages for IoT intrusion detection compared to traditional machine learning methods.

Advantages:

1. Automatic Feature Learning: Deep learning models can automatically learn hierarchical feature representations from raw or minimally preprocessed data, reducing the need for manual feature engineering. For instance, Al-Haija et al. [108] emphasize that deep learning models can identify complex patterns and characteristics by utilizing artificial neural networks, automatically building hierarchical representations from data, resulting in more precise and efficient intrusion detection.

2. Detection of Complex Patterns: The non-linear transformations and multiple layers in deep networks enable detection of complex, subtle patterns that may be missed by shallow learning approaches. Tsimenidis et al. [109] highlight deep learning's data-driven, anomaly-based approach and its ability to detect emerging, unknown attacks, addressing the inefficiency of old strategies against large-scale IoT architectures and novel threats.
3. Superior Performance: Numerous studies have reported that deep learning approaches achieve higher detection accuracy compared to traditional machine learning methods. Alsoufi et al. [110] emphasize that deep learning is superior for anomaly intrusion detection in IoT, achieving high detection accuracy and low false alarm rates compared to traditional shallow learning approaches.

Challenges:

Despite these advantages, deep learning-based IDS face several significant challenges that limit their practical deployment in IoT environments.

1. **Computational Cost:** Deep learning models, particularly deep architectures with many layers and parameters, require substantial computational resources for both training and inference. Vaiyapuri et al. [102] identify high computational cost as a key limitation of deep learning methods on resource-constrained IIoT devices, noting the need for specialized tools to simulate IIoT environments for network traffic capture and experiment evaluation. This computational burden is particularly problematic for IoT devices with limited processing power, memory, and energy budgets.
2. **Training Data Requirements:** Deep learning models typically require large amounts of training data to achieve good performance and avoid overfitting. Vaiyapuri et al. [102] also note the lack of sufficient network traffic samples for training deep

learning models as a significant challenge. Obtaining representative training datasets covering diverse IoT devices, protocols, and attack types is resource-intensive.

3. **Overfitting:** The large number of parameters in deep networks makes them prone to overfitting, particularly when training data is limited or imbalanced. Alsoufi et al. [110] highlight that building an effective anomaly intrusion detection system requires comprehending complex structures from noisy data, identifying dynamic anomaly patterns, and detecting anomalies without sufficient labels, all of which contribute to overfitting risks.
4. **Lack of Explainability:** Deep learning models are often criticized as "black boxes" with limited interpretability. Understanding why a particular traffic pattern is classified as malicious is difficult, which can hinder trust, debugging, and regulatory compliance. Haripriya [93] notes in this context that Explainable AI (XAI) approaches are being integrated to provide interpretation and explanation, indicating a recognized need for greater interpretability in deep learning-based IDS.
5. **Real-Time Deployment Challenges:** The computational requirements of deep learning models can make real-time detection challenging, particularly in resource-constrained IoT environments. Abdulla et al. [80] emphasize the need for real-time and lightweight IDS with less detection time and resource consumption, highlighting the gap between deep learning capabilities and practical deployment requirements.
6. **Adversarial Vulnerability:** Deep learning models can be vulnerable to adversarial attacks where carefully crafted perturbations to input data cause misclassification. Alfahaid et al. [111] identify adversarial vulnerabilities as a key limitation of current ML approaches, noting that future research should address privacy-preserving ML, explainable AI, and edge-based security frameworks.

These challenges have motivated research into optimization techniques including model compression, pruning, quantization, knowledge distillation, and feature selection methods that can reduce computational complexity while maintaining detection accuracy. Hassan et al. [112] explore lightweight deep learning techniques for intrusion detection in IoT networks, discussing methods like pruning, quantization, clustering, and collaborative optimization to address computational requirements in resource-constrained environments.

3.4. Feature Selection and Dimensionality Reduction Techniques in IDS

Feature selection (FS) and dimensionality reduction techniques play a crucial role in enhancing the effectiveness and efficiency of intrusion detection systems, particularly in resource-constrained IoT environments [113]. By identifying and retaining only the most relevant features while eliminating redundant or irrelevant ones, these techniques can reduce computational complexity, improve detection accuracy, mitigate overfitting, and enable deployment on resource-limited devices [114] [115].

Mukhaini et al. [116] conduct a systematic literature review of lightweight detection approaches in IoT networks, finding that filter-based feature engineering, particularly correlation algorithms for feature selection, are frequently employed due to their performance and lightness. The study highlights that filter methods are preferred in IoT contexts because they are computationally efficient and can be applied as a preprocessing step before model training.

Hassan et al. [112] discusses wrapper methods as one of the feature selection approaches for intrusion detection in IoT networks, noting their potential for improved accuracy but also highlighting the computational challenges they introduce in resource-constrained environments.

Several hybrid feature selection approaches have been proposed for IoT intrusion detection. Duraibi et al. [107] develop a hybrid approach combining Improved Mayfly Optimization Algorithm (IMFO) for feature selection with Long Short Term Memory based Deep Stacked Sequence-to-Sequence Autoencoder (LSTM-DSSAE) for intrusion detection. The IMFO-based feature selection method demonstrated effectiveness in selecting optimal feature subsets, contributing to improved detection performance.

Alruwaili [103] proposes a hybrid model that utilizes Equilibrium Optimizer-based Feature Selection (EO-FS) technique to select an optimal subset of features, combined with LSTM-RNN for classification. The hybrid approach demonstrated superior performance over existing models, effectively recognizing False Data Injection attacks in IoT networks.

Sulaiman et al. [99] highlight the significance of feature selection as a preprocessing technique essential for handling imbalanced and distributed IoT data, noting that it should be combined with other techniques such as PCA and oversampling methods like SMOTE for optimal results.

However, the feature selection process itself introduces computational overhead, particularly for wrapper and some hybrid methods [117]. The trade-off between the cost of feature selection and the benefits of reduced dimensionality must be carefully considered. For IoT applications, filter methods and computationally efficient embedded methods are often preferred due to their lower overhead [118].

3.5. Comparative Analysis and Limitations of Existing Works

This section provides a comparative analysis of major studies in IoT intrusion detection, synthesizing their methodologies, datasets, performance metrics, and limitations. Table 3.1 summarizes key characteristics of representative works reviewed in this chapter.

Study	Year	Methodology	Key Techniques	Datasets	Best Performance	Limitation
Sulaiman et al. [99]	2025	Survey of ML/DL for anomaly detection	CNN, LSTM, Autoencoders, Feature Selection, PCA, SMOTE	Various IoT datasets	Survey paper	Highlights need for handling imbalanced data and distributed processing
Al-Haija et al. [119]	2024	Survey of DL-based IDS	Deep learning architectures, supervised/unsupervised methods	Various IoT datasets	Survey paper	Dynamic IoT environment challenges existing IDS solutions
Haripriya [93]	2023	Survey of ML/DL IDS	RF, CNN, LSTM, RNN, DNN, PCA, Two-Stage DL	KDD-CUP, NSL-KDD, MQTT datasets	RF: 99.65% (KDD-CUP), DCNN: 99.996% (KDD99), LSTM: 96.3% (MQTT)	Data imbalance, overfitting, need for efficient models, interpretability challenges
Mukhaini et al. [116]	2024	Systematic review of lightweight IDS	ML/DL with filter-based feature engineering, correlation algorithms	Various IoT datasets	Survey paper	Resource constraints hinder complex system integration
Mishra et al. [120]	2021	Systematic review of IoT security	IDS/IPS models, ML/DL for DDoS mitigation	Various	Survey paper	Resource constraints hinder

						complex system integration
Vaiyapuri et al. [102]	2021	Survey of DL for IIoT IDS	CNN, RNN, LSTM, DNN, DBN, Autoencoders	NSL-KDD, UNSW-NB15	Up to 99.967% accuracy, 100% recall	High computational cost, lack of network traffic samples, need for lightweight models
Abdulla et al. [80]	2023	Review of supervised ML IDS	LR, NB, ANN, SVM, DT, RF, Ensemble Learning	IoTID20 (most recent)	Tree-based algorithms top performers	Lack of real IoT datasets, high resource consumption, need for lightweight real-time IDS
Tsimenidis et al. [109]	2022	Review of DL in IoT IDS	Deep learning models classified by type	Various	Survey paper	Large-scale IoT architecture and novel threats render old strategies inefficient
Alsoufi et al. [110]	2021	Survey of anomaly-based DL IDS	Deep learning techniques	Various IoT datasets	DL superior to shallow learning	Building effective systems from noisy data, insufficient

						labels, large computation demands
Ashraf et al. [92]	2022	Survey of ML/DL IDS	CNN, RNN, DNN, LSTM, GRU, SVM, RF, NB, KNN, XGBoost	Various	Federated learning improves privacy	Resource constraints, communication overhead, privacy concerns, lack of real-world verification
Darley et al. [91]	2022	Systematic review of ML IDS	ML-based IDS, DL methods	Various	DL superior to traditional ML	Increased computational complexity, changing attack nature, trade-offs between accuracy and false positives

Table 3.1. Comparative Overview of Existing Studies.

3.6. Summary

Chapter 3 presented a critical review of existing research on IDS, focusing on IoT networks. Both traditional ML and modern DL approaches were examined, highlighting their effectiveness, limitations, and applicability in resource-constrained IoT environments. The chapter also analyzed various feature selection techniques and their impact on model performance and computational efficiency. By systematically evaluating prior studies, key research gaps and challenges were identified, providing the foundation and motivation for the design of the proposed DL-based intrusion detection system presented in the following chapters.

Chapter 4

Proposed Deep Learning-Based IDS Architecture and Methodology

4.1. Introduction

This chapter presents the proposed Intrusion Detection System (IDS) architectures and the associated methodological framework developed in this doctoral research. In order to address both general network intrusion detection and realistic IoT-specific security challenges, two distinct yet conceptually related IDS architectures are proposed. Both architectures are derived from a unified detection pipeline; however, they differ in their feature selection strategies to accommodate the intrinsic characteristics of the datasets employed.

The first architecture is designed using the NSL-KDD dataset and relies on Pearson correlation–based feature selection. This architecture primarily targets generalizability and benchmarking against established IDS solutions. The second architecture is tailored to realistic IoT environments using the RT-IoT2022 dataset and employs a hybrid feature selection strategy that combines Pearson correlation and Mutual Information to represent relationships among features that are linear as well as non-linear dependencies. This dual-architecture design reflects a dataset-aware methodology and demonstrates adaptability to heterogeneous traffic characteristics.

4.2. Global IDS Framework

Despite their differences, both proposed architectures share a common global IDS framework composed of four fundamental stages:

1. Data Preprocessing
2. Feature Selection

3. Deep Learning Model
4. Intrusion Classification and Decision Output

This unified framework ensures methodological consistency while allowing flexibility in feature selection strategies. The modular design enables each component to be optimized independently without altering the overall detection logic. The materials and methodology of both architectures will be discussed in detail in the next section.

4.3. Materials and Methodology

This section describes the datasets and methodological framework used to develop the IDS implemented through a deep learning framework aimed at multi-class intrusion detection in IoT environments.

4.3.1. NSL-KDD dataset

The NSL-KDD dataset is an improved version of the widely used KDD Cup'99 dataset [46], in which several known limitations have been addressed by eliminating redundant and duplicate instances. Over the past decade, a significant proportion of intrusion detection studies have relied on the KDD Cup'99 dataset for performance evaluation [121]. NSL-KDD is publicly available in CSV and ARFF formats through the official website of the Canadian Institute for Cybersecurity[2]. In this study, the KDDTrain+ and KDDTest+ subsets, containing 125,973 and 22,544 records respectively, are employed. A summary of these datasets is presented in Figure 4.1, and the distribution of the different attack categories are illustrated in Table 4.1.

[2] https://www.unb.ca/cic/datasets/nsl.html

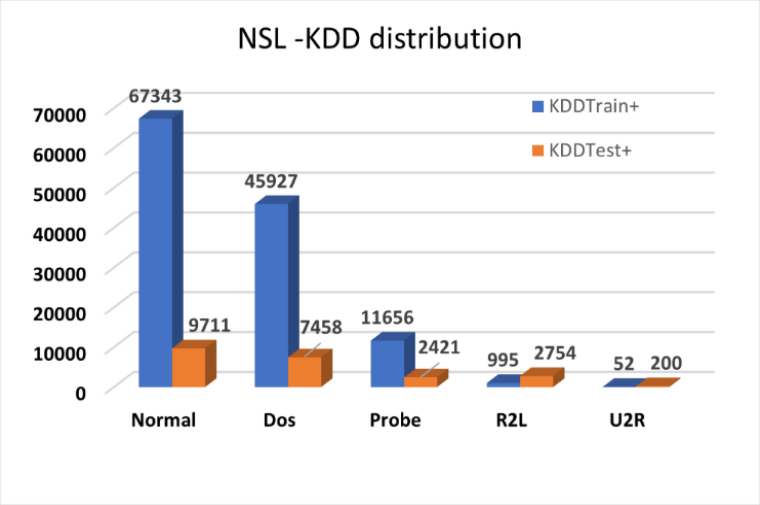


Figure 4.1. Details of the NSL-KDD dataset

Class	Attacks
DoS	'apache2', 'back', 'land', 'neptune', 'mailbomb', 'pod', 'processtable','smurf', 'teardrop', 'udpstorm', 'worm'
R2L	ftp_write', 'guess_passwd', 'httptunnel', 'imap', 'multihop', 'named','phf', 'sendmail', 'snmpgetattack', 'snmpguess', 'spy', 'warezclient', 'warezmaster', 'xlock', 'xsnoop'
Probe	'ipsweep', 'mscan', 'nmap', 'portsweep', 'saint', 'satan'
U2R	'buffer_overflow', 'loadmodule', 'perl', 'ps', 'rootkit', 'sqlattack', 'xterm'

Table 4.1. Distribution of the different attack classes in the NSL-KDD dataset

4.3.2. RT-IoT2022 dataset

This research also employs the RT-IoT2022 dataset, an open-source dataset publicly available through the UCI Machine Learning Repository[3]. The dataset was constructed

[3] https://archive.ics.uci.edu/dataset/942/rt-iot2022

using extensive real-time network traffic collected from operational IoT devices deployed in realistic environments, including Amazon Alexa and MQTT-based systems. It represents network behavior using 83 input features and a single output label indicating the type of attack. The dataset comprises a total of 123,117 instances [122]. A detailed breakdown of normal and malicious traffic distributions is presented in Figure 4.2.

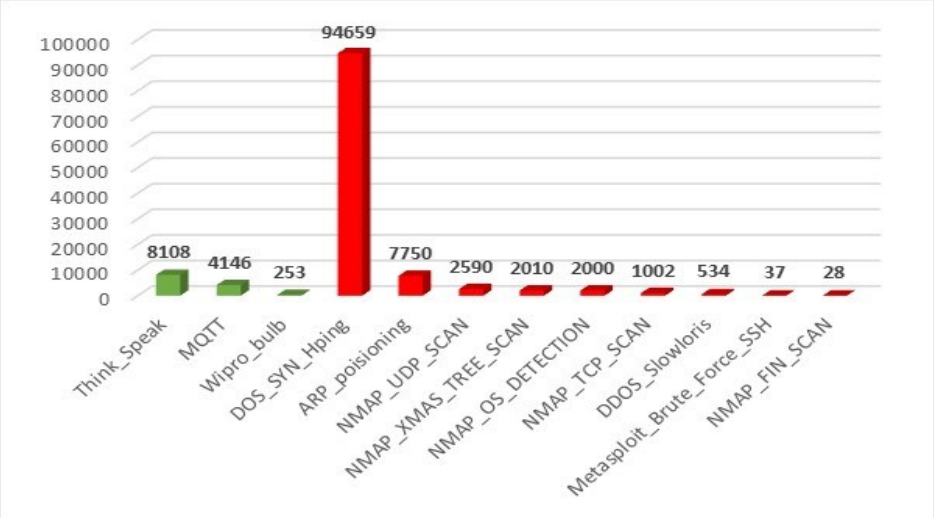


Figure 4.2. Details of the RT-IoT2022 dataset

4.3.3. Data Preprocessing

Data preprocessing is a fundamental step in intrusion detection, as raw network traffic often contains heterogeneous features, noise, and inconsistencies that can negatively impact learning performance [123]. Proper data processing ensures reliable feature representation and improves the effectiveness of the proposed IDS. This module includes data normalization and one-hot encoding.

4.3.3.1. Data Normalization

Normalization serves as a key data preprocessing step in machine learning, whereby the values of numerical features are scaled to fall within a common range. This technique

ensures that all features contribute on a comparable basis while maintaining their relative relationships within the dataset. It is especially valuable when the original features display substantially different scales or ranges of variation [124]. Through this scaling process, normalization helps prevent features with larger absolute values from exerting a disproportionate influence on the learning algorithm.

The Standard Scaler, which uses a normalized distribution that centers the data at 0 and scales it to a variance of 1, was used to normalize the dataset. This method's mathematical expression is given in the equation (1):

$$\tilde{x} = \frac{x - MEAN}{\sigma} \tag{1}$$

Here, $\tilde{x}$ represents the normalized feature, scaled to lie within the range $[-1,1]$, while MEAN denotes the feature's average value and σ corresponds to its standard

4.3.3.2. One-Hot Encoding

As artificial neural networks require exclusively numerical inputs, categorical (non-numeric) features must undergo appropriate encoding [125]. To this end, one-hot encoding was employed to convert the categorical attributes of the datasets into binary form. Specifically, the protocol type, service, and ag features from the NSL-KDD dataset, as well as the proto and service features from the RT-IoT2022 dataset, were transformed into binary representations. Consequently, this encoding process expanded the original feature sets from 41 to 122 dimensions for the NSL-KDD dataset and from 83 to 95 features for the RT-IoT2022 dataset, respectively.

4.3.4. Feature Selection

Feature selection (FS) is a critical preprocessing step in the development of Intrusion Detection Systems (IDSs), aimed at identifying and retaining the most relevant and informative features from the often high-dimensional network traffic data while discarding redundant, irrelevant, or noisy attributes [113]. In the context of IDSs, where datasets commonly contain dozens to hundreds of features (e.g., basic, content-based, time-based, and host-based traffic characteristics), the application of feature selection techniques becomes essential.

4.3.4.1. NSL-KDD dataset's Feature Selection

To reduce the dimensionality of the preprocessed dataset and mitigate redundancy, the Pearson correlation coefficient was employed as a filter-based feature selection method. This statistical measure evaluates the strength and direction of the linear relationship between each feature and the target variable (intrusion class label), producing correlation values ranging from -1 to $+1$. The absolute value of the coefficient indicates the degree of linear association: values close to ± 1 signify a strong linear relationship (suggesting that one variable can be effectively predicted from the other using a linear model), whereas values near zero indicate the absence of any meaningful linear dependency between the variables. Features demonstrating either very weak correlation with the target or excessive correlation with other features were subsequently identified for potential removal, thereby enhancing computational efficiency and model generalization[126].

Using equation (2), the covariance and standard deviation are calculated to derive the Pearson correlation coefficient for each feature X_i with respect to the target feature T.

$$\rho_{X_i,T} = \frac{cov\ (X_i\ , T)}{\sigma_{X_i}\sigma_T} \tag{2}$$

Where:

- X_i is the feature i
- T is the target feature
- cov is the covariance
- σ_{X_i} is the standard deviation of X_i
- σ_T is the standard deviation of T

A quartile-based analysis of Pearson correlation coefficients was employed to determine an appropriate threshold for feature selection [127]. Four cutoff values corresponding to the first through fourth quartiles (Q1 = 0.25, Q2 = 0.5, Q3, and Q4, where Q4 represents the full feature set) were systematically evaluated with respect to both classification performance and computational efficiency. Experimental results indicate no significant difference in predictive performance between Q1 and Q2, suggesting the existence of a performance plateau within this range. However, Q1 resulted in a slightly larger selected feature subset and incurred higher computational cost, with a training time of 96 seconds compared to 94 seconds for Q2.

A similar pattern was observed for higher quartiles: Q3 and Q4 exhibited comparable performance to each other, yet both demonstrated inferior classification metrics relative to Q2. For instance, the full feature set (Q4) achieved an accuracy of 98.61%, whereas the Q2-based feature subset reached a substantially higher accuracy of 99.97%. This degradation at higher quartiles can be attributed to the inclusion of weakly correlated or redundant features, which introduce noise and adversely affect model generalization.

Based on these findings, a Pearson correlation threshold of 0.5 (Q2) was selected as the optimal cutoff. This threshold corresponds to a moderate-to-strong linear association with the target class according to conventional statistical interpretation and represents a balanced trade-off between maximizing predictive performance and minimizing computational overhead. By operating at the onset of the observed performance plateau, the chosen threshold achieves effective dimensionality reduction while enhancing classification accuracy and training efficiency. These properties are particularly advantageous for resource-constrained IoT environments and remain well aligned with the scale and characteristics of the NSL-KDD dataset.

Figures 4.3 and Figure 4.4 provide a visual comparison of the quartile-based Pearson correlation thresholds in terms of predictive performance and computational efficiency, respectively. Figure 4.3 illustrates the variation in classification accuracy across Q1–Q4, highlighting a performance plateau between Q1 and Q2 and a noticeable degradation at higher quartiles. Complementarily, Figure 4.4 depicts the corresponding training time, emphasizing the increased computational cost associated with lower thresholds and larger feature subsets. Together, these figures support the selection of Q2 as the most balanced threshold.

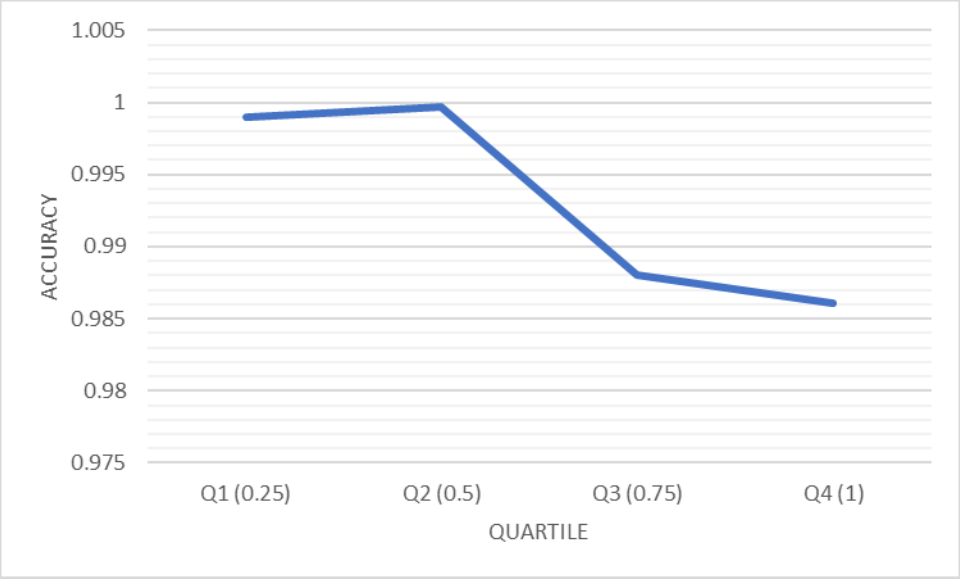


Figure 4.3. Classification Accuracy as a Function of Quartile-Based Pearson Correlation Thresholds

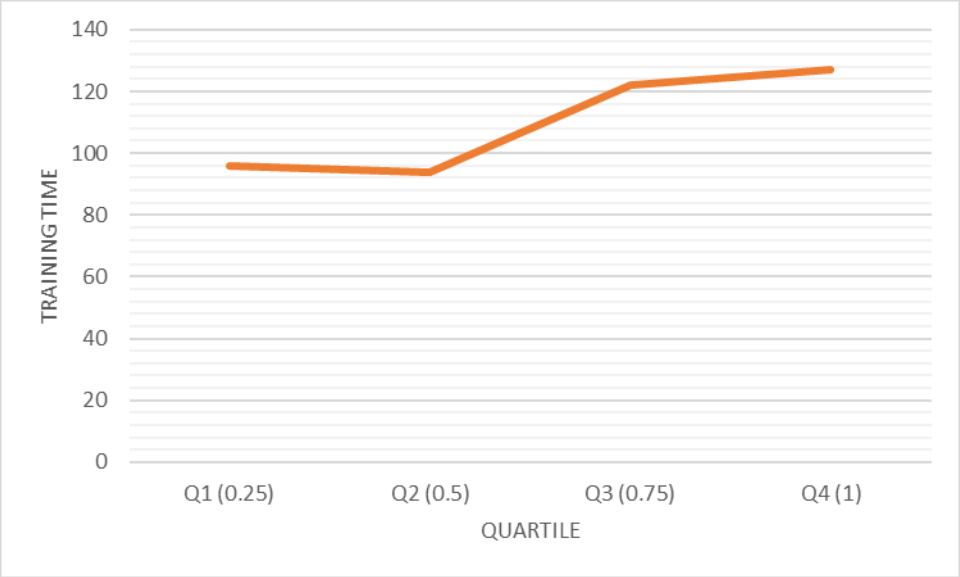


Figure 4.4. Training Time Variation Across Quartile-Based Pearson Correlation Thresholds

Algorithm 1 employs Pearson correlation to methodically remove features that exhibit weak linear relationships with the target variable, enhancing the efficiency of the proposed deep learning-based intrusion detection system for IoT networks.

Algorithm 1: Feature Dropping

Input: Original dataset, Target variable T, Threshold

Output: Reduced dataset

```
for each Ci column in dataset do
    if Abs(correlation(Ci , T)) <  threshold then
        drop column Ci from dataset
    End if
End for
Return reduced dataset
```

Following this process, 29 features were removed due to having a correlation score below 0.5 with the target variable. Figure 4.5 shows both the eliminated and retained features.

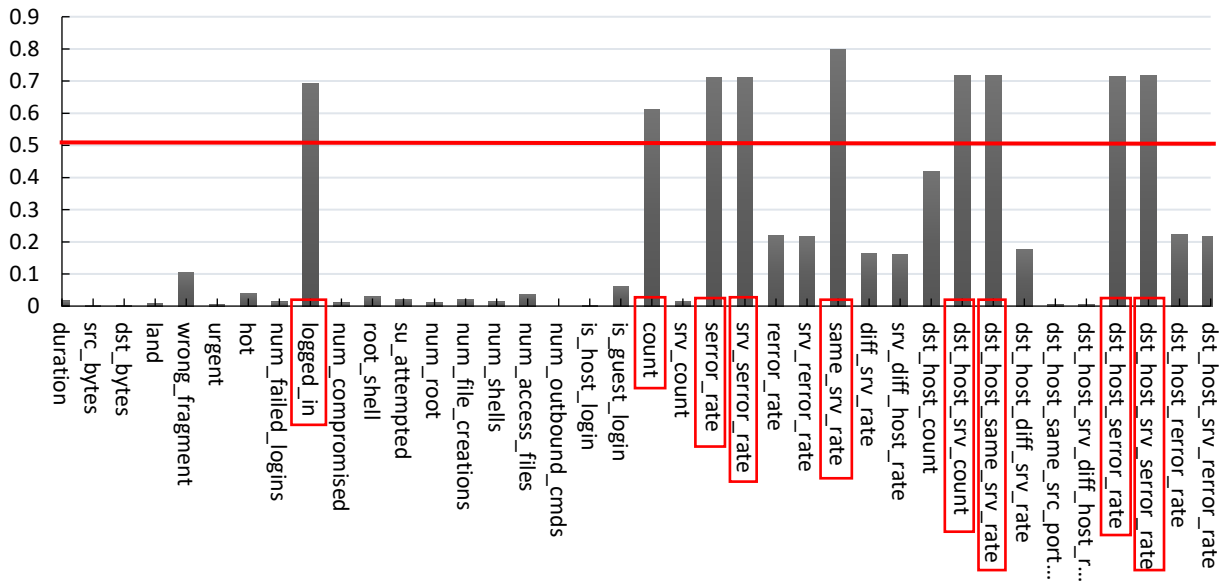


Figure 4.5. Histogram of numeric features included in the NSL-KDD dataset. Features selected are framed in red

4.3.4.2. RT-IoT2022 dataset's Feature Selection

To reduce the size of the preprocessed dataset, a two-step feature selection strategy was applied, combining Pearson correlation and Mutual Information (MI) measures. In this approach, features showing weak linear relationships with the target variable, as indicated by Pearson correlation, were first identified. Then, Mutual Information was used to capture nonlinear dependencies between features and the target, ensuring that only the most relevant features were retained. The thresholds for both methods were carefully adjusted through experimentation to balance the removal of irrelevant features with the preservation of critical information, ultimately improving the efficiency and effectiveness of the dataset for the subsequent deep learning-based intrusion detection model. Pearson correlation is calculated using the formula (2), whereas MI is computed using the formula (3).

$$I(X;Y) = \sum_{x}\sum_{y} P(x,y) log\left(\frac{P(x,y)}{P(x)P(y)}\right) \tag{3}$$

Where

- $P(x,y)$ is the joint probability distribution of (X) and (Y)
- $(P(x))$ is the marginal probability distribution of (X)
- $(P(y))$ is the marginal probability distribution of (Y)

The feature selection process was finalized by merging the results obtained from Pearson correlation and Mutual Information (MI). Features that exceeded the respective thresholds of 0.25 for Pearson correlation and 0.05 for MI were combined using a union operation. As outlined in Algorithm 2, this procedure reduced the original set of 94 features to 76, resulting in the dataset $X_{\text{selected}}$with 123,117 instances. This selection effectively preserves

both linear and nonlinear relationships with the target variable, Attack_type. The threshold values were determined through extensive experimentation, with the chosen combination demonstrating the optimal performance for the subsequent deep learning-based intrusion detection system.

Algorithm 2: Combined Feature Selection

```
Input: X(all features), y(Attack_type), corr_threshold=0.25,
mi_threshold=0.05
Output: X_selected
corr_scores ← PearsonCorrelation(X,y)
mi_scores ← MutualInformation(X,y)
selected_by_corr ←{i||corr_scores[i]| > corr_threshold, i ∈ X}
selected_by_mi ← {i | mi_scores[i] > mi_threshold, i ∈ X}
selected_features ← selected_by_corr U selected_by_mi
X_selected ← X[:, selected_features]
Return X_selected
```

4.4. Architecture I: Pearson Correlation–Based IDS (NSL-KDD)

In this section, we present the first architecture of our proposed intrusion detection system (IDS), which leverages Pearson correlation for feature selection in the NSL-KDD dataset. This architecture provides a basis for evaluating the impact of correlation-based feature selection on the performance of deep learning models in detecting various types of network attacks. The workflow of this architecture is shown in Figure 4.6.

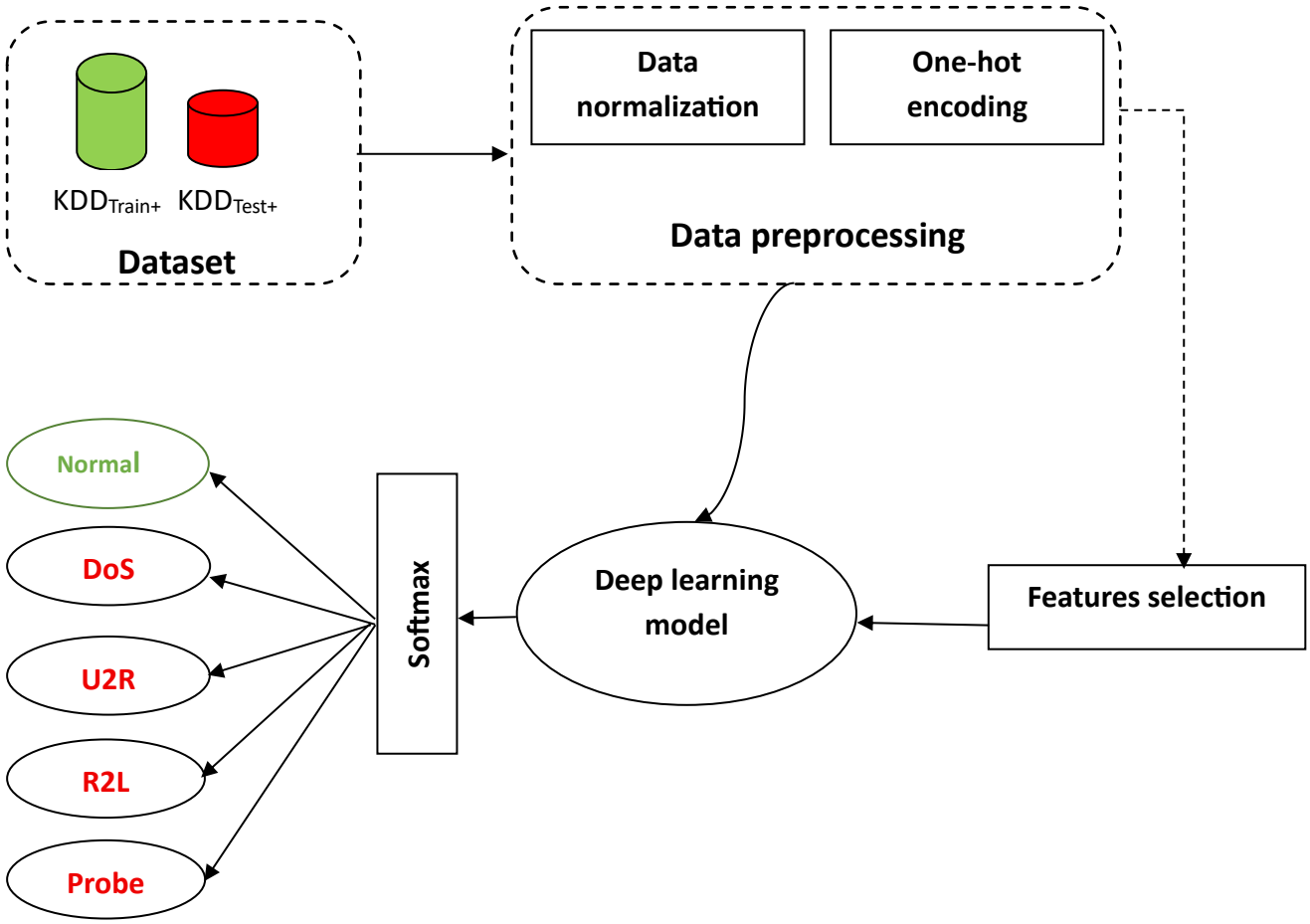


Figure 4.6. Workflow of the proposed Pearson Correlation–Based IDS (NSL-KDD)

4.5. Architecture II: Hybrid Pearson–Mutual Information–Based IDS (RT-IoT2022)

IoT traffic data are characterized by complex, non-linear relationships resulting from device heterogeneity, dynamic communication patterns, and diverse attack behaviors. To effectively capture these relationships, a hybrid feature selection technique that combines Pearson correlation and Mutual Information is adopted. This architecture is specifically designed for realistic IoT environments. Figure 4.7 highlights the workflow of this architecture.

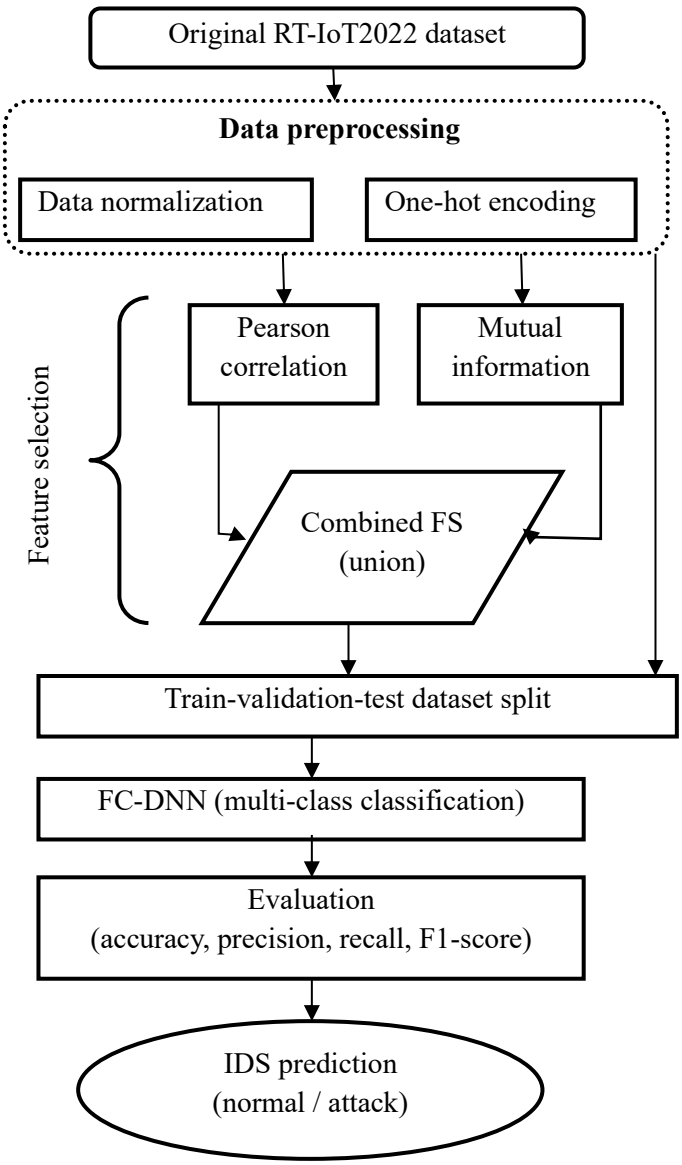


Figure 4.7. Workflow of the Hybrid Pearson–Mutual Information–Based IDS (RT-IoT2022)

4.6. Deep Neural Network Model

A fully connected deep neural network (FC-DNN) is adopted in this work, as it represents the fundamental structure upon which artificial neural networks are built [128]. While certain neural network architectures are better suited to specific application domains—

such as recurrent neural networks for natural language processing and convolutional neural networks for image-related tasks [129]—a fully connected model provides a general and flexible learning framework. In this architecture, input features are fed into the input layer, where each neuron is connected to all neurons in the subsequent hidden layer. This dense connectivity is maintained across successive hidden layers up to the output layer.

4.6.1. Design of the Proposed DNN

A fully connected deep neural network (FC-DNN) is utilized in the proposed intrusion detection system to assign RT-IoT2022 network traffic to one of ten classes, comprising normal traffic and nine distinct attack types. Two experimental scenarios are considered. First, the deep learning model is trained using the complete feature set consisting of 94 attributes. In the second configuration, the model utilizes a reduced feature set of 76 attributes obtained through a hybrid feature selection approach combining Pearson correlation (threshold of 0.25) and Mutual Information (threshold of 0.05).

The network architecture remains identical in both configurations and includes an input layer corresponding to the selected feature dimensionality, followed by three hidden layers containing 10, 50, and 10 neurons, respectively, each employing the ReLU activation function. The output layer comprises ten neurons with a Softmax activation function to produce multi-class probability estimates. Architectural design choices and hyperparameters are determined empirically through iterative experimentation aimed at performance optimization.

Model training is performed using the Adam optimization algorithm with categorical cross-entropy as the loss function. An early stopping mechanism is applied with a patience value of five and a minimum improvement threshold of 0.001. The dataset is divided into training, validation, and testing subsets following a 60–20–20 split to ensure robust performance evaluation. Although the training process allows up to 30 epochs, the model

generally converges within 15 to 22 epochs across multiple experiments, thereby preventing overfitting. A schematic illustration of the proposed DL model is shown in Figure 4.8.

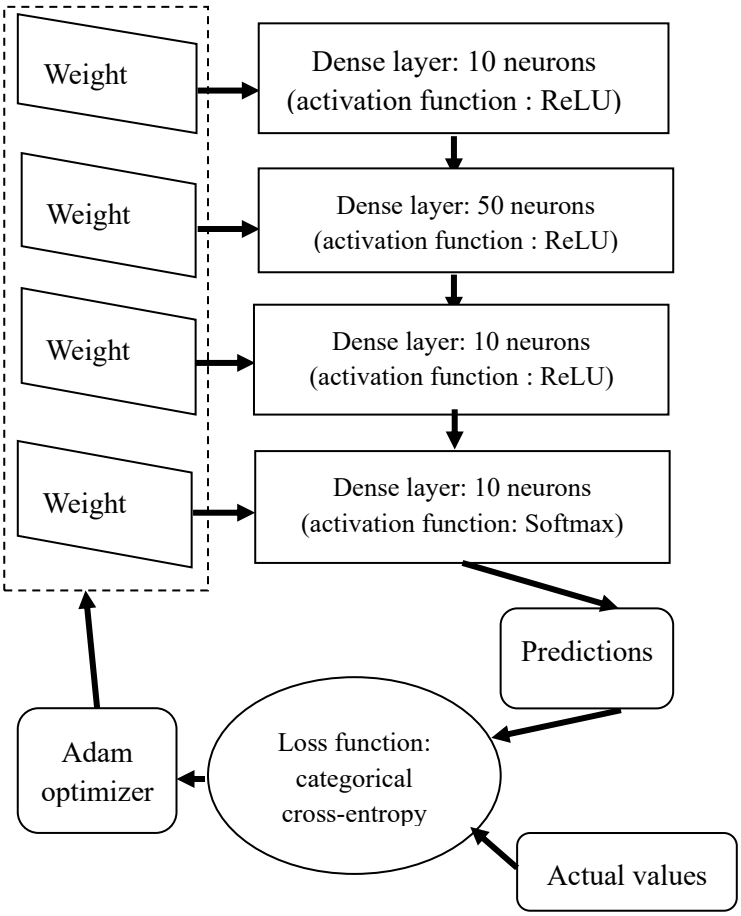


Figure 4.8. Design of the proposed FC-DNN

4.7. Summary

This chapter presented the design and methodology of the proposed deep learning-based intrusion detection system (IDS) for IoT networks. The chapter began by describing the global IDS framework, outlining the overall system architecture, workflow, and integration of feature selection with deep learning. The materials and methodology used for experimentation, including datasets and evaluation metrics, were also detailed.

Two IDS architectures were then introduced. Architecture I, based on Pearson correlation, was applied to the NSL-KDD dataset, while Architecture II combined Pearson correlation with Mutual Information in a hybrid feature selection approach for the RT-IoT2022 dataset. Both architectures were designed to reduce feature dimensionality while maintaining high detection performance.

The chapter also described the deep neural network (DNN) model, including its architecture, training procedure, and parameter settings, highlighting how it learns complex patterns in IoT network traffic. Overall, Chapter 4 established the methodological foundation and system design that underpin the experimental evaluation and results presented in Chapter 5.

Chapter 5

Experimental Setup, Results and Discussion

5.1. Introduction

This chapter presents the experimental setup and evaluation methodology adopted to validate the effectiveness of the proposed intrusion detection system (IDS) architectures. The objective of the experimental analysis is to assess the detection performance, robustness, and generalization capability of the proposed models under both benchmark and realistic IoT traffic conditions. To this end, extensive experiments are conducted using two distinct datasets, namely NSL-KDD and RT-IoT2022, corresponding to the two architectures introduced in Chapter 4.

5.2. Experimental Environment

All experiments are conducted in a controlled software environment to ensure reproducibility and fair comparison. The proposed models are implemented using Python, leveraging widely adopted deep learning libraries such as TensorFlow and Keras. Data preprocessing and feature selection operations are carried out using standard scientific computing libraries.

TensorFlow and Keras were used to implement the deep learning model, due to their flexibility and capability to efficiently process extensive datasets. Model development and training were conducted using Google Colab, which provided the required computational resources, including CPU-based execution.

5.3. Datasets and Experimental Scenarios

5.3.1. NSL-KDD Experimental Scenario

For the NSL-KDD dataset, two experimental scenarios are defined in order to analyze the effect of feature selection on intrusion detection performance. In the first scenario, the proposed IDS is trained and evaluated using the complete set of preprocessed features. In the second scenario, a reduced feature subset obtained through Pearson correlation–based feature selection is employed. This configuration allows for a direct assessment of dimensionality reduction on detection accuracy, false alarm rate, and model efficiency while maintaining comparability with existing IDS studies.

5.3.2. RT-IoT2022 Experimental Scenarios

For the RT-IoT2022 dataset, two experimental scenarios are considered. First, the DNN was trained and evaluated using the complete preprocessed feature set comprising 94 features. In the second configuration, the model was trained on a reduced subset of 76 features, selected through the hybrid filter approach combining Pearson correlation (with a threshold of 0.25) and Mutual Information (with a threshold of 0.05). These scenarios are designed to evaluate IDS performance under realistic IoT traffic conditions and to quantify the benefits of capturing both linear and non-linear feature dependencies.

5.4. Data Splitting Strategy

To ensure reliable and unbiased evaluation, the RT-IoT2022 dataset is partitioned into training, validation, and testing subsets using a 60–20–20 split [106]. The training set is used for model learning, the validation set for hyperparameter tuning and early stopping, and the testing set for final performance evaluation. For the NSL-KDD dataset, the standard KDDTrain+ and KDDTest+ subsets are utilized, enabling fair comparison with existing approaches reported in the literature.

5.5. Evaluation Metrics

In general, to evaluate the detection accuracy of an intrusion detection system (IDS), the following metrics are commonly employed:

— True Positive (TP): This metric represents the number of attack connections that are correctly detected and classified by the model.

— True Negative (TN): This measure indicates the number of normal instances that are correctly predicted and classified as legitimate traffic.

— False Positive (FP): This metric corresponds to the number of normal connections that are incorrectly classified as attack instances by the model.

— False Negative (FN): This measure represents the number of attack connections that are incorrectly predicted and classified as normal traffic by the model.

These measures constitute the fundamental components of the confusion matrix, which is typically used to summarize the performance of a binary classification model, as illustrated in the table below.

		Predicted Classes	
		Normal traffic	Attack
Actual Classes	Normal traffic	TN	FP
	Attack	FN	TP

Table 5.1. Confusion Matrix

In this study, four primary evaluation metrics are employed, namely Accuracy, Precision, Recall, and F1-score. Correct and incorrect classifications are denoted as true (T) and false (F), respectively, while positive (P) and negative (N) correspond to abnormal and

normal traffic as predicted by the IDS. Accordingly, each data instance in the dataset is categorized as either TP, TN, FP, or FN.

Accuracy is computed as the ratio of correctly classified instances to the total number of samples, as formally defined in equation (4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

Precision measures the proportion of instances correctly predicted as positive relative to the total number of instances classified as positive. Its computation is formally defined in equation (5).

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

Recall (also known as sensitivity or true positive rate) represents the proportion of actual positive instances that are correctly identified by the model. It is defined as the ratio of true positives to the total number of positive instances in the dataset and is computed according to equation (6)

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

The F1-score provides a balanced measure of a model's performance by computing the harmonic mean of precision and recall. It is particularly useful when seeking a single metric that effectively trades off between these two complementary evaluation criteria. The F1-score is calculated according to equation (7)

$$F1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{7}$$

5.6. Training Configuration

Model training is conducted using the Adam optimization algorithm with categorical cross-entropy as the loss function. Early stopping is employed to mitigate overfitting, with a patience value of five epochs and a minimum validation loss improvement threshold of 0.001. The maximum number of training epochs is set to 30; however, convergence is typically achieved between 15 and 22 epochs. Hyperparameters, including learning rate and hidden layer sizes, are selected empirically through iterative experimentation to ensure stable convergence and optimal detection performance.

5.7. Experimental Results and Discussion

The following section presents and analyzes the experimental results obtained from evaluating the proposed fully connected deep neural network (FC-DNN) models on the two benchmark datasets used throughout this research: NSL-KDD and RT-IoT2022. The primary objective is to assess the effectiveness of the developed intrusion detection approach, with particular emphasis on the impact of the applied feature selection strategies on classification performance, computational efficiency, and suitability for real-world deployment. Results are reported using standard multi-class classification metrics — accuracy, precision, recall, and F1-score — calculated on independent test sets. Detailed performance comparisons are provided between models trained on the full feature sets and those utilizing the reduced feature subsets obtained through Pearson correlation and Mutual Information-based selection methods, enabling a comprehensive evaluation of the trade-offs between dimensionality reduction and predictive capability.

5.7.1. Pearson Correlation–Based IDS (NSL-KDD)

Table 5.2 summarizes the performance of the fully connected deep neural network (FC-DNN) when trained on the NSL-KDD dataset, comparing the configuration using the

complete set of 41 features (without feature selection) against the reduced feature subset obtained through Pearson correlation-based feature selection.

Metric	Performance of the proposed solution	
	Without FS	*With FS*
Training time	127 s	94 s ($<$ 33 s)
Precision (%)	97.72	99.96 ($>$ 2.24)
Recall (%)	98.61	99.96 ($>$ 1.35)
F1 score (%)	98.16	99.60 ($>$ 1.44)
Accuracy (%)	98.61	99.97 ($>$ 1.36)

Table 5.2. Performance of the Proposed Pearson Correlation-Based IDS

The application of feature selection resulted in a substantial reduction in training time, decreasing from 127 seconds to 94 seconds (a reduction of more than 33 seconds, equivalent to ≈26% faster training). More importantly, the feature selection approach led to consistent and considerable improvements across all classification metrics on the test set. Precision increased from 97.72% to 99.96% (+2.24 percentage points), recall improved from 98.61% to 99.96% (+1.35 points), F1-score rose from 98.16% to 99.60% (+1.44 points), and overall accuracy advanced from 98.61% to 99.97% (+1.36 points).

These results demonstrate that the Pearson correlation-based feature selection not only significantly reduced computational cost but also markedly enhanced the model's discriminative power, leading to near-perfect classification performance on the NSL-KDD benchmark dataset.

To evaluate the performance of the proposed IDS on different types of network traffic, per-class metrics including Precision, Recall, and F1 Score were computed. The results for each class are summarized in Table 5.3.

Metrics	Precision	Recall	F1 score
Dos	0.998	0.998	0.998
Normal	0.995	0.997	0.995
Probe	0.981	0.998	0.989
R2L	0.97	0.975	0.972
U2R	0.94	0.962	0.95

Table 5.3. Per-Class Performance Metrics for the NSL-KDD-Based Solution

The IDS model shows excellent performance across all classes, achieving very high precision, recall, and F1 scores for the majority of attack types.

- DoS and Normal classes: These classes demonstrate near-perfect results (F1 scores ≥ 0.995), indicating that the model is highly effective in distinguishing between normal traffic and Denial-of-Service attacks. This is likely due to the relatively large number of instances available for these classes in the NSL-KDD dataset.
- Probe class: The Probe attacks also achieve strong results (F1 score = 0.989), reflecting that the model can reliably detect reconnaissance activities.
- R2L and U2R classes: These two classes show comparatively lower performance (F1 scores of 0.972 and 0.950, respectively). This reduction is primarily due to the class imbalance problem in the NSL-KDD dataset, where R2L and U2R attacks are significantly underrepresented. The limited number of samples makes it more challenging for the model to generalize and accurately classify these attacks.

Overall, the evaluation demonstrates that while the proposed IDS performs exceptionally well for common attack types, additional strategies such as data augmentation, oversampling, or cost-sensitive learning could further improve detection rates for rare attack classes like R2L and U2R.

5.7.2. Hybrid Pearson-Mutual Information-Based IDS (RT-IoT2022)

To assess the impact of the proposed feature selection (FS) mechanism, we evaluated key performance metrics, including Training Time, Precision, Recall, F1 Score, and Accuracy, both with and without feature selection. The results are summarized in Table 5.4.

Metric	Performance of the proposed solution	
	Without FS	With FS
Training time	181 s	153 s *(< 28 s)*
Precision	99.42	99.51 *(> 0.09)*
Recall	99.39	99.50 *(> 0.11)*
F1 score	99.40	99.50 *(> 0.1)*
Accuracy	99.39	99.50 *(> 0.11)*

Table 5.4. Performance of Hybrid Pearson-Mutual Information-Based IDS

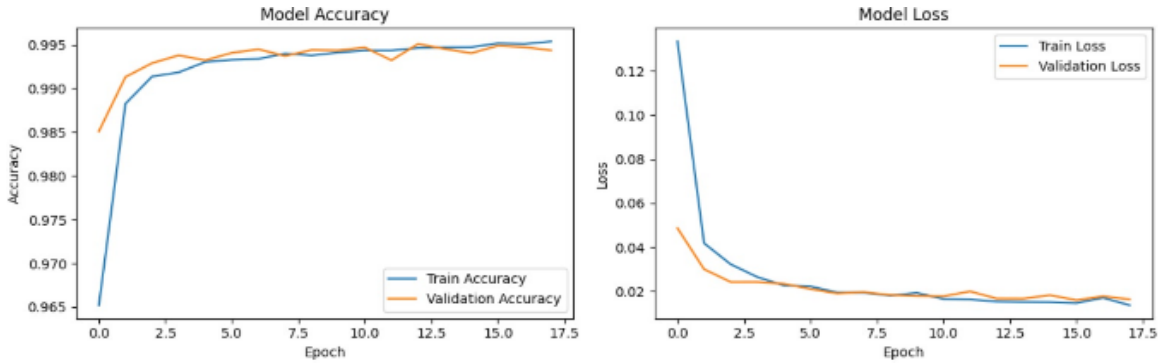
The results show that integrating feature selection into the hybrid Pearson-mutual information-based IDS improves both efficiency and performance:

- Training Time: The use of feature selection reduces training time by 28 seconds, indicating that fewer features lead to faster model training without compromising detection capability.
- Detection Performance: Precision, Recall, F1 Score, and Accuracy all show slight but consistent improvements (0.09–0.11%). These gains, although numerically small, reflect more accurate classification across the dataset.

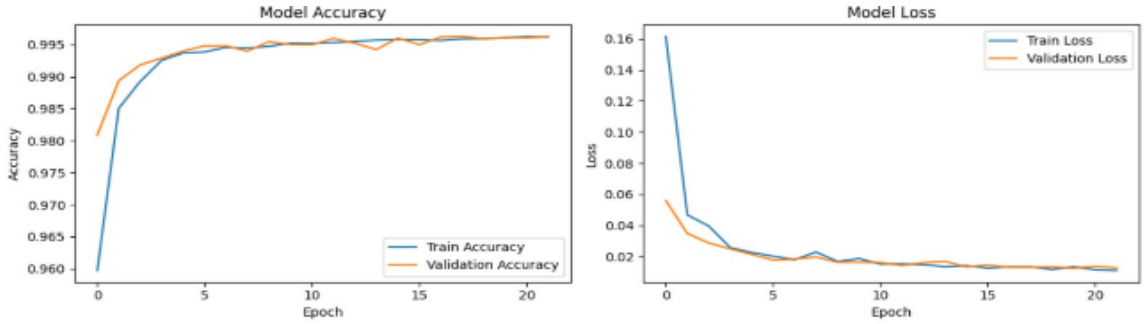
Overall, these results demonstrate that feature selection not only accelerates training but also slightly enhances detection performance, making the IDS more efficient and reliable.

To provide a clearer view of the FC-DNN's training behavior, Figure 5.1 illustrates the training and validation accuracy and loss curves for both experimental setups, with and without feature selection (FS). The curves show the model's convergence over 18 epochs

for 94 features and 22 epochs for 76 features, demonstrating how feature selection influences both performance and stability. The use of early stopping (patience = 5, min_delta = 0.001) ensures optimal generalization, with validation metrics closely following training trends, highlighting the reliability and robustness of the proposed IDS.



(a)



(b)

Figure 5.1. Evolution of training and validation accuracy in the FC-DNN architecture: (a) incorporating feature selection and (b) using the full feature set

Table 5.5 reports the confusion matrix illustrating the classification performance of the proposed FC-DNN model for the various attack types of the RT-IoT2022 dataset.

Actual label											
	ARP_poisioning	549	0	0	0	0	0	1	0	0	30
	DDOS_Slowloris	3	96	0	0	0	0	0	0	0	1
	DOS_SYN_Hping	0	0	889	0	0	0	0	0	0	0
	Metasploit_Brute_Force_SSH	1	0	0	5	0	0	0	0	0	0
	NMAP_FIN_SCAN	0	0	0	0	3	0	0	0	0	0
	NMAPS_OS_DETECTION	0	0	0	0	0	393	0	0	0	0
	NMAP_TCP_SCAN	0	0	0	0	0	0	220	0	0	0
	NMAP_UDP_SCAN	7	25	0	0	0	0	0	454	0	3
	NMAP_XMAP_TREE_SCAN	3	0	0	0	0	0	0	0	381	0
	Normal	58	0	0	0	0	1	0	0	0	2495
		ARP_poisioning	DDOS_Slowloris	DOS_SYN_Hping	Metasploit_Brute_Force_SSH	NMAP_FIN_SCAN	NMAPS_OS_DETECTION	NMAP_TCP_SCAN	NMAP_UDP_SCAN	NMAP_XMAP_TREE_SCAN	Normal
Predicted label											

Table 5.5. Confusion Matrix of the Proposed FC-DNN's performance

To comprehensively assess the effectiveness of the proposed model, we conducted a comparative study against the results reported in [130], which investigated the performance of a variety of machine learning and deep learning algorithms for intrusion detection. The classifiers examined in that study included Deep Autoencoder (AE), Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP), Linear Support Vector Machine (L-SVM), Quadratic Support Vector Machine (Q-SVM), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA).

The comparative evaluation clearly highlights the superior performance of our proposed solution (PS) relative to these models. Specifically, our model achieved precision scores of 99.42% without feature selection (FS) and 99.51% with FS, demonstrating a consistent improvement when FS is applied. In comparison, the other methods yielded significantly

lower precision rates: AE (87.85%), LSTM (82.34%), MLP (85.05%), L-SVM (86.77%), Q-SVM (87.22%), LDA (80.82%), and QDA (78.26%).

These results underscore the proficiency of the proposed DL-driven intrusion detection system in achieving accurate threat detection, particularly when enhanced with feature selection. For a clearer understanding of these differences, Figures 5.2 and Figure 5.3 provide visual comparisons of the precision and accuracy scores achieved by our model alongside those obtained by both deep and traditional (shallow) classifiers. The figures highlight the substantial gains in predictive performance, demonstrating that the proposed solution not only outperforms conventional machine learning techniques but also offers improvements over other deep learning architectures.

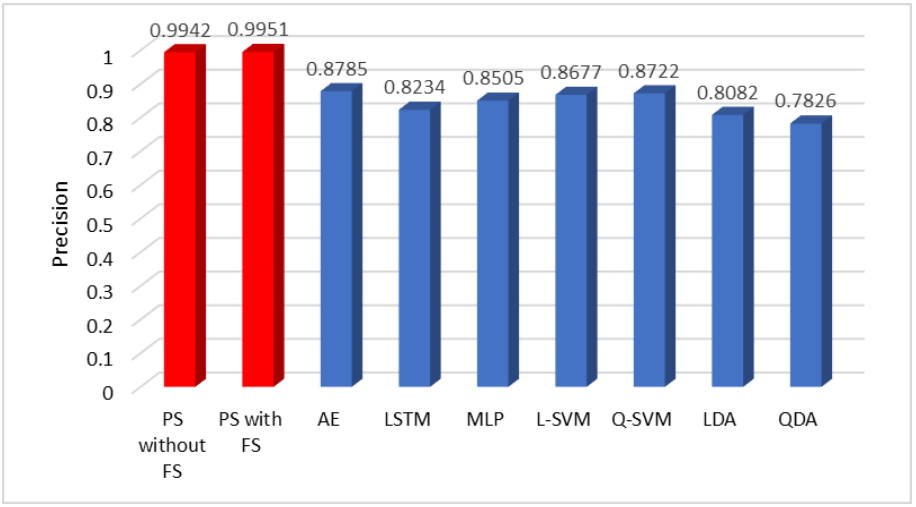


Figure 5.2. Evaluation of precision scores for the proposed solution (PS) and competing deep and shallow models

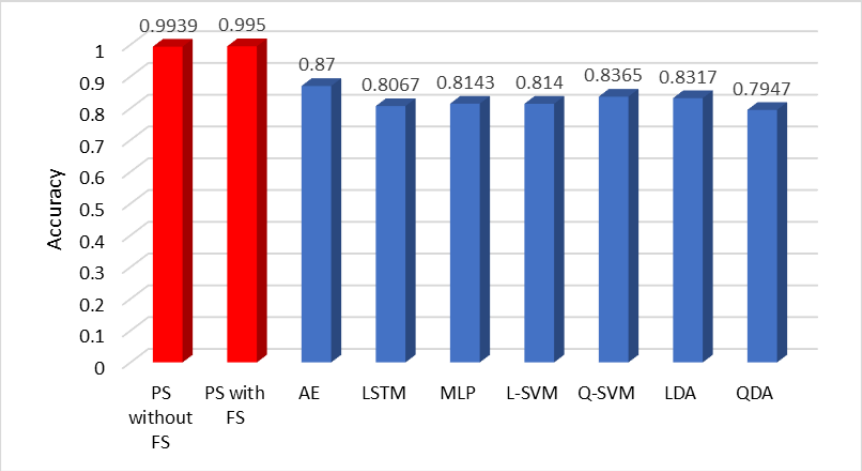


Figure 5.3. Evaluation of accuracy scores for the proposed solution (PS) and competing deep and shallow models

5.8. Limitation and Future Work

Although the proposed deep learning-based intrusion detection system (IDS), enhanced with dual statistical feature selection, demonstrates high accuracy, efficiency, and robustness on the RT-IoT2022 dataset, several limitations must be acknowledged.

Limitations:

1. Static Feature Selection Thresholds: The current approach relies on empirically determined static thresholds for feature selection. While these thresholds proved effective on the chosen dataset, they may not generalize well to other IoT datasets with different distributions, feature correlations, or attack characteristics. This could limit the IDS's adaptability in heterogeneous IoT environments, where traffic patterns and attack vectors can vary significantly.
2. Fixed Neural Network Architecture: The FC-DNN used in this study employs a fixed structure (10-50-10-10 neurons). Although this architecture provides strong classification performance, it may not fully capture the temporal or spatial dependencies present in IoT network traffic. As a result, certain sequential or

context-dependent attacks could be less effectively detected, particularly those requiring analysis of packet sequences or patterns over time.

3. Limited Evaluation Scope: The experimental evaluation was conducted solely on a single benchmark dataset. While this allows for controlled comparison with existing methods, it restricts the ability to assess the model's performance in real-world IoT scenarios, which often involve dynamic traffic, device heterogeneity, and evolving attack strategies. Consequently, the generalizability of the proposed IDS to diverse or unforeseen network conditions remains uncertain.

Future Work:

To address these limitations and further strengthen the proposed IDS, several avenues for future research are suggested:

1. Adaptive Feature Selection Mechanisms: Implementing dynamic or data-driven thresholding strategies for feature selection could improve the system's flexibility across different datasets. Techniques such as percentile-based cutoffs, adaptive mutual information thresholds, or automated threshold tuning could help maintain high detection accuracy in varied IoT environments.
2. Exploration of Advanced Neural Architectures: Investigating alternative deep learning architectures could enhance the model's ability to capture complex patterns. For instance, Convolutional Neural Networks (CNNs) could extract spatial correlations among features, while Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks could model temporal dependencies in sequential traffic data. Hybrid architectures combining CNNs and RNNs may offer even greater potential for detecting sophisticated and context-dependent attacks.

3. Extensive Multi-Dataset Evaluation: Expanding the experimental evaluation to include multiple IoT benchmark datasets and real-time network traffic simulations would provide a more comprehensive assessment of the IDS's robustness. Testing the model against evolving and previously unseen attack types would validate its generalizability and practical applicability in real-world IoT deployments.
4. Integration with Real-Time Systems: Future research could also explore deployment in live IoT environments, focusing on computational efficiency, scalability, and low-latency detection to ensure the IDS can operate effectively under real-time constraints.

By addressing these limitations, future work can enhance the adaptability, robustness, and practical utility of deep learning-based IDS solutions, moving closer to resilient security mechanisms for the increasingly complex IoT landscape.

5.9. Summary

This chapter presented a comprehensive evaluation of the proposed intrusion detection system (IDS) architectures through extensive experiments on the NSL-KDD dataset. The per-class analysis of the first architecture demonstrated excellent performance for well-represented classes, such as DoS, Normal, and Probe, while highlighting the challenges of detecting rare attacks like R2L and U2R due to class imbalance.

For the second architecture, the impact of feature selection was clearly demonstrated. Incorporating feature selection reduced training time and provided slight but consistent improvements in all key performance metrics, including precision, recall, F1 score, and accuracy. Training and validation curves confirmed stable convergence and robust generalization, supported by the early stopping mechanism.

Furthermore, a comparative study with existing machine learning and deep learning classifiers illustrated the superiority of the proposed solution, with precision and accuracy significantly higher than those of traditional and deep learning-based models.

Overall, the experimental results validate the effectiveness, efficiency, and robustness of the proposed IDS, highlighting that combining deep learning with feature selection enhances detection performance and provides a reliable approach for intrusion detection in imbalanced network datasets.

General Conclusion

The exponential proliferation of the Internet of Things (IoT) has transformed modern life, allowing heterogeneous devices to connect seamlessly across different application domains. While IoT systems offer significant benefits in healthcare, smart homes, industrial automation, and transportation, they also introduce unique security challenges due to their heterogeneous, resource-constrained, and widely distributed nature. Intrusion detection systems (IDSs) have emerged as a crucial defense mechanism for safeguarding IoT networks against increasingly sophisticated cyberattacks. However, the high-dimensionality of IoT network data, coupled with imbalanced datasets and evolving attack patterns, continues to pose significant challenges for traditional IDS approaches.

This thesis addressed these challenges by developing a deep learning-based IDS enhanced with a hybrid feature selection framework, designed to improve detection accuracy while maintaining computational efficiency in IoT environments. The study combined rigorous theoretical foundations, methodological innovation, and extensive experimentation to propose a robust and adaptable solution for IoT security.

Summary of Contributions

The primary contribution of this research is the design and implementation of a novel two-stage feature selection framework integrated with deep neural networks (DNNs) to enhance intrusion detection performance. Specifically, the study:

1. Developed a dual-stage feature selection methodology combining Pearson correlation and Mutual Information, enabling the identification of the most relevant features for intrusion detection while reducing dimensionality and computational cost. This approach preserves interpretability and improves the efficiency of the subsequent learning process.

2. Designed optimized deep learning architectures tailored for IoT network traffic, including a fully connected deep neural network capable of learning complex patterns across high-dimensional data. Two experimental architectures were developed:
    - Architecture I: Pearson correlation-based IDS applied to the NSL-KDD dataset.
    - Architecture II: Hybrid Pearson–Mutual Information-based IDS applied to the RT-IoT2022 dataset.
3. Integrated feature selection with deep learning models, demonstrating that the combination of statistical selection techniques and DNN architectures enhances both detection performance and computational efficiency.
4. Conducted comprehensive experimental evaluations, including per-class analysis, performance metrics (accuracy, precision, recall, F1 score), and comparisons with existing machine learning and deep learning classifiers. The results confirmed the superiority of the proposed IDS over conventional and state-of-the-art approaches.
5. Provided critical insights into model behavior and practical applicability, including the impact of dataset imbalance on rare attack classes (R2L and U2R), the benefits of feature selection in reducing training time, and the ability of DNN models to generalize to complex attack patterns.

Key Findings

The experimental results provide several important insights:

- High detection performance: The proposed IDS achieved excellent performance across all evaluated datasets, with overall accuracy and F1 scores exceeding 99% in the majority of experiments. Notably, the integration of feature selection

consistently improved model performance while reducing computational requirements.

- Impact of feature selection: Dual feature selection effectively reduced the number of input features without compromising detection accuracy. This led to faster training times and lower computational overhead, demonstrating that dimensionality reduction can significantly improve the efficiency of deep learning-based IDSs in resource-constrained IoT environments.
- Handling class imbalance: While the model performed exceptionally well for well-represented attack classes, performance for rare attack types (R2L and U2R) was slightly lower due to dataset imbalance. This highlights the need for specialized strategies, such as oversampling, cost-sensitive learning, or adaptive thresholding, to improve detection of underrepresented attack types.
- Comparative advantage: When compared to existing machine learning and deep learning approaches, the proposed IDS outperformed traditional classifiers such as SVM, LDA, QDA, and shallow neural networks, as well as deep learning models like LSTM and autoencoders. This demonstrates the effectiveness of combining statistical feature selection with deep learning for IoT intrusion detection.

Limitations

Despite the promising results, certain limitations must be acknowledged:

1. Static thresholds for feature selection: The current dual statistical framework relies on empirically determined thresholds, which may not generalize optimally across all IoT datasets with different characteristics or emerging attack patterns.
2. Fixed neural network architecture: The DNN architecture employed a predetermined structure (10–50–10–10 neurons), which may not fully capture

temporal or spatial dependencies in IoT network traffic, particularly for sequential or context-dependent attacks.

3. Evaluation on limited datasets: The experimental evaluations were performed primarily on NSL-KDD and RT-IoT2022 datasets, limiting the ability to assess real-world generalizability. Dynamic traffic patterns, device heterogeneity, and novel attack types in operational IoT networks may pose additional challenges.

Future Directions

Several avenues for future research emerge from this study:

1. Adaptive feature selection: Implementing dynamic, data-driven thresholding mechanisms could enhance flexibility across datasets and improve detection performance in heterogeneous IoT environments.
2. Advanced neural architectures: Exploring convolutional neural networks (CNNs) for spatial feature extraction, recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) for temporal dependencies, and hybrid architectures could improve detection of complex attack patterns.
3. Multi-dataset and real-time evaluation: Expanding experimental evaluations to include multiple IoT benchmark datasets and real-time network traffic simulations would validate the robustness and scalability of the proposed IDS against evolving threats.
4. Real-world deployment: Integrating the proposed IDS into live IoT environments to assess performance under real-time constraints, resource limitations, and large-scale deployment scenarios.

In summary, this thesis presents a robust, efficient, and high-performing deep learning-based IDS for IoT networks, demonstrating that the combination of dual feature selection and deep neural networks can effectively address key challenges in IoT security. The

research contributes both methodological innovations and practical insights, providing a foundation for future improvements in intrusion detection systems. The study confirms that intelligent feature selection and optimized deep learning architectures are critical for designing IDSs capable of operating efficiently in the increasingly complex and dynamic landscape of IoT networks.

References

[1] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," Journal of Network and Computer Applications, vol. 88, pp. 10–28, Jun. 2017, doi: 10.1016/j.jnca.2017.04.002.

[2] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," in 2012 10th International Conference on Frontiers of Information Technology, Dec. 2012, pp. 257–260. doi: 10.1109/FIT.2012.53.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, doi: 10.1016/j.future.2013.01.010.

[4] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," IEEE Internet Initiative, vol. 1, no. 1, pp. 1–86, 2015.

[5] S. Krčo, B. Pokrić, and F. Carrez, "Designing IoT architecture(s): A European perspective," in 2014 IEEE World Forum on Internet of Things (WF-IoT), Mar. 2014, pp. 79–84. doi: 10.1109/WF-IoT.2014.6803124.

[6] J. Bradley, J. Loucks, A. Noronha, J. Macaulay, and L. Buckalew, "Internet of everything (IoE)," Survey Report, 2013, Accessed: Dec. 09, 2025. [Online]. Available: https://www.ap-publishing.com/wp-content/uploads/2014/02/IT_IoE_Value_Index_Top_10_Insights.pdf

[7] T. Mohamed, T. Otsuka, and T. Ito, "Towards Machine Learning Based IoT Intrusion Detection Service," in Recent Trends and Future Technology in Applied

Intelligence, M. Mouhoub, S. Sadaoui, O. Ait Mohamed, and M. Ali, Eds., Cham: Springer International Publishing, 2018, pp. 580–585. doi: 10.1007/978-3-319-92058-0_56.

[8] T. Pavithra and S. Jammalamadaka, "Strategies to handle heterogeneity prevalent within an IOT based network," International Journal of Engineering & Technology, vol. 7, p. 77, Mar. 2018, doi: 10.14419/ijet.v7i2.7.10264.

[9] J. Sobral, J. Rodrigues, R. Rabelo, J. Al-Muhtadi, and V. Korotaev, "Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications," Sensors, vol. 19, p. 2144, May 2019, doi: 10.3390/s19092144.

[10] H. An, W. Park, S. Park, and E. Lee, "Logical Space Composition of IoT for a Scalable and Adaptable Smart Environment," in 2024 International Conference on Information Networking (ICOIN), Jan. 2024, pp. 614–618. doi: 10.1109/ICOIN59985.2024.10572086.

[11] D. Willy, Internet of Things (IoT) Paradigms for Mobility and Heterogeneity. 2023. doi: 10.31219/osf.io/wda4g.

[12] "A Survey of Internet of Things and its Applications," International Journal of Advanced Research in Computer and Communication Engineering, vol. 6, no. 3, pp. 456–458, Mar. 2017, doi: 10.17148/IJARCCE.2017.63105.

[13] M. Lal, A. Vyas, B. Basumatary, N. Ramanna, and S. A., "Practical and Innovative Applications of IOT&IOT Networks in Smart Cities," 2024, pp. 174–181. doi: 10.58532/nbennurcech18.

[14] C. A. Irfana Parveen, O. Anjali, and R. Sunder, "Internet of Things: A Review on Its Applications," in Information and Communication Technology for Competitive Strategies (ICTCS 2021), vol. 400, A. Joshi, M. Mahmud, and R. G. Ragel, Eds., in

Lecture Notes in Networks and Systems, vol. 400. , Singapore: Springer Nature Singapore, 2023, pp. 123–134. doi: 10.1007/978-981-19-0095-2_13.

[15] R. Wajgi, J. V. Tembhurne, D. Wajgi, and T. Jain, "Internet of Everything: Applications," in Modern Approaches in IoT and Machine Learning for Cyber Security, V. K. Gunjan, M. D. Ansari, M. Usman, and T. Nguyen, Eds., in Internet of Things. , Cham: Springer International Publishing, 2024, pp. 131–157. doi: 10.1007/978-3-031-09955-7_9.

[16] R. Lohiya and A. Thakkar, "Application Domains, Evaluation Data Sets, and Research Challenges of IoT: A Systematic Review," IEEE Internet of Things Journal, vol. 8, no. 11, pp. 8774–8798, Jun. 2021, doi: 10.1109/JIOT.2020.3048439.

[17] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," Computer Networks, vol. 76, pp. 146–164, Jan. 2015, doi: 10.1016/j.comnet.2014.11.008.

[18] K. Zhao and L. Ge, "A survey on the internet of things security," in 2013 Ninth international conference on computational intelligence and security, IEEE, 2013, pp. 663–667. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6746513/

[19] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," Computer Networks, vol. 57, no. 10, pp. 2266–2279, Jul. 2013, doi: 10.1016/j.comnet.2012.12.018.

[20] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in 2015 IEEE symposium on computers and

communication (ISCC), IEEE, 2015, pp. 180–187. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7405513/

[21] GranjalJorge, MonteiroEdmundo, and S. SilvaJorge, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," IEEE Communications Surveys & Tutorials, Jul. 2015, doi: 10.1109/COMST.2015.2388550.

[22] A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," IEEE Transactions on emerging topics in computing, vol. 5, no. 4, pp. 586–602, 2016.

[23] M. Abomhara and G. M. Køien, "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," Journal of Cyber Security and Mobility, vol. 4, no. 1, pp. 65–88, 2015.

[24] W. Trappe, R. Howard, and R. S. Moore, "Low-energy security: Limits and opportunities in the internet of things," IEEE Security & Privacy, vol. 13, no. 1, pp. 14–21, 2015.

[25] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," Journal of Electrical and Computer Engineering, vol. 2017, pp. 1–25, 2017, doi: 10.1155/2017/9324035.

[26] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," IEEE Internet of things journal, vol. 5, no. 4, pp. 2483–2495, 2017.

[27] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of things security: A top-down survey," Computer Networks, vol. 141, pp. 199–221, 2018.

[28] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," Computer Networks, vol. 31, no. 8, pp. 805–822, Apr. 1999, doi: 10.1016/S1389-1286(98)00017-6.

[29] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," 2000, Accessed: Jan. 09, 2026. [Online]. Available: https://www.cse.msu.edu/~cse960/Papers/security/axelsson00intrusion.pdf

[30] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.

[31] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in 2010 IEEE symposium on security and privacy, IEEE, 2010, pp. 305–316. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5504793/

[32] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. De Alvarenga, "A survey of intrusion detection in Internet of Things," Journal of Network and Computer Applications, vol. 84, pp. 25–37, 2017.

[33] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, "Kalis—A system for knowledge-driven adaptable intrusion detection for the Internet of Things," in 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2017, pp. 656–666. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7980009/

[34] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A specification-based IDS for detecting attacks on RPL-based network topology," Information, vol. 7, no. 2, p. 25, 2016.

[35] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 9042–9053, 2019.

[36] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology," in 2016 IEEE international conference on communications (ICC), IEEE, 2016, pp. 1–6. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7510811/

[37] N. K. Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, and J. Isoaho, "Distributed internal anomaly detection system for Internet-of-Things," in 2016 13th IEEE annual consumer communications & networking conference (CCNC), IEEE, 2016, pp. 319–320. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7444797/

[38] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," Expert Systems with Applications, vol. 36, no. 10, pp. 11994–12000, Dec. 2009, doi: 10.1016/j.eswa.2009.05.029.

[39] C. J. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba, "Trust Management for Host-Based Collaborative Intrusion Detection," in Managing Large-Scale Service Deployment, F. De Turck, W. Kellerer, and G. Kormentzas, Eds., Berlin, Heidelberg: Springer, 2008, pp. 109–122. doi: 10.1007/978-3-540-87353-2_9.

[40] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and Survey of Collaborative Intrusion Detection," ACM Comput. Surv., vol. 47, no. 4, pp. 1–33, Jul. 2015, doi: 10.1145/2716260.

[41] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," Future Generation Computer Systems, vol. 78, pp. 680–698, 2018.

[42] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," Applied Sciences, vol. 8, no. 12, p. 2663, 2018.

[43] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in 2010 IEEE symposium on security and privacy, IEEE, 2010, pp. 305–316. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5504793/

[44] F. Iglesias and T. Zseby, "Analysis of network traffic features for anomaly detection," Mach Learn, vol. 101, no. 1–3, pp. 59–84, Oct. 2015, doi: 10.1007/s10994-014-5473-9.

[45] Y. Xin et al., "Machine learning and deep learning methods for cybersecurity," Ieee access, vol. 6, pp. 35365–35381, 2018.

[46] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE symposium on computational intelligence for security and defense applications, Ieee, 2009, pp. 1–6. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5356528/

[47] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & security, vol. 86, pp. 147–167, 2019.

[48] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep Learning for Anomaly Detection: A Review," ACM Comput. Surv., vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439950.

[49] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," Jan. 23, 2019, arXiv: arXiv:1901.03407. doi: 10.48550/arXiv.1901.03407.

[50] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[51] Y. Bengio, I. Goodfellow, and A. Courville, Deep learning, vol. 1. MIT press Cambridge, MA, USA, 2017. Accessed: Jan. 09, 2026. [Online]. Available: https://www.academia.edu/download/62266271/Deep_Learning20200303-80130-1s42zvt.pdf

[52] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep packet: a novel approach for encrypted traffic classification using deep learning," Soft Comput, vol. 24, no. 3, pp. 1999–2012, Feb. 2020, doi: 10.1007/s00500-019-04030-2.

[53] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "Deep-Full-Range : a deep learning based network encrypted traffic classification and intrusion detection framework," IEEE Access, vol. 7, pp. 45182–45190, 2019.

[54] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," Ieee Access, vol. 5, pp. 21954–21961, 2017.

[55] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in 2016 international conference on platform technology and service (PlatCon), IEEE, 2016, pp. 1–5. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7456805/

[56] R. C. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection," South African Computer Journal, vol. 56, no. 1, pp. 136–154, 2015.

[57] C. Zhou and R. C. Paffenroth, "Anomaly Detection with Robust Deep Autoencoders," in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax NS Canada: ACM, Aug. 2017, pp. 665–674. doi: 10.1145/3097983.3098052.

[58] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," Special lecture on IE, vol. 2, no. 1, pp. 1–18, 2015.

[59] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "CatchSync: catching synchronized behavior in large directed graphs," in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, New York New York USA: ACM, Aug. 2014, pp. 941–950. doi: 10.1145/2623330.2623632.

[60] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," Feb. 15, 2016, arXiv: arXiv:1510.00149. doi: 10.48550/arXiv.1510.00149.

[61] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," Mar. 20, 2015, arXiv: arXiv:1412.6572. doi: 10.48550/arXiv.1412.6572.

[62] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in 2017 ieee symposium on security and privacy (sp), Ieee, 2017, pp. 39–57. Accessed: Jan. 09, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7958570/

[63] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of machine learning research, vol. 3, no. Mar, pp. 1157–1182, 2003.

[64] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 5, pp. 516–524, 2010.

[65] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," IEEE transactions on pattern analysis and machine intelligence, vol. 19, no. 2, pp. 153–158, 2002.

[66] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," bioinformatics, vol. 23, no. 19, pp. 2507–2517, 2007.

[67] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers & electrical engineering, vol. 40, no. 1, pp. 16–28, 2014.

[68] J. Li et al., "Feature Selection: A Data Perspective," ACM Comput. Surv., vol. 50, no. 6, pp. 1–45, Nov. 2018, doi: 10.1145/3136625.

[69] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," Neural Comput & Applic, vol. 24, no. 1, pp. 175–186, Jan. 2014, doi: 10.1007/s00521-013-1368-0.

[70] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in Proceedings of the 20th international conference on machine learning (ICML-03), 2003, pp. 856–863. Accessed: Jan. 09, 2026. [Online]. Available: https://cdn.aaai.org/ICML/2003/ICML03-111.pdf

[71] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," IEEE Transactions on neural networks, vol. 5, no. 4, pp. 537–550, 1994.

[72] K. Kira and L. A. Rendell, "A practical approach to feature selection," in Machine learning proceedings 1992, Elsevier, 1992, pp. 249–256. Accessed: Jan. 09, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781558602472500371

[73] T. N. Lal, O. Chapelle, J. Weston, and A. Elisseeff, "Embedded Methods," in Feature Extraction, vol. 207, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds., in Studies in Fuzziness and Soft Computing, vol. 207. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 137–165. doi: 10.1007/978-3-540-35488-8_6.

[74] R. Tibshirani, "Regression shrinkage and selection via the lasso," Journal of the Royal Statistical Society Series B: Statistical Methodology, vol. 58, no. 1, pp. 267–288, 1996.

[75] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[76] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in 2018 IEEE security and privacy workshops (SPW), IEEE, 2018, pp. 29–35. Accessed: Jan. 10, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8424629/

[77] A. Sivanathan et al., "Characterizing and classifying IoT traffic in smart cities and campuses," in 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2017, pp. 559–564. Accessed: Jan. 10, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8116438/

[78] E. Hodo et al., "Threat analysis of IoT networks using artificial neural network intrusion detection system," in 2016 International symposium on networks, computers and

communications (ISNCC), IEEE, 2016, pp. 1–6. Accessed: Jan. 10, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7746067/

[79] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," Future Generation Computer Systems, vol. 100, pp. 779–796, 2019.

[80] A. R. Abdulla and N. G. M. Jameel, "A review on IoT intrusion detection systems using supervised machine learning: Techniques, datasets, and algorithms," UHD Journal of Science and Technology, vol. 7, no. 1, pp. 53–65, 2023.

[81] K. Albulayhi, A. A. Smadi, F. T. Sheldon, and R. K. Abercrombie, "IoT intrusion detection taxonomy, reference architecture, and analyses," Sensors, vol. 21, no. 19, p. 6432, 2021.

[82] Y. Alotaibi and M. Ilyas, "Ensemble-learning framework for intrusion detection to enhance internet of things' devices security," Sensors, vol. 23, no. 12, p. 5568, 2023.

[83] A. Aldaej, I. Ullah, T. A. Ahanger, and M. Atiquzzaman, "Ensemble technique of intrusion detection for IoT-edge platform," Scientific Reports, vol. 14, no. 1, p. 11703, 2024.

[84] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1686–1721, 2020.

[85] R. Shahbazian, G. Macrina, E. Scalzo, and F. Guerriero, "Machine learning assists IoT localization: a review of current challenges and future trends," Sensors, vol. 23, no. 7, p. 3551, 2023.

[86] M. AlShaikh, W. Alsemaih, S. Alamri, and Q. Ramadan, "Using supervised learning to detect command and control attacks in iot," International Journal of Cloud Applications and Computing (IJCAC), vol. 14, no. 1, pp. 1–19, 2024.

[87] M. Shahnawaz Ahmad and S. Mehraj Shah, "Unsupervised ensemble based deep learning approach for attack detection in IoT network," Concurrency and Computation, vol. 34, no. 27, p. e7338, Dec. 2022, doi: 10.1002/cpe.7338.

[88] D. Dwivedi, A. Bhushan, and A. K. Singh, "Leveraging K-means clustering for enhanced detection of network traffic attacks," in 2024 3rd International conference on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC), IEEE, 2024, pp. 72–76. Accessed: Jan. 10, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10486408/

[89] M. Shenify, F. A. Mazarbhuiya, and A. S. Wungreiphi, "Detecting IoT anomalies using fuzzy subspace clustering algorithms," Applied Sciences, vol. 14, no. 3, p. 1264, 2024.

[90] M. A. Alsoufi et al., "Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review," Applied sciences, vol. 11, no. 18, p. 8383, 2021.

[91] O. G. Darley, A. A. Adenowo, and A. I. Yussuff, "Machine learning intrusion detection as a solution to security and privacy issues in IoT: a systematic review," FUOYE Journal of Engineering and Technology, vol. 7, no. 2, pp. 148–156, 2022.

[92] E. Ashraf, N. Areed, H. Salem, E. Abdelhady, and A. Farouk, "IoT based intrusion detection systems from the perspective of machine and deep learning: a survey and comparative study," Delta University Scientific Journal, vol. 5, no. 2, pp. 367–386, 2022.

[93] H. P, "A Survey on Machine and Deep Learning Based Intrusion Detection Systems for IoT," International Journal for Research in Applied Science and Engineering Technology, vol. 11, pp. 3490–3494, Jun. 2023, doi: 10.22214/ijraset.2023.54325.

[94] J. Zhang, X.-Y. Zhang, C. Wang, and C.-L. Liu, "Deep representation learning for domain generalization with information bottleneck principle," Pattern Recognition, vol. 143, p. 109737, 2023.

[95] J. Yang, H. Qian, H. Zou, and L. Xie, "Learning decomposed hierarchical feature for better transferability of deep models," Information Sciences, vol. 580, pp. 385–397, 2021.

[96] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An introductory review of deep learning for prediction models with big data," Frontiers in artificial intelligence, vol. 3, p. 4, 2020.

[97] K. Zhang et al., "Compacting deep neural networks for Internet of Things: Methods and applications," IEEE Internet of Things Journal, vol. 8, no. 15, pp. 11935–11959, 2021.

[98] M. M. Bejani and M. Ghatee, "A systematic review on overfitting control in shallow and deep neural networks," Artif Intell Rev, vol. 54, no. 8, pp. 6391–6438, Dec. 2021, doi: 10.1007/s10462-021-09975-1.

[99] S. M. Sulaiman and W. M. Abduallah, "A Comprehensive Review of Learning-Based Anomaly Detection Techniques in IoT Security Systems," East Journal of Computer Science, vol. 1, no. 4, pp. 18–27, 2025.

[100] J. Krupski, W. Graniszewski, and M. Iwanowski, "Data transformation schemes for cnn-based network traffic analysis: A survey," Electronics, vol. 10, no. 16, p. 2042, 2021.

[101] M. Krichen and A. Mihoub, "Long short-term memory networks: A comprehensive survey," AI, vol. 6, no. 9, p. 215, 2025.

[102] T. Vaiyapuri, Z. Sbai, H. Alaskar, and N. A. Alaseem, "Deep learning approaches for intrusion detection in IIoT networks–opportunities and future directions," International Journal of Advanced Computer Science and Applications, vol. 12, no. 4, 2021, Accessed: Jan. 10, 2026. [Online]. Available: https://www.researchgate.net/profile/Thavavel-Vaiyapuri/publication/351311900_Deep_Learning_Approaches_for_Intrusion_Detection_in_IIoT_Networks_-_Opportunities_and_Future_Directions/links/60c74605a6fdcc2e6141032f/Deep-Learning-Approaches-for-Intrusion-Detection-in-IIoT-Networks-Opportunities-and-Future-Directions.pdf

[103] F. F. Alruwaili, "Optimal hybrid deep learning enabled attack detection and classification in IoT environment," Computers, Materials & Continua, vol. 75, no. 1, pp. 99–115, 2023.

[104] A. Aldaej, T. A. Ahanger, and I. Ullah, "Deep learning-inspired IoT-IDS mechanism for edge computing environments," Sensors, vol. 23, no. 24, p. 9869, 2023.

[105] Y. Rizk and M. Awad, "On extreme learning machines in sequential and time series prediction: A non-iterative and approximate training algorithm for recurrent neural networks," Neurocomputing, vol. 325, pp. 1–19, 2019.

[106] F. Khatemi and meriem meddeber, "Enhancing IoT Intrusion Detection: Deep Learning with Dual Statistical Feature Selection," International Journal of Intelligent Engineering and Systems, vol. 18, p. 63, Sep. 2025.

[107] S. Duraibi and A. M. Alashjaee, "Enhancing cyberattack detection using dimensionality reduction with hybrid deep learning on internet of things environment," IEEE Access, vol. 12, pp. 84752–84762, 2024.

[108] Q. A. Al-Haija and A. Droos, "A comprehensive survey on deep learning-based intrusion detection systems in Internet of Things (IoT)," Expert Systems, vol. 42, no. 2, p. e13726, Feb. 2025, doi: 10.1111/exsy.13726.

[109] S. Tsimenidis, T. Lagkas, and K. Rantos, "Deep Learning in IoT Intrusion Detection," J Netw Syst Manage, vol. 30, no. 1, p. 8, Jan. 2022, doi: 10.1007/s10922-021-09621-9.

[110] Muaadh. A. Alsoufi, S. Razak, M. M. Siraj, A. Ali, M. Nasser, and S. Abdo, "Anomaly Intrusion Detection Systems in IoT Using Deep Learning Techniques: A Survey," in Innovative Systems for Intelligent Health Informatics, vol. 72, F. Saeed, F. Mohammed, and A. Al-Nahari, Eds., in Lecture Notes on Data Engineering and Communications Technologies, vol. 72. , Cham: Springer International Publishing, 2021, pp. 659–675. doi: 10.1007/978-3-030-70713-2_60.

[111] A. Alfahaid, E. Alalwany, A. M. Almars, F. Alharbi, E. Atlam, and I. Mahgoub, "Machine Learning-Based Security Solutions for IoT Networks: A Comprehensive Survey," Sensors, vol. 25, no. 11, p. 3341, 2025.

[112] H. A. A. Hassan and M. Zolfy, "Exploring Lightweight Deep Learning Techniques for Intrusion Detection Systems in IoT Networks: A Survey," Journal of Electrical Systems, vol. 20, pp. 1944–1958, 2024.

[113] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," J Big Data, vol. 11, no. 1, p. 36, Feb. 2024, doi: 10.1186/s40537-024-00892-y.

[114] J. Li, H. Chen, M. O. Shahizan, and L. M. Yusuf, "Enhancing IoT Security: A comparative study of feature reduction techniques for intrusion detection system," Intelligent Systems with Applications, vol. 23, p. 200407, 2024.

[115] Z. Wang, H. Chen, S. Yang, X. Luo, D. Li, and J. Wang, "A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization," PeerJ Computer Science, vol. 9, p. e1569, 2023.

[116] G. A. Mukhaini, M. Anbar, S. Manickam, T. A. Al-Amiedy, and A. Al Momani, "A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in Internet of Things networks," Journal of King Saud University-Computer and Information Sciences, vol. 36, no. 1, p. 101866, 2024.

[117] P. H. Prastyo, I. Ardiyanto, and R. Hidayat, "A review of feature selection techniques in sentiment analysis using filter, wrapper, or hybrid methods," in 2020 6th International Conference on Science and Technology (ICST), IEEE, 2020, pp. 1–6. Accessed: Jan. 10, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9732885/

[118] J. Pirgazi, M. Alimoradi, T. Esmaeili Abharian, and M. H. Olyaee, "An Efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets," Scientific reports, vol. 9, no. 1, p. 18580, 2019.

[119] Q. A. Al-Haija and A. Droos, “A comprehensive survey on deep learning-based intrusion detection systems in Internet of Things (IoT),” Expert Systems, vol. 42, no. 2, p. e13726, Feb. 2025, doi: 10.1111/exsy.13726.

[120] N. Mishra and S. Pandya, “Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review,” IEEE Access, vol. 9, pp. 59353–59377, 2021.

[121] H. Hindy et al., “A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems,” IEEE Access, vol. 8, pp. 104650–104675, 2020, doi: 10.1109/ACCESS.2020.3000179.

[122] B. S. Sharmila and R. Nagapadma, “Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset,” Cybersecurity, vol. 6, no. 1, p. 41, Sep. 2023, doi: 10.1186/s42400-023-00178-5.

[123] Md. A. Talukder et al., “Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction,” J Big Data, vol. 11, no. 1, p. 33, Feb. 2024, doi: 10.1186/s40537-024-00886-w.

[124] M. Suárez-Álvarez, D. Pham, M. Y. Prostov, and Y. Prostov, “Statistical approach to normalization of feature vectors and clustering of mixed datasets,” Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 468, pp. 2630–2651, 2012, doi: 10.1098/rspa.2011.0704.

[125] P. Rodríguez, M. A. Bautista, J. Gonzàlez, and S. Escalera, “Beyond one-hot encoding: Lower dimensional target embedding,” Image and Vision Computing, vol. 75, pp. 21–31, Jul. 2018, doi: 10.1016/j.imavis.2018.04.004.

[126] P. Schober, C. Boer, and L. Schwarte, "Correlation Coefficients: Appropriate Use and Interpretation," Anesthesia & Analgesia, vol. 126, 2018, doi: 10.1213/ANE.0000000000002864.

[127] A. Al Marouf, M. K. Hasan, and H. Mahmud, "Comparative analysis of feature selection algorithms for computational personality prediction from social media," IEEE Transactions on Computational Social Systems, vol. 7, no. 3, pp. 587–599, 2020.

[128] S. Nayyar, S. Arora, and M. Singh, "Recurrent Neural Network Based Intrusion Detection System," in 2020 International Conference on Communication and Signal Processing (ICCSP), Jul. 2020, pp. 0136–0140. doi: 10.1109/ICCSP48568.2020.9182099.

[129] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," Computer Networks, vol. 186, p. 107784, Feb. 2021, doi: 10.1016/j.comnet.2020.107784.

[130] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," Neurocomput., vol. 387, no. C, pp. 51–62, Apr. 2020, doi: 10.1016/j.neucom.2019.11.016.