LATİFE İLAYDA IŞIN

ATILIM UNIVERSITY 2025

# UTILIZATION OF FEDERATED LEARNING FOR IOT DEVICE SECURITY

THE GRADUATE SCHOOL OF NATURAL
AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

LATİFE İLAYDA IŞIN

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2025

# UTILIZATION OF FEDERATED LEARNING FOR IOT DEVICE SECURITY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

LATİFE İLAYDA IŞIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2025

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

_____

Prof. Dr. Ender Keskinkılıç
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Electrical and Electronics Engineering, Atılım University.**

_____

Prof. Dr. Reşat Özgür Doruk
Head of Department

This is to certify that we have read the thesis UTILIZATION OF FEDERATED LEARNING FOR IOT DEVICE SECURITY submitted by LATİFE İLAYDA IŞIN and that our opinion it if fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Yaser Dalveren
Supervisor

**Examining Committee Members:**

Prof. Dr. Ali Kara
Electrical and Electronics Engineering
Gazi University _____

Assoc. Prof. Dr. Yaser Dalveren
Electrical- Electronics Engineering
İzmir Bakırçay University _____

Asst. Prof. Dr. Bengisu Yalçınkaya
Electrical and Electronics Engineering
Atılım University _____

**Date:** 22 January 2025

I here by declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Latife İlayda Işın

Signature:

# ABSTRACT

## UTILIZATION OF FEDERATED LEARNING FOR IOT DEVICE SECURITY

Işın, Latife İlayda
MS, Department Of Electrical and Electronics Engineering
Supervisor: Assoc. Prof Dr.Yaser Dalveren

January 2025, 54 pages

The proliferation of IoT devices in critical systems has increased the need for robust cybersecurity mechanisms. This study investigates the implementation of a Federated Learning (FL) framework for network intrusion detection, addressing privacy concerns by decentralizing model training across multiple nodes. Utilizing datasets such as Bot-IoT, TON_IoT, UNSW-NB15, and CICIDS2017, we assess the effectiveness of FL in detecting various attack types, including DoS, DDoS, Man-in-the-Middle, and brute force attacks. Extensive experiments demonstrate high model performance, achieving 98.2% accuracy and 97.8% precision on the Bot-IoT dataset, while maintaining competitive results across other datasets. Comparative evaluations highlight the efficacy of the FL approach against traditional centralized learning models. Loss and accuracy curves reflect the stability of the training process, and confusion matrix analyses provide insights into misclassification patterns. This work underscores the potential of FL as a scalable and privacy-preserving solution for securing IoT environments against evolving threats.

Keywords: Iot, Intrusion Detection, Federated Learning, Cyberattacks.

# ÖZ

## IOT CİHAZ GÜVENLİĞİ İÇİN FEDERE ÖĞRENMENİN KULLANIMI

Işın, Latife İlayda
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Yaser Dalveren

Ocak 2025, 54 sayfa

IoT cihazlarının kritik sistemlerde yaygınlaşması, sağlam siber güvenlik mekanizmalarına olan ihtiyacı artırdı. Bu çalışma, ağ saldırı tespiti için bir Federe Öğrenme (FL) çerçevesinin uygulanmasını araştırmakta ve model eğitimini birden çok düğümde ademi merkeziyetçi hale getirerek gizlilik endişelerini ele almaktadır. Bot-IoT, TON_IoT, UNSW-NB15 ve CICIDS2017 gibi veri kümelerini kullanarak, fl'nin DoS, DDoS, Ortadaki Adam ve kaba kuvvet saldırıları dahil olmak üzere çeşitli saldırı türlerini tespit etmedeki etkinliğini değerlendiriyoruz. Kapsamlı deneyler, diğer veri kümelerinde rekabetçi sonuçları korurken, Bot-IoT veri kümesinde% 98,2 doğruluk ve% 97,8 hassasiyet elde ederek yüksek model performansı göstermektedir. Karşılaştırmalı değerlendirmeler, FL yaklaşımının geleneksel merkezi öğrenme modellerine karşı etkinliğini vurgulamaktadır. Kayıp ve doğruluk eğrileri, eğitim sürecinin istikrarını yansıtır ve karışıklık matrisi analizleri, yanlış sınıflandırma kalıpları hakkında fikir verir. Bu çalışma, IoT ortamlarını gelişen tehditlere karşı güvence altına almak için ölçeklenebilir ve gizliliği koruyan bir çözüm olarak fl'nin potansiyelinin altını çiziyor.

Anahtar Kelimeler: Nesnelerin İnterneti, İzinsiz Giriş, Federe Öğrenme, Siber atak.

## ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my supervisor Assoc. Prof. Dr. Yaser Dalveren for his constant support, valuable advice, and the countless hours they invested in helping me refine this work. His guidance was crucial to my progress.

I am incredibly grateful to Prof. Dr. Ali Kara for his guidance and support throughout this research. His insightful comments and suggestions were vital to the completion of this work.

Finally, I would like to thank my family, and my friends for their unwavering support, love, and understanding. Without their constant encouragement, this accomplishment would not have been possible.

## TABLE OF CONTENTS

## LIST OF TABLES

TABLES

## LIST OF FIGURES

FIGURES

## LIST OF SYMBOLS/ABBREVIATIONS

### Symbols

| | |
|---|---|
| $D(x)$ | Features of observed activity |
| $f(x)$ | Observed behavior |
| $P(y\|x)$ | Event of activity |
| $\beta_n$ | Model parameters |
| $p(x)$ | Probability density function |
| $L_i(w)$ | Local loss function |
| $w(t)$ | Updated global model |
| $N$ | Total number of devices |
| $x'$ | Normalized values |
| $\min(x)$ | Minimum values of dataset |
| $\max(x)$ | Maximum values of dataset |
| $T$ | Window size |
| $R_s$ | Fixed rate |
| $X(f)$ | Frequency domain representation |
| $y_j$ | Updated global model |
| $\hat{y}_j$ | True label |
| $x'$ | Prediction |

### Acronyms

| | |
|---|---|
| IoT | Internet of Things |
| IIoT | Industrial Internet of Things |
| IDS | Intrusion Detection System |
| DDoS | Distributed Denial of Service |
| DoS | Denial of Service |
| ML | Machine Learning |

| | |
|---|---|
| FL | Federated Learning |
| GDPR | General Data Protection Regulation |
| LocalSGD | Local Stochastic Gradient Descent |
| FedSVRG | Federated Stochastic Variance Reduced Gradient |
| LSTM | Long Short Term Memory |
| RF | Random Forest |
| SVM | Support Vector Machines |
| FFT | Fast Fourier Transform |
| STFT | Short Time Fourier Transform |
| AE | AutoEncoder |
| FedAvg | Federated Average |
| DNN | Deep Neural Network |
| DP | Differential Privacy |
| XAI | Explainable Artificial Intelligence |

# CHAPTER 1

## INTRODUCTION

The Internet of Things (IoT) is changing the way devices communicate and work together, marking a major shift in how technology intgerates with daily life. Essentially, IoT refers to connecting everyday objects to the internet, enabling smooth data sharing that drives automation, boosts efficiency, and sparks and innovation. The rapid spread of IoT has been driven by advancements in sensor technology, wireless communication, and data analytics, as well as the decreasing cost of hardware components . These factors have fueled its growth across various industries, reshaping traditional practices in the process.There are several reasons for IoT's widespread adoption. For one, its scalability and flexibility make it suitable for a range of settings, from individual homes to large industrial facilities. In addition, real-time data collection and analysis offer valuable insights that help organizations improve their operations and make better decisions [1].

IoT is being used in many different fields. In healthcare, for example, IoT-enabled devices support remote patient monitoring, manage chronic conditions, and provide predictive analytics, leading to better outcomes while cutting costs. In agriculture, IoT is transforming farming with tools like soil sensors and weather monitoring systems, which use data to optimize farming practices, conserve resources, and improve crop quality. Similarly, the energy industry uses IoT to monitor and manage power grids, promoting efficiency and sustainability. Smart cities use IoT to improve infrastructure, managing everything from traffic and waste to public safety systems. Retailers are also benefiting by using IoT solutions to track inventory and create personalized shopping experiences, which streamlines supply chains and enhances customer satisfaction [2].

The rise of IoT devices has brought unprecedented connectivity, but it has also introduced significant challenges, especially in managing and securing data. With their diverse designs and the massive volume of data they generate, IoT systems face unique vulnerabilities [3]. These devices often function in various environments, handling vast amounts of unstructured, sensitive data in real time. This dynamic setup, combined with the limited processing capabilities of many IoT devices, makes them susceptible to numerous cyber threats [4].

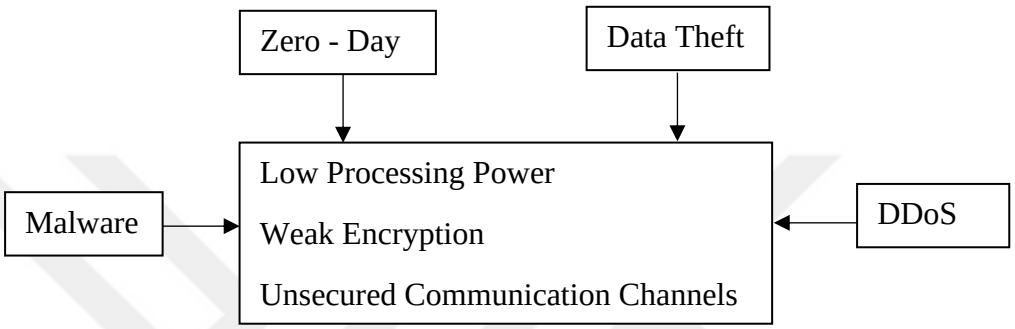


Figure 1. 1. Security Vulnerabilities of IoT Devices and Common Threats

One key solution to IoT security issues is the use of Intrusion Detection Systems (IDS). IDS are critical for monitoring network activity, spotting unusual behaviors, and preventing security breaches. However, traditional IDS approaches are often inadequate for IoT settings because cyberattacks are becoming increasingly sophisticated, and IoT ecosystems are highly complex. Conventional systems are typically built for fixed network structures and threats, which makes them less effective for IoT networks' decentralized and changing nature. Additionally, the limited computing power and memory of IoT devices add to the challenge of implementing robust security measures [5].

With cyber threats such as zero-day attacks, distributed denial-of-service (DDoS) attacks, and ransomware growing more complex, new IDS strategies tailored to IoT environments are essential. Innovative methods, like machine learning-based IDS and deep learning systems, show promise in detecting and countering advanced attack patterns [6]. These solutions can handle large-scale, real-time data, improving

anomaly detection and allowing for more proactive threat management. Furthermore, lightweight yet effective detection techniques can overcome the resource constraints of IoT devices, making IDS a practical and efficient security solution.

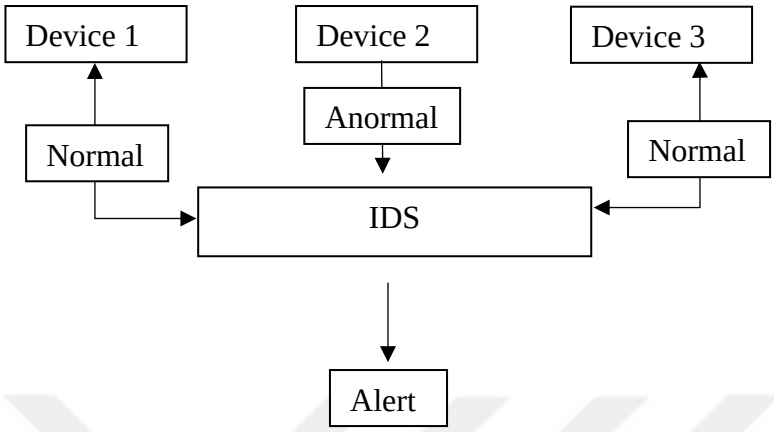


Figure 1. 2. Relationship between IDS and IoT Devices

The challenges faced by traditional Intrusion Detection Systems (IDS) in managing the dynamic and diverse nature of IoT environments have driven researchers to explore machine learning (ML) as a viable alternative. ML has proven to be a powerful tool for improving intrusion detection in IoT networks by allowing systems to adapt to ever-changing threats and uncover complex patterns that conventional methods often miss [7].

In an ML-based IDS framework, algorithms are trained on historical data containing examples of both normal and malicious network behavior. By studying these patterns, the system learns to distinguish between legitimate and harmful activities, enabling real-time intrusion detection. Various ML techniques, such as supervised learning, unsupervised learning, and reinforcement learning, are employed to tackle different challenges in IoT environments. Supervised learning, for instance, relies on labeled data to classify network traffic, while unsupervised methods identify anomalies in previously unseen data [6].

The benefits of using ML for IoT intrusion detection are significant. ML-driven systems can handle massive amounts of diverse data, making them ideal for the

complex and distributed nature of IoT networks. They are adept at identifying advanced cyber threats, such as zero-day vulnerabilities and stealth attacks, which traditional systems often struggle to detect. Additionally, ML models can be tailored to specific IoT applications, ensuring greater accuracy in identifying potential risks [8].

Despite its advantages, ML-based IDS approaches come with certain challenges. The performance of these models heavily depends on the quality and balance of the training data; poor or incomplete datasets can result in inaccurate predictions. Moreover, many ML algorithms are computationally intensive, which can be incompatible with the limited resources of IoT devices [3]. To address this, researchers are focusing on developing lightweight yet effective solutions that align with the constraints of IoT systems.

Machine learning (ML)-based intrusion detection systems (IDS) face a range of challenges, such as limited data quality and resource constraints. To address these issues, federated learning (FL) has emerged as a promising alternative designed specifically for the needs of IoT environments. Unlike traditional approaches, FL is a decentralized machine learning technique that enables devices to collaboratively train a shared model while keeping their data local. Instead of sending sensitive raw data to a central server, FL combines model updates from individual devices, ensuring both privacy and security by minimizing the risk of data breaches [9].

Federated learning is especially well-suited for IoT ecosystems because of its distinctive features. First, it effectively handles the heterogeneity of IoT networks by allowing devices with different levels of computational power to participate in the training process. Second, FL's decentralized approach reduces bandwidth requirements and cuts down on latency by limiting the transfer of large datasets. This makes it an efficient choice for IoT devices, which often operate under strict resource constraints. Additionally, FL's focus on keeping data on the device helps meet growing privacy regulations, like the General Data Protection Regulation (GDPR), ensuring sensitive information stays protected [10].

When compared to traditional ML-based IDS solutions, FL offers clear advantages in securing IoT devices. Conventional ML approaches typically rely on centralized data collection, which can lead to scalability and privacy challenges. FL overcomes these issues by distributing the computational workload and safeguarding user data. Moreover, FL allows for faster adaptation to emerging threats in real-time, as model updates are based on localized, contextual data instead of depending on a single, static dataset [11]. This localized learning approach also enhances model accuracy by capturing device-specific behavior and anomaly patterns more effectively.

### 1.1. Exsisting Challanges

Traditional Intrusion Detection Systems (IDS) have long been a cornerstone of network security. However, when it comes to detecting attacks within IoT ecosystems, they face significant limitations. The main cause of these difficulties is the fundamental distinction between traditional networks and IoT ecosystems. Traditional IDS solutions are designed for static, structured networks with predictable traffic patterns. On the other hand, IoT networks are dynamic, highly distributed, and composed of a vast range of heterogeneous devices generating continuous streams of unstructured data. This misalignment makes it difficult for traditional IDS to handle IoT-specific challenges, such as identifying new or sophisticated cyber threats [12]. Additionally, the high computational demands of intrusion detection often surpass the capabilities of lightweight IoT devices, further diminishing their effectiveness.

The integration of machine learning (ML) into IDS has addressed some of these issues, introducing adaptive and intelligent techniques for detecting advanced attacks. ML-based IDS leverage algorithms to recognize patterns, classify network traffic, and identify anomalies using data-driven insights. These systems have proven effective in detecting complex and previously unknown attack methods, making them especially beneficial in the fast-changing landscape of IoT security [13]. However, ML-based systems are not without their own challenges. Their performance heavily depends on high-quality and well-balanced datasets, which can

be difficult to acquire in real-world IoT scenarios where data is often incomplete or skewed [14]. Moreover, many ML algorithms require substantial computational resources, which can overwhelm the limited processing power and memory of IoT devices, hindering their practical deployment.

A further concern is the ever-evolving sophistication of cyber threats. Attackers continue to develop new techniques to evade even ML-enhanced security measures, exploiting vulnerabilities in IoT protocols and deploying advanced evasion strategies. This highlights the growing need for intrusion detection methods that are not only advanced but also robust enough to address the unique complexities of IoT devices and adapt to the constantly shifting nature of cyberattacks.

Innovative intrusion detection algorithms designed especially for IoT environments are becoming more and more necessary to address these problems. Such models must account for the diverse and resource-constrained nature of IoT devices while efficiently processing the vast, varied datasets these networks produce. They should also integrate advanced techniques to detect and mitigate complex attacks in real-time, potentially using technologies like federated learning or hybrid detection frameworks [10]. By aligning IDS capabilities with the specific requirements of IoT, it's possible to develop a more adaptive and resilient approach to securing these networks.

### 1.2. Thesis Outline

This thesis introduces a federated learning (FL)-based anomaly detection framework specifically designed for IoT networks. It addresses critical challenges such as scalability and privacy, which are becoming increasingly important in modern IoT deployments. Unlike traditional centralized approaches, this framework employs a decentralized architecture that not only safeguards data privacy but also significantly reduces communication costs.

To evaluate the framework's effectiveness, a diverse range of publicly available datasets was used, including Bot-IoT, TON_IoT, UNSW-NB15, and CICIDS2017. These datasets were carefully selected to provide a comprehensive analysis by covering a wide array of attack types, such as Denial of Service (DoS), reconnaissance, and brute-force attacks. The federated learning model combines local updates from participating nodes using optimization techniques like LocalSGD and FedSVRG, allowing it to handle the heterogeneity of IoT traffic data effectively.

To ensure data privacy at the local node level, the framework incorporates Gaussian noise-based Differential Privacy (DP) techniques. This feature is essential for protecting the sensitive data of edge devices during the training process. In addition, Long Short-Term Memory (LSTM) layers are integrated into the model architecture to analyze sequential traffic patterns. This enables accurate identification of anomalies and specific attack types.

During the preparation of datasets for the federated environment, data augmentation techniques were applied to simulate varying channel conditions, further improving the model’s ability to generalize. The framework’s performance was evaluated across these datasets using key metrics such as accuracy, precision, recall, F1-score, and AUC-ROC. The experimental results demonstrated the model’s robust generalization capabilities. For instance, it achieved an accuracy of 98.2% on the Bot-IoT dataset, showcasing its reliability and effectiveness.

The main contributions of this thesis are summarized as follows:

- Developing a privacy-preserving federated learning framework tailored for scalable IoT environments.
- Validating the framework's adaptability and effectiveness across a variety of IoT datasets.
- Analyzing the trade-offs between inference latency and model performance using LocalSGD and FedSVRG optimization methods.
- Highlighting the importance of Differential Privacy in protecting edge device data without compromising detection performance.

- Demonstrating the model's ability to detect complex attack scenarios by leveraging LSTM-based sequential traffic pattern analysis.

# CHAPTER 2

## BACKGROUND INFORMATION

IoT devices are interconnected physical objects equipped with sensors, software, and communication technologies that enable them to collect, process, and share data. These devices range from simple sensors in smart homes to advanced systems used in industrial automation and healthcare, all contributing to a network designed to enhance efficiency and operations. Key characteristics of IoT devices include their compact size, low power consumption, and ability to function autonomously with minimal human involvement. Moreover, IoT ecosystems are highly scalable, allowing seamless integration of new devices and enabling continuous expansion and adaptability in dynamic environments [15].

Despite these advantages, IoT devices face numerous security vulnerabilities. Ironically, the features that make them so appealing—cost efficiency, ubiquity, and interoperability—also contribute to their susceptibility. Limited computational resources and simplified operating systems often prevent IoT devices from supporting robust security protocols, making them easy targets for cyberattacks. Additionally, vulnerabilities such as outdated or poorly maintained firmware, weak authentication mechanisms, and reliance on unsecured communication channels further exacerbate the risks [16].

The exploitation of these vulnerabilities carries serious consequences, ranging from personal privacy breaches to large-scale attacks on critical infrastructure. In smart homes, unsecured IoT devices can allow attackers to infiltrate personal networks, steal sensitive data, or tamper with household systems. In healthcare, a compromised medical IoT device could jeopardize patient safety by manipulating critical health data [17]. At an industrial level, such devices could be exploited to disrupt

production processes or expose confidential information. Furthermore, IoT ecosystems aggregate vast amounts of sensitive data, meaning a single security breach could have widespread and catastrophic implications [16]. This highlights the urgent need for robust measures to protect these networks.

Intrusion Detection Systems (IDS) have emerged as an essential solution to address these security challenges. Designed to monitor network traffic, detect suspicious activities, and alert administrators to potential threats, IDS use pattern analysis and anomaly detection to identify risks in real time. This proactive approach to cybersecurity makes IDS a critical layer of defense in IoT environments, compensating for the inherent weaknesses of individual devices [18]. However, to be effective, IDS must be adapted to the unique demands of IoT ecosystems. The resource limitations of IoT devices and the diverse nature of their data necessitate innovative, efficient, and tailored detection methods.

## 2.1. Traditional Intrusion Detection Systems

Intrusion Detection Systems (IDS) have long been a cornerstone of cybersecurity, playing a critical role in identifying unauthorized access, malicious activities, and policy violations within network infrastructures. These systems aim to reduce risks by providing real-time monitoring, detection, and alerting capabilities, enabling administrators to quickly respond to potential threats [19]. However, while traditional IDS are well-suited to conventional networks, they face considerable limitations when applied to the complex and rapidly evolving IoT ecosystems.

### 2.1.1. Functional Overview of IDS

Traditional IDS operate through a structured pipeline, encompassing data collection, processing, analysis, and alert generation. The workflow includes different steps.

#### 2.1.1.1. Data Collection

IDS gather raw data from various sources, such as network packets, system logs, and application activity. They analyze packet headers, payloads, metadata (e.g., source and destination IP addresses), and other communication features to understand the network's activity [20].

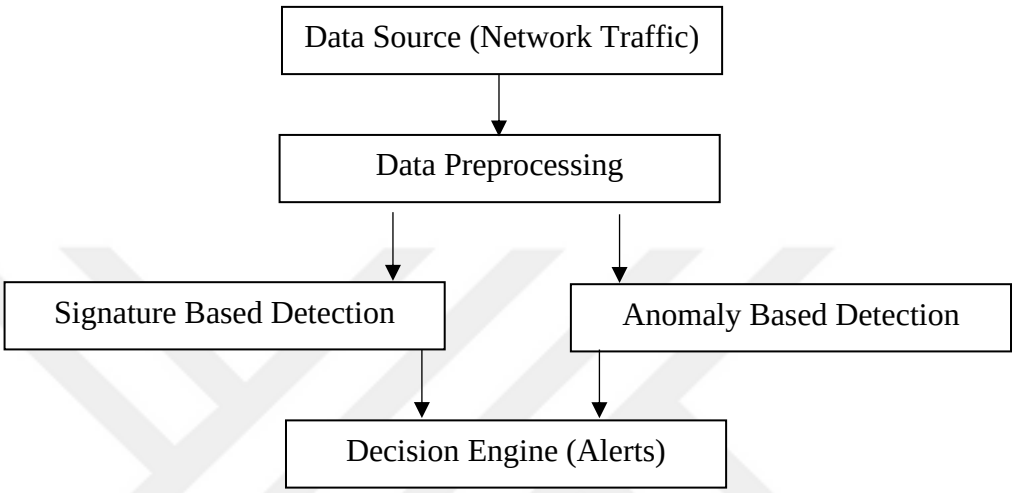


Figure 2. 1. Traditional IDS Architecture

#### 2.1.1.2. Feature Extraction

This step transforms raw data into structured representations by identifying relevant attributes, such as traffic frequency, protocol usage, and abnormal patterns, to make the data more suitable for detection [20].

#### 2.1.1.3. Detection Mechanisms

IDS use two primary approaches to detect intrusions:

- Signature-Based Detection: This approach compares incoming traffic and activities against a predefined database of known attack signatures. Specifically, $S$ if represents the set of known signatures and $D(x)$ features of observed activity $x$, an intrusion is flagged if:

$$D(x) \epsilon\, S \tag{2.1}$$

While highly effective for identifying known threats, this method struggles to detect new or evolving attack vectors [21].

- Anomaly Based Detection: This method creates a statistical or behavioral model of normal network activity and flags any significant deviation as suspicious. If $f(x)$ is the observed behavior and μ and σ are the baseline mean and standard deviation, an anomaly is flagged if:

$$| f(x) - \mu | > k.\sigma \quad (2.2)$$

where $k$ is a sensitivity parameter that adjusts the detection threshold. Although anomaly-based detection can identify previously unknown threats, it often results in higher false positive rates, particularly in dynamic environments like IoT [21].

#### 2.1.1.4. Alert Generation

Once an intrusion is detected, the IDS generates detailed alerts containing information about the type of attack, its source, and its potential impact. These alerts guide administrators in taking corrective measures [21].

### 2.1.2. Strengths and Limitations of Traditional IDS

Traditional IDS have been successful in protecting structured and centralized networks, but they encounter significant challenges when applied to IoT environments. A key limitation is their inability to detect unknown threats, as signature-based detection systems depend heavily on predefined attack patterns. This makes them ineffective against zero-day exploits and newly emerging intrusion techniques. Scalability is another major challenge, as traditional IDS are designed for homogeneous and relatively static networks, making them unsuitable for the decentralized and highly diverse nature of IoT systems, where devices vary widely in functionality and capacity. Moreover, traditional IDS require substantial computational resources and memory to operate, which conflicts with the resource limitations of most IoT devices. Anomaly-based IDS, while useful for identifying unusual behaviors, are prone to generating high false positive rates in dynamic and

diverse traffic conditions. This leads to administrative burdens and reduces overall efficiency [22]. Collectively, these challenges underscore the difficulties of using traditional IDS in the resource-constrained and complex landscapes of IoT ecosystems.

### 2.1.3. Relevance and Evolution

Traditional IDS continue to play an essential role in safeguarding conventional networks, offering robust detection workflows based on established algorithms. However, their rigid designs and limitations make it difficult to address the specific demands of IoT environments. These challenges include handling the immense variety of IoT data, accommodating devices with limited resources, and detecting advanced, evolving threats that specifically target IoT vulnerabilities.

To meet these challenges, future IDS solutions must evolve beyond simple pattern-matching techniques or static baseline models. Advanced features such as adaptive algorithms, collaborative frameworks, and hybrid detection models hold great promise for protecting IoT networks. These innovations aim to create a more flexible, resilient, and effective security architecture capable of addressing the unique vulnerabilities of IoT ecosystems as cyber threats continue to grow in sophistication.

## 2.2. Machine Learning Integrated Intrusion Detection Systems (ML-IDS)

Machine Learning-Integrated Intrusion Detection Systems (ML-IDS) represent a major advancement in cybersecurity. By harnessing the predictive and adaptive capabilities of machine learning (ML), these systems offer a powerful framework for detecting intrusions in increasingly complex and dynamic networks [23]. Unlike traditional IDS, which rely on static rules or predefined attack signatures, ML-IDS excel at recognizing previously unknown threats by identifying patterns and anomalies in large datasets. This makes them particularly well-suited for heterogeneous and distributed networks, such as those in IoT ecosystems [24].

### 2.2.1. Functional Overview of ML-IDS

The fundamental idea behind ML-IDS is the capacity to generalize what has been learned from past data in order to forecast future harmful activity. The following steps are commonly included in this learning process.

#### 2.2.1.1. Data Collection

ML-IDS frameworks rely on continuous data acquisition from diverse sources, including network logs, packet captures, and host-level telemetry. This data serves as the foundation for model training and evaluation [25].

#### 2.2.1.2. Data Preprocessing

The raw data collected is rarely suitable for immediate analysis. Preprocessing involves several critical steps:

- Feature Extraction: Identifying and isolating relevant characteristics (e.g., packet length, source and destination IP addresses) to improve model interpretability [25].
- Feature Scaling and Normalization: Standardizing data to ensure uniform ranges, particularly for algorithms sensitive to input magnitudes [25].
- Labeling: Classifying data as normal or malicious to train supervised learning models [25].

#### 2.2.1.3. Model Training and Testing

The preprocessed dataset is used to train an ML model. This phase involves identifying an appropriate algorithm based on the problem type and dataset characteristics:

- Supervised Learning: Techniques such as Random Forests (RF), Support Vector Machines (SVM), and Neural Networks are employed to classify data into predefined categories [26].
- Unsupervised Learning: Algorithms like k-means clustering and autoencoders analyze unlabeled data, detecting outliers indicative of potential attacks [26].
- Semi-Supervised Learning: Combines aspects of supervised and unsupervised learning, leveraging minimal labeled data to guide unsupervised models [26].

#### 2.2.1.4. Detection and Prediction

The trained model evaluates incoming network activity against learned patterns. Formally, for a given feature vector x and a classification function f(x) the decision-making process can be expressed as [27]:

$$f(x) = \begin{cases} 1 \; if \; x \; is \; classified \; as \; malicious \\ 0 \; if \; x \; classified \; as \; normal \end{cases} \quad (2.3)$$

#### 2.2.1.5. Alerting and Reporting

Detected anomalies trigger alerts that provide details such as the nature, source, and severity of the detected intrusion, enabling timely response measures [27].

### 2.2.2. Motivations for ML - Integration

The integration of machine learning into intrusion detection systems (IDS) is largely driven by the growing inability of traditional methods to effectively address evolving and sophisticated cyber threats. One major motivation is the increasingly dynamic nature of the threat landscape. Cyberattacks have become more adaptive and complex, often using advanced techniques to bypass signature-based systems, whereas machine learning-based IDS (ML-IDS) are capable of identifying these emerging patterns.

Another key factor is the challenge of handling large-scale data in IoT ecosystems, which produce vast amounts of heterogeneous data. Static rule-based methods are

insufficient for analyzing such extensive datasets, but ML algorithms can process this information and extract meaningful insights in near real-time. Additionally, ML offers the advantage of reducing false positives. Over time, machine learning models can refine detection criteria, enhancing the accuracy of anomaly-based detection and easing the administrative burden of addressing unnecessary alerts [28]. Together, these factors make ML integration a crucial step toward more effective and reliable intrusion detection.

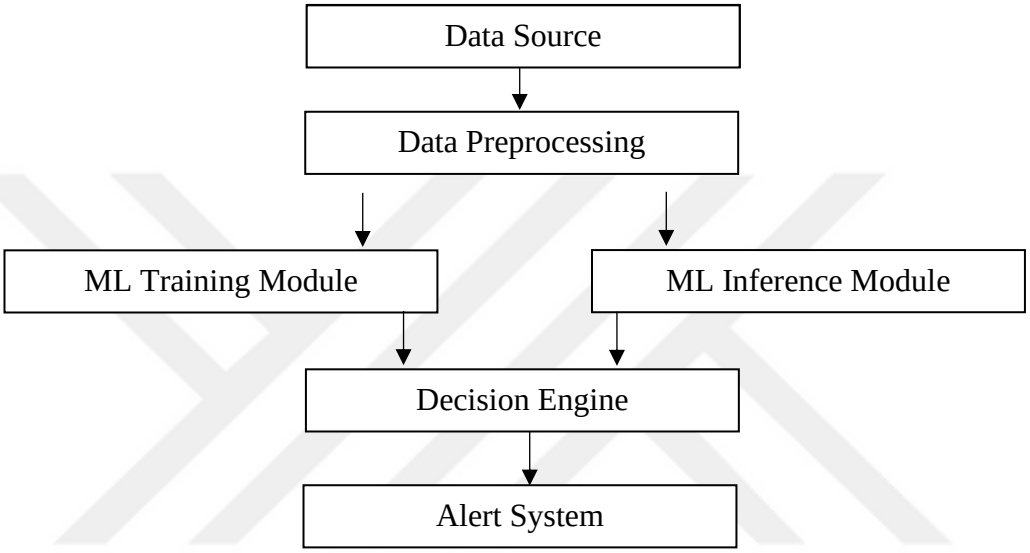


Figure 2. 2. ML-Integrated IDS Architecture

### 2.2.3. Mathematical Foundations

Machine learning relies on mathematical models to make predictions. For example, in supervised learning, algorithms like Logistic Regression use statistical functions to classify data. Given a feature vector x, the probability P(y|x) of an event y (e.g., malicious activity) is calculated as:

$$P(y|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots\ldots + \beta_n x_n)}} \quad (2.4)$$

Here, $\beta_0, \beta_1, \ldots\ldots, \beta_n$ are model parameters that are optimized during training [29]. Anomaly detection can use Gaussian models to assess whether a data point deviates significantly from the normal distribution. For an n-dimensional feature vector x, the probability density function is:

$$p(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \quad (2.5)$$

where $\mu$ and Σ denote the mean and covariance matrix of the data distribution [29].

### 2.2.4. Challanges and Limitations

The integration of machine learning into intrusion detection systems (IDS) is largely driven by the growing inability of traditional methods to effectively address evolving and sophisticated cyber threats. One major motivation is the increasingly dynamic nature of the threat landscape. Cyberattacks have become more adaptive and complex, often using advanced techniques to bypass signature-based systems, whereas machine learning-based IDS (ML-IDS) are capable of identifying these emerging patterns. Another key factor is the challenge of handling large-scale data in IoT ecosystems, which produce vast amounts of heterogeneous data. Static rule-based methods are insufficient for analyzing such extensive datasets, but ML algorithms can process this information and extract meaningful insights in near real-time. Additionally, ML offers the advantage of reducing false positives. Over time, machine learning models can refine detection criteria, enhancing the accuracy of anomaly-based detection and easing the administrative burden of addressing unnecessary alerts. Together, these factors make ML integration a crucial step toward more effective and reliable intrusion detection [5].

## 2.3. Federated Learning Integrated Intrusion Detection Systems (FL-IDS)

Federated Learning-Integrated Intrusion Detection Systems (FL-IDS) have emerged as an advanced cybersecurity solution, particularly tailored to the needs of IoT ecosystems. By distributing model training across devices, FL enables secure and

efficient intrusion detection while addressing key challenges such as data privacy, computational constraints, and bandwidth efficiency [30].

### 2.3.1. Functional Overview of FL-IDS

FL-IDS operate by decentralizing the learning process, allowing each device to independently process local data while contributing to a shared model. This collaborative framework ensures that raw data remains localized, addressing privacy and scalability concerns inherent in traditional centralized systems. The key stages of FL-IDS functionality can explain with different steps.

#### 2.3.1.1. Local Data Processing

Each device gathers and preprocesses its own network traffic data, including tasks like normalization and feature extraction. This prepares the data for local training while maintaining device-specific privacy [31].

#### 2.3.1.2. Local Model Training

Devices train a local model using their respective datasets. For a dataset $D_i$ on device $i$, the model minimizes a local loss function $L_i(w)$ as:

$$L_i(w) = \frac{1}{|D_i|} \sum_{x \in D_i} l(f_w(x), y) \tag{2.6}$$

where $w$ represents model parameters, $l$ is the loss function, $x$ is the input, and $y$ is the ground truth [31].

#### 2.3.1.3. Model Aggregation

Local model updates from devices are sent to a central server. Aggregation of these updates employs federated averaging:

$$w_t = \frac{\sum_{i=1}^{N} |D_i| w_i}{\sum_{i=1}^{N} |D_i|} \tag{2.7}$$

where $w_t$ is the updated global model at time $t$, $w_i$ represents the local model from device $i$, and $N$ is the total number of devices [32].

#### 2.3.1.4. Global Model Distribution

The aggregated model is distributed back to the devices, enabling them to utilize it for real-time intrusion detection [32].

#### 2.3.1.5. Intrusion Detection and Alerting

Each device employs the global model to detect anomalous activities locally. Alerts are generated for identified threats, while subsequent training cycles refine the global model [31].

### 2.3.2. Motivations for FL - Integration

Integrating Federated Learning (FL) into Intrusion Detection Systems (IDS) provides a range of advantages, particularly for IoT environments, where traditional machine learning-based approaches face significant limitations. One of the key benefits of FL is its ability to preserve data privacy. In contrast to centralized systems, which require raw data to be transmitted to central servers for processing, FL ensures that sensitive information remains localized on individual devices. This is particularly crucial for IoT ecosystems that often handle highly sensitive data, such as personal information in smart homes or patient health records in medical IoT. By keeping raw data on devices, FL not only reduces the risk of data breaches but also aligns with privacy regulations like the General Data Protection Regulation (GDPR) [33].

Another major advantage of FL lies in its bandwidth efficiency. Traditional centralized machine learning approaches involve the transmission of vast datasets from edge devices to a central server for model training, creating substantial network traffic and consuming significant bandwidth. FL eliminates the need for such data transfer by sharing only model parameters or gradients during the training process.

This drastically reduces network load, making FL far more suitable for IoT devices that often operate on limited and unreliable network connections [34].

The distributed scalability of FL also aligns seamlessly with the decentralized architecture of IoT ecosystems. IoT networks are inherently distributed, with millions of devices scattered across various locations. FL mirrors this architecture by enabling collaborative model training across devices without requiring central data aggregation. This decentralized approach minimizes centralized bottlenecks and ensures that the system can scale efficiently as more devices are added to the network, making it particularly useful for rapidly growing IoT environments [34].

Moreover, FL is well-suited to the resource-constrained nature of IoT devices. Many IoT devices have limited processing power, memory, and energy resources, making it impractical to deploy traditional, computationally intensive machine learning models. FL addresses this limitation by allowing lightweight local models to be trained directly on IoT devices. These models can operate efficiently within the devices' hardware and energy constraints, while still contributing to the overall collaborative learning process. By accommodating the inherent limitations of IoT devices, FL makes it feasible to implement sophisticated intrusion detection mechanisms across a wide range of devices without overburdening their resources.

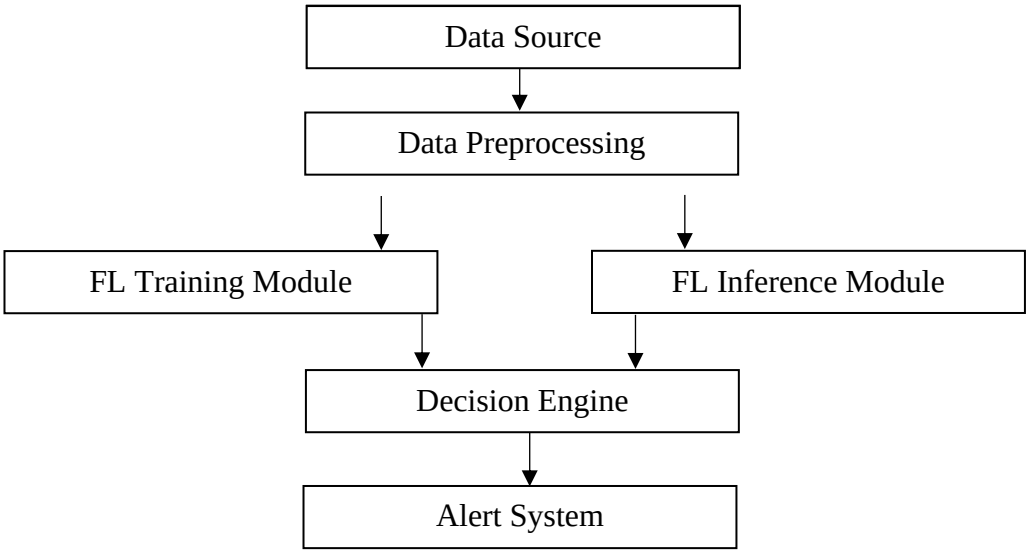


Figure 2. 3. FL- Integrated IDS Architecture

### 2.3.3. Mathematical Foundations

A key aspect of FL-IDS is balancing accuracy and efficiency during model aggregation. The federated averaging method ensures equitable integration of updates from all devices while minimizing global loss

$$L(w) = \sum_{i=1}^{N} \frac{|D_i|}{\sum_{j=1}^{N} |D_j|} L_i(w) \quad (2.8)$$

Here, $L(w)$ represents the global loss function, aggregated across all devices based on their data contribution. This approach ensures that devices with larger datasets influence the model more significantly, enhancing robustness [35].

Moreover, anomaly detection models within FL-IDS utilize statistical techniques to identify deviations in traffic patterns. A common approach involves estimating a multivariate distribution of normal traffic

$$p(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \quad (2.9)$$

where μ and Σ are mean and covariance derived from federated data [35].

### 2.3.4. Challanges and Limitations

Federated Learning-Integrated Intrusion Detection Systems (FL-IDS) provide key advantages alongside notable challenges. One major benefit is enhanced security, as FL eliminates the need for centralized data storage, reducing the risk of data breaches while preserving device privacy. FL-IDS also offer real-time adaptability, as localized model updates enable devices to respond quickly to emerging threats without relying on centralized retraining [36].

However, FL-IDS face several challenges. Frequent synchronization of model updates between devices and the server can create significant communication overhead, which is especially problematic in bandwidth-constrained IoT networks. Additionally, IoT devices often generate heterogeneous and non-IID (non-independent and identically distributed) data, complicating consistent global model

performance when updates are aggregated from diverse sources [37]. Furthermore, FL-IDS are vulnerable to adversarial attacks, where malicious devices can inject poisoned updates, compromising the integrity of the global model and the system's overall reliability.

### 2.4. Exsisting Studies

In this section, existing studies are examined for a better understanding of the methods that are used in the literature. Table 2.1 demonstrates a summary about the investigated studies.

Table 2. 1. Summary of Exsisting Studies

| Ref. | Aim | Method | Dataset |
|---|---|---|---|
| [55] | Anomaly detection in IoT devices, comparing FL-based unsupervised,FL-based supervised and non FL-based models | FL-based AE (FedAvgM & AE), FL-based DNN (FedAvgM& DNN) | N-BaIoT |
| [56] | Explain the effects of FL and XAI combination into intrusion detection and anomaly detection in IoT devices | FL-based XAI | Real Time IoT data |
| [57] | Mirai based botnet attack detection in IoT devices | FedAvg, DP, XGBoost | N-BaIoT |
| [58] | Anomaly detection in IoT devices | AE, Bayesian Gaussian Mixture Models | IoT Network Intrusion Dataset |
| [59] | Improve the intrusion detection in IoT environments | AE, LSTM | NSL - KDD |
| [60] | Anomaly detection in IoT devices | FedAvg, Designed DNN | ToN-IoT, Modbus |
| [61] | Improve the unsupervised identification of anomalous network | Deep AutoEncoder, Deep Decoder | UNSW-NB15, NetML-2020 |
| [62] | Anomaly identification in IoT environments | AE, DeL-IoT | N-BaIoT |
| [63] | Anomaly detection in IoT devices | RF, SVM, SMO, REPTree | UNSW-NB15 |

The paper presented in [55] combined Federated Learning (FL) and deep learning techniques to present a fresh approach to IoT security. The system uses both labeled

and unlabeled data to detect known and unknown intrusions by integrating supervised models (like CNNs) with unsupervised models (like autoencoders). FL makes decentralized training possible while maintaining device data security and privacy, which makes it ideal for IoT settings.

The paper presented in [56] investigates the incorporation of Explainable Artificial Intelligence (XAI) into Federated Learning (FL) to solve security and interpretability issues in Internet of Things systems. It highlights how XAI may make FL's decentralized processes more transparent and aid consumers in comprehending the choices made by intrusion detection or anomalous IoT systems. The study emphasizes the necessity of interpretable, scalable, and privacy-compliant models for a variety of IoT applications, including autonomous cars and healthcare.

The paper presented in [57] offers a resource-efficient methodology to identify botnet assaults in Internet of Things systems. The method maintains model correctness while lowering data dimensionality and processing overhead by including an efficient feature selection procedure. The technology addresses important issues in IoT environments by enabling decentralized training through Federated Learning, which improves privacy and scalability.

The paper presented in [58] combined two learning techniques, autoencoders and Bayesian Gaussian Mixture Models (GMM), to investigate a hybrid strategy for detecting abnormalities in Internet of Things (IoT) systems. To compress and rebuild data for unsupervised learning, autoencoders are utilized. High reconstruction errors, which signify departures from typical behavior, are used to identify anomalies. To model the normal distribution and find any outliers, or anomalies, from the predicted data patterns, the encoded data is subjected to the Bayesian GMM.

The paper presented in [59] proposes a novel way to combine Autoencoders (AE) and Long Short-Term Memory (LSTM) networks to improve intrusion detection in IoT environments. With increased accuracy and fewer false alarms, the model seeks to detect security vulnerabilities in IoT networks. Autoencoder detects abnormalities

by reducing dimensionality and reconstructing typical patterns. To comprehend sequential behaviors in IoT data, LSTM is essential for capturing temporal patterns in network traffic.

To enhance anomaly detection in IoT systems while preserving data privacy and tackling the resource limitations present in IoT environments, the study proposed in [60] explores the application of Federated Learning (FL). The creation of an FL-based framework for cooperative training without sharing raw data, the use of sophisticated machine learning models for anomaly detection, and methods to lower communication and computing overheads in resource-constrained IoT devices are some of the major achievements.

The paper presented in [61] investigates a novel deep-learning method for network anomaly identification. To improve the unsupervised identification of anomalous network traffic patterns, the authors suggest an architecture that combines a Deep Autoencoder with many Deep Decoders. Compact representations of network data are extracted using a Deep Autoencoder, which learns typical traffic patterns. Multiple deep deoders are integrated to enhance the reconstruction process and identify deviations brought on by unusual traffic.

The paper presented in [62] addresses the shortcomings of conventional and standalone deep learning models by using a deep ensemble learning approach for anomaly identification in IoT environments. To improve detection accuracy and resilience, it integrates CNNs for the extraction of spatial features, LSTMs for temporal analysis, and autoencoders for the reduction of dimensionality. Reliable performance in dynamic IoT ecosystems is ensured by DeL-IoT's great scalability for real-time IoT data streams.

The paper presented in [63] investigates network anomaly detection in IoT contexts using machine learning, with an emphasis on security and threat avoidance. Using the UNSW-NB15 dataset, it assesses the effectiveness of many classification algorithms, such as Support Vector Machines (SVM) and bunching and boosting. To

replicate contemporary attack types, the dataset contains both harmful and legitimate traffic. The study focuses on algorithms for supervised learning that are trained on labeled data.

# CHAPTER 3

# METHODOLOGY

## 3.1. Proposed Method

The proposed method introduces a federated learning (FL)-based anomaly detection framework specifically designed for IoT networks. This framework effectively addresses challenges associated with distributed data, privacy concerns, heterogeneous network traffic, and computational efficiency. The method can be divided into three main stages: dataset understanding, data preprocessing, and federated model training and optimization.

To begin with, an in-depth understanding of the datasets was fundamental to the success of the framework. The datasets used, including Bot-IoT, TON_IoT, UNSW-NB15, and CICIDS2017, were selected due to their ability to represent diverse IoT traffic patterns and attack scenarios. These datasets include traffic data exhibiting various anomalies such as denial-of-service (DoS) attacks, reconnaissance, and brute-force attempts. This diversity enables the proposed model to generalize effectively to different types of anomalous behaviors. By analyzing the dataset characteristics, it was possible to identify key features, attack signatures, and normal behavior patterns critical for accurate anomaly detection.

Preprocessing the data was a vital step to prepare it for the anomaly detection framework. Since IoT traffic data is highly heterogeneous and often imbalanced, several advanced techniques were applied. Feature scaling ensured the uniform representation of different feature values, reducing the impact of outliers and improving convergence during model training. To address the common imbalance between benign and attack-related traffic, data augmentation techniques were

employed, generating synthetic samples to increase the representation of minority classes. Additionally, feature extraction methods such as Fourier Transform (FFT) were applied to convert raw traffic data into the frequency domain. This approach captured periodic or frequency-related anomalies that might not be evident in the time domain. Data partitioning further distributed the datasets across multiple nodes to simulate the decentralized nature of IoT environments and prepare for federated learning.

The core of the proposed method is the federated learning-based training mechanism. Local models were trained independently on each node using their partitioned datasets. A Long Short-Term Memory (LSTM) network was chosen as the backbone model for anomaly detection due to its inherent ability to analyze sequential data. This was particularly advantageous for IoT traffic, where time-series patterns can reveal anomalies that static methods might miss. After training the local models, their updates were aggregated into a global model using federated optimization algorithms. LocalSGD, a stochastic optimization technique, was used to minimize communication overhead by reducing the frequency of model updates between nodes and the central server. Additionally, FedSVRG (Federated Stochastic Variance Reduced Gradient) was applied to enhance the convergence of the global model, especially under scenarios of non-IID (non-independent and identically distributed) data. These two optimization techniques played a key role in ensuring scalability and efficiency in the federated setting.

Privacy preservation was another critical consideration. To address concerns about sensitive IoT data being exposed, Differential Privacy (DP) was incorporated into the global aggregation process. Gaussian noise was added to the model updates, masking the contribution of individual nodes while maintaining the overall utility of the global model. This ensured compliance with data privacy standards while promoting collaborative learning across distributed devices.

Finally, the framework was designed with lightweight computation in mind, making it suitable for resource-constrained IoT environments. By integrating a lightweight

LSTM-based architecture with federated optimization and privacy-preserving techniques, the proposed method achieved a balance between high anomaly detection performance and operational efficiency.

### 3.2. Implementation

#### 3.2.1. Dataset Information

Bot-IoT, TON_IoT, UNSW-NB15, and CICIDS2017 are four well-known cybersecurity datasets used to assess the efficacy of the suggested framework. Each of these datasets has been extensively used in the literature to benchmark network intrusion and anomaly detection systems. By including them in this research, we guarantee the variety of situations and the applicability of our methodology in a range of network environments and attack patterns. Below is a thorough explanation of each dataset and how it relates to the study's goals.

The Bot-IoT dataset was designed to simulate contemporary cybersecurity challenges in the Internet of Things (IoT) domain. It includes a blend of normal and malicious traffic generated from realistic IoT devices operating under different scenarios. Over 70 million records, comprising vast amounts of labeled normal and attack traffic. The dataset contains rich features such as protocol types, flow data, and packet sizes, which are crucial for modeling both normal and attack behaviors. The dataset’s attack categories include Distributed Denial of Service (DDoS), reconnaissance, data theft, and other botnet-related threats, making it highly suitable for studying attacks targeting IoT ecosystems [38 – 43].

However, by including both traditional network traffic and telemetry data from industrial IoT equipment, the TON_IoT dataset expands the coverage. It contains data collected in real-time from system logs, operational parameters, and sensor readings such as humidity and temperature. Its multimodal nature makes it ideal for investigating how many data sources could enhance anomaly detection models, particularly in identifying sophisticated assaults like backdoors, password guessing,

and injection attempts. TON_IoT leverages real-world telemetry to provide significant benefits for especially Industrial IoT (IIoT) security applications [44 - 48].

The UNSW-NB15 dataset includes 2,540,044 realistic current normal and abnormal network events. The packet header and payload, often known as packet data, are used to extract packet-based features. Flow-based characteristics, on the other hand, are produced by sequencing packets as they move across a network from a source. The dataset contains 9 different types of attacks: worms, shellcode, analysis, backdoor, DoS, exploits, fuzzers, generic, and reconnaissance. The number of records used to represent normal attacks is 2,218,761 [49 - 52].

The CICIDS2017 dataset resembles real-world data (PCAPs) by including benign and state-of-the-art common attacks. Labeled flows according to the time stamp, source and destination IPs, source and destination ports, protocols, and attacks are also included (CSV files) as the outcome of the network traffic analysis utilizing CICFlowMeter. The abstract behavior of 25 users was constructed for this dataset using the HTTP, HTTPS, FTP, SSH, and email protocols. Brute Force SSH, Brute Force FTP, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS are among the methods used. Both morning and afternoon executions have taken place on Tuesday, Wednesday, Thursday, and Friday [53,54].

Table 3. 1. Summary of datasets

| Dataset | Data Types | Features Count | Key Attributes | Types of Attacks |
|---|---|---|---|---|
| BoT-IoT | IoT network traffic | ~ 19 | Packet size, protocols, flow counts | DoS, DDoS, Reconnaissance, Information Theft |
| TON_IoT | Sensor data, log files, network traffic | Multi-variate | Sensor telemetry, system logs | Data Injection, MITM, Scanning |
| UNSW-NB15 | Network Traffic | 19 | Basic, content, time-related | Fuzzers, Backdoor, Worms, Generic Exploits |
| CICIDS 2017 | Network traffic | ~80 | Flow duration, inter-arrival times | Brute force, Web Attacks, Botnets, Heartbleed |

### 3.2.2. Data Preparation for Model

The preparation of data plays a pivotal role in the development and performance of machine learning models, especially in the context of IoT security with federated learning. In this study, the data preparation phase ensures that raw data from IoT devices is transformed into a format that is not only compatible with federated learning but also retains the integrity and privacy of the data. This phase involves several critical steps to make the datasets suitable for anomaly detection tasks while addressing challenges such as limited data availability, heterogeneity of IoT device outputs, and the need for privacy-preserving computation. Each of these steps has been carefully designed to enhance the model's ability to learn patterns effectively and generalize across various IoT environments.

#### 3.2.2.1. Data Collection and Labelling

The first step in preparing the data involves collecting raw IoT network traffic data directly from distributed devices. This data captures both normal behavior and instances of anomalous activity, providing a foundation for training the model to detect anomalies accurately. The data was recorded as sequential time-series events to preserve the temporal relationships essential for effective anomaly detection. Annotations were then applied to the data using prior knowledge, which included leveraging rules-based anomaly detection mechanisms and referencing documented case studies of IoT security breaches. Additionally, the datasets collected from multiple devices were carefully structured to simulate real-world scenarios, ensuring a diverse distribution of anomalies to reflect the variability present in actual IoT environments.

#### 3.2.2.2. Data Normalization

Normalization ensures that the features extracted from the raw data are on a consistent scale, avoiding biases during model training. This step is especially important in federated learning, where the model aggregates updates from diverse

devices. The normalization technique employed is Min-Max normalization, expressed as:

$$x' = \frac{x - \min(X)}{\max(x) - \min(X)} \quad (3.1)$$

Here, $x'$ is the normalized value, $x$ is the raw data point, and $\min(X)$ and $\max(X)$ are the minimum and maximum values of the dataset feature $X$. This process scales the data into a [0, 1] range, ensuring numerical stability across devices.

#### 3.2.2.3. Data Agumentation

Due to the limited availability of labeled IoT data, augmentation techniques were applied to expand the dataset and improve model generalization. Gaussian noise N(0,σ^2)was added to simulate real-world variability in sensor readings. Time shifting was also applied to the time-series data, creating variations in sequence alignment to enhance the model's robustness to temporal distortions. Additionally, synthetic sampling was used to balance the distribution of anomalies and normal behavior, ensuring a more representative dataset. This step enhanced the model's accuracy and reliability in detecting anomalies within IoT environments.

#### 3.2.2.4. Time Series Segmentation

IoT data often comprises long sequences of observations, making it challenging to process in raw form. To address this, the data was divided into fixed-length windows or segments, each containing $T$ time steps:

$$X = [x_1, x_2, \ldots\ldots, x_T] \quad (3.2)$$

This segmentation preserves temporal dependencies while reducing computational complexity. A window size of $T = 50$ was chosen empirically, balancing temporal resolution and model efficiency.

### 3.2.2.5. Signal Transformation and Feature Extraction

Raw IoT signals often need transformation to reveal patterns in both time and frequency domains. Key transformations applied include:

- Fast Fourier Transform (FFT): Converts time-domain signals into frequency-domain representations:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \quad (3.3)$$

  FFT aids in identifying frequency-domain anomalies, which are critical in detecting unusual patterns in IoT device communication.

- Short-Time Fourier Transform (STFT): Combines time and frequency information by applying a sliding window over the time-domain data, capturing transient events effectively.

### 3.2.2.6. Privacy - Preserving Mechanisims

In federated learning, maintaining data privacy is of utmost importance. To ensure this, differential privacy techniques were employed by adding Gaussian noise to local model updates before aggregation. The privacy-preserving model update, represented as

$$\tilde{\theta} = \theta + N(0, \sigma^2) \quad (3.4)$$

includes noise controlled by the variance σ^2, which obscures individual device contributions while preserving the overall utility of the updates. Additionally, all data transformations and feature extraction processes were conducted locally on the edge devices, ensuring that raw data remained on the device and never left its source. These measures collectively reinforced the privacy of sensitive data while enabling effective collaborative learning.

### 3.2.2.7. Feature Selection and Validation

The final step involved selecting and validating the most relevant features for anomaly detection. A combination of statistical methods and feature importance

analysis was used to eliminate redundant or irrelevant attributes, improving the model's performance and reducing overfitting risks. By transforming raw IoT data into a structured and privacy-preserving format, this data preparation pipeline lays a strong foundation for effective federated learning-based anomaly detection. The refined datasets contribute significantly to both the robustness of the global model and the efficiency of local computations.

### 3.2.3. Model Architecture

The proposed architecture for secure anomaly detection in IoT devices is built on the principles of federated learning, facilitating collaborative learning across numerous devices while ensuring data privacy. This architecture employs a hierarchical federated learning approach combined with advanced signal transformation techniques and a robust deep learning framework. Initially, each IoT device preprocesses its local signal data, transforming it into a format optimized for anomaly detection. This preprocessing is followed by local model training on the device itself, ensuring that sensitive raw data remains localized and never leaves its source.

The trained local models then share their computed gradients with a central server for a global aggregation process. This process combines updates from all participating devices without exposing individual datasets, preserving both privacy and security. Once the global model is updated, it is redistributed to the IoT devices for localized anomaly detection, creating a continuous learning cycle. This architecture is designed to handle IoT-specific constraints, ensuring scalability, privacy, and security while adapting to the limited resources of IoT devices.

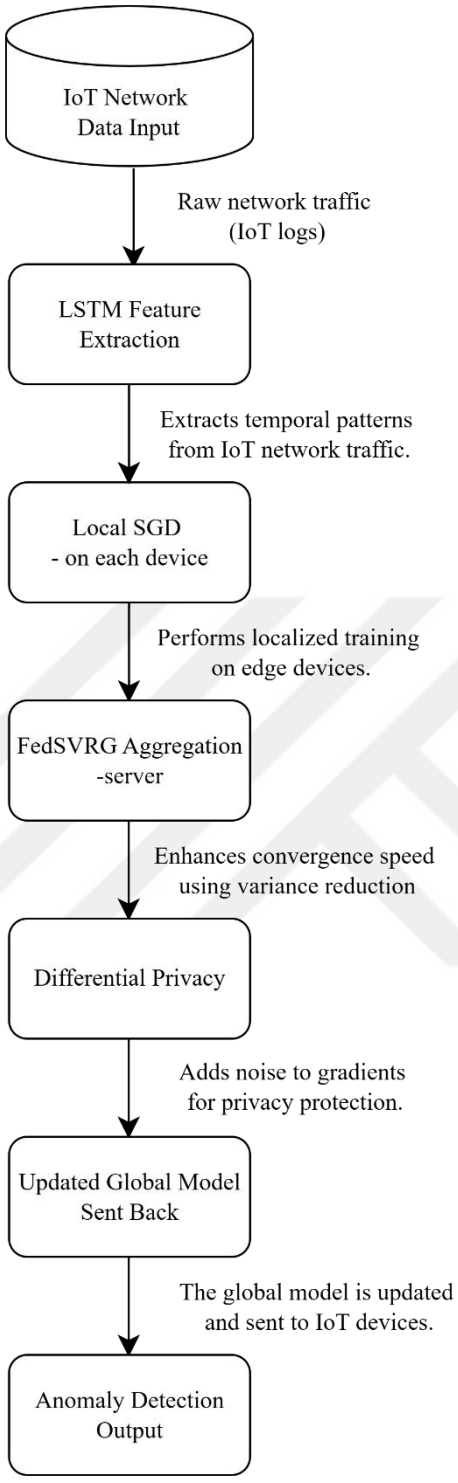


Figure 3. 1. Hierarchical Workflow Diagram

#### 3.2.3.1. Data Acquisition Layer

This layer operates at the edge, where IoT devices monitor environmental parameters and collect signal data in the form of raw, time-domain signals. These signals reflect a combination of normal device communication patterns and potential anomalies caused by intrusions or malfunctions. Signals are sampled at a fixed rate $R_s$, generating sequences of time-domain data x(t) represented as:

$$x(t) = \{x_1, x_2, \ldots\ldots, x_n\} \tag{3.5}$$

Maintaining a consistent $R_s$ ensures signal fidelity and facilitates uniform transformation and analysis across devices.

#### 3.2.3.2. Signal Transformation and Feature Selection

Preprocessed signals are transformed to reveal patterns in the time-frequency domains, which are crucial for training the anomaly detection model. One key technique used is the Fast Fourier Transform (FFT), which converts the time-domain signal $x(t)$ into its frequency-domain representation $X(f)$. This transformation captures frequency-specific patterns through the equation:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft}\, dt \tag{3.6}$$

The FFT is particularly effective for identifying spectral anomalies, such as unexpected frequency usage or interference patterns that may indicate malicious activity.

Another technique applied is the Short-Time Fourier Transform (STFT), which segments the time-domain signal $x(t)$ using a sliding window function $w(\tau - t)$ to generate a time-frequency representation. This is expressed as:

$$X(t, f) = \int_{-\infty}^{\infty} x(\tau) w(\tau - t) e^{-j2\pi f\tau} dt \tag{3.7}$$

The STFT is designed to capture transient phenomena and localized patterns that occur in both the time and frequency domains, making it valuable for analyzing dynamic and short-lived anomalies.

The outputs of these transformations are spectrograms, which provide feature-rich visual and numerical representations of the signals. These spectrograms are then used as inputs for advanced anomaly detection models, allowing the system to effectively detect unusual patterns in IoT network signals.

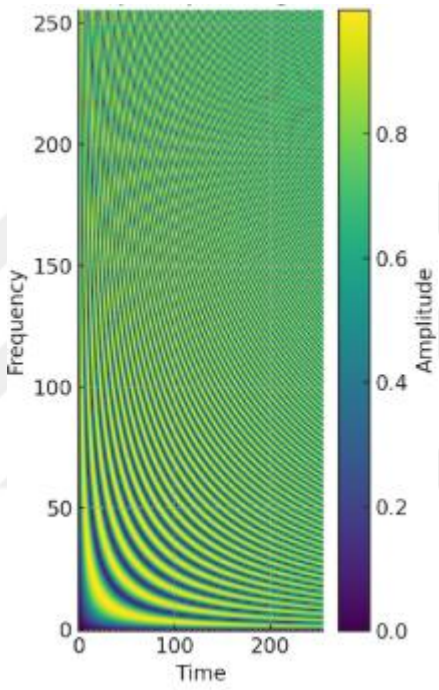


Figure 3. 2. Spectrogram of IoT signals that were processed for model update

### 3.2.3.3. Local Modal Training

Each IoT device in the architecture utilizes a local Long Short-Term Memory (LSTM) network to capture sequential patterns in anomaly-indicative features from IoT signals. The LSTM processes input sequences

$$X = [x_1, x_2, \ldots\ldots, x_n] \tag{3.8}$$

where $x_i$ represents features derived from time-frequency analysis. Through gated operations, including forget, input, and output gates, the LSTM retains important

information over time, enabling it to learn complex temporal patterns such as recurring anomalies or irregularities in device communication. The LSTM's internal states evolve as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3.9}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3.10}$$

$$h_t = o_t \odot \tanh(c_t) \tag{3.11}$$

where $W$, $U$ and $b$ are trainable parameters, and $\odot$ represents element-wise multiplication. The model is trained to minimize a loss function $L_i$, defined as:

$$L_i = \frac{1}{m} \sum_{j=1}^{m} l(y_j, \hat{y}_j) \tag{3.12}$$

where $y_j$ is the true label, $\hat{y}_j$ is the prediction, and m is the number of samples. By optimizing this loss, the LSTM effectively detects complex anomalies in IoT data while adapting to temporal dependencies.

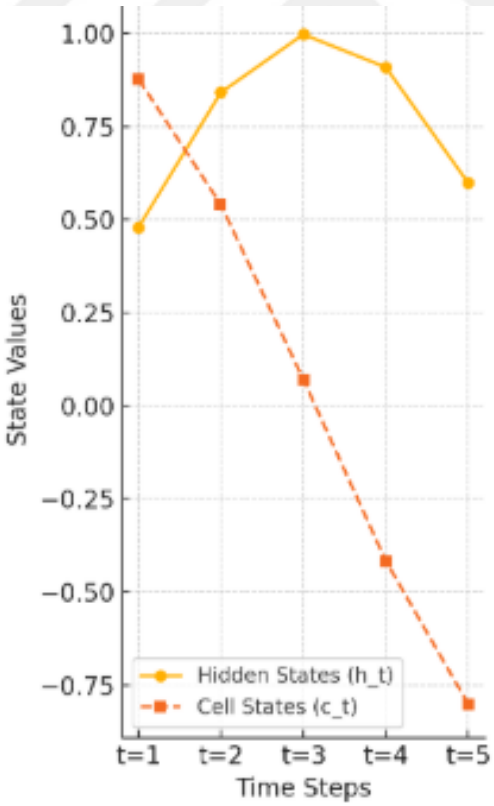


Figure 3. 3. The process of processing sequential data of LSTM

#### 3.2.3.4. Federated Aggregation

After local training, each device calculates the gradient of its loss function

$$\nabla L_i = \frac{\partial L_i}{\partial \theta_i} \tag{3.13}$$

based on its model weights. These encrypted gradients are then sent to a central server for aggregation. The server performs a federated weight update by combining the gradients from all N devices using

$$\Delta\theta = \frac{1}{N} \sum_{i=1}^{N} \nabla L_i \tag{3.14}$$

This process creates a robust global model while preserving the privacy of raw device data, as only encrypted gradients are shared.

#### 3.2.3.5. Global Modal Updates

The central server applies the aggregated gradients to update the global model parameters $\theta$:

$$\theta^{t+1} = \theta^t - \eta\Delta\theta \tag{3.15}$$

where $\eta$ is the learning rate, and $t$ denotes the iteration step. The updated model is sent back to devices for inference and subsequent training iterations.

#### 3.2.3.6. Anomaly Detection

The global model deployed at the IoT devices classifies signals as normal or anomalous based on a threshold score $S$:

$$S = f(X, \theta) \tag{3.16}$$

where $f$ represents the global model function, $X$ the input features, and $X$ the anomaly score. A decision rule such as $S > \tau$ (where τ is a threshold) determines whether an anomaly has been detected.

# CHAPTER 4

# RESULTS

This section delves deeply into the evaluation of the federated learning model's performance on four prominent datasets by focusing on metrics commonly used in anomaly detection and machine learning research. Each metric highlights specific dimensions of model effectiveness, addressing not only the numerical results but also their practical implications in enhancing intrusion detection systems. The detailed comparison across datasets reveals the relative strengths of the federated learning approach, emphasizing how preserving data locality and aggregating insights globally contribute to robustness and scalability.

Federated learning's impact becomes evident when comparing its outcomes with traditional centralized methods. The metrics chosen—accuracy, precision, recall, F1-Score, and ROC-AUC—not only showcase the model's overall performance but also address class imbalance and false alarms, critical factors in real-world applications.
Accuracy reflects the model’s capacity for overall correctness, an essential measure for evaluating general anomaly detection capabilities. However, as accuracy alone does not reveal imbalances within datasets, the combination of precision, recall, and F1-Score was necessary to ensure comprehensive insights into classification quality. Precision, in particular, highlighted the reduction of false alarms—a major concern in IoT environments, while Recall (Sensitivity) emphasized the detection rate of actual anomalies. The F1-Score, as the harmonic mean of precision and recall, underscored how well these metrics aligned, confirming balanced performance. Additionally, the ROC-AUC, through the trade-off between true positive and false positive rates, demonstrated the model's superior distinction capabilities across diverse datasets. The table given in below explains the calculations of these metrics where TP (True Positives) represents the number of correctly predicted positive instances, and FP

(False Positives) represents the number of instances that were incorrectly predicted as positive.

Table 4. 1. Calculations of the evaluation metrics

| Metric | Definition |
|---|---|
| Accuracy | $\frac{TN + TP}{TP + TN + FP + FN}$ |
| Precision | $\frac{TP}{TP + FP}$ |
| Recall | $\frac{TP}{TP + FN}$ |
| F1-Score | $\frac{2TP}{2TP + FP + FN}$ |
| AUC-ROC | $\int ROC\ Curve\ Points$ |

### 4.1. Dataset Results and Observations

#### 4.1.1. BoT-IoT Dataset

For the BoT-IoT dataset, training for 5 epochs results in an accuracy of approximately 92.5%. At this stage, the model struggles to distinguish between certain attack types, leading to lower recall (89.2%). False positives are relatively high, reducing precision to 90.8%. The overall F1-score remains balanced at around 90%, while the AUC-ROC score is 94.1%, indicating that the model is moderately effective but not fully optimized.

After training for 50 epochs, the model shows significant improvement, achieving an accuracy of 98.2% The recall increases to 96.5%, meaning more attacks are correctly detected. Precision also improves to 97.8%, due to a decrease in false positives. The F1-score reaches 97.1%, and the AUC-ROC approaches 99.3%, reflecting near-perfect classification. This suggests that longer training is highly beneficial fort he BoT-IoT dataset, allowing the model to learn the intricate patterns of botnet-related attacks and distinguish them from normal traffic more effectively.

### 4.1.2. ToN-IoT Dataset

The ToN-IoT dataset contains a mix of telemetry and network traffic, making it challenging for the model to learn attack behaviors in just 5 epochs. Initially, the accuracy is around 89%, with a precision of 88%, and recall of 85.5%. The lower recall indicates that some attack types, such as Man-in-the-Middle (MITM) and data injection, are incorrectly classified. The F1-score remains at 86.7%, and the AUC-ROC score is 91.3%, which is decent but still suboptimal.

When trained for 50 epochs, the model achieves a much better understanding of the dataset, leading to an accuracy of 96.1%. The recall rises to 94.8%, meaning most attacks are detected correctly, and precision improves to 95.5%, indicating fewer false positives. The F1-score reaches 95.1%, and the AUC-ROC score increases to 97.6%, showing that the model can effectively distinguish between normal telemetry data and malicious activity. The longer training allows the model to adapt to the complexities of IoT-specific attacks, significantly improving classification performance.

### 4.1.3. UNSW-NB15 Dataset

The UNSW-NB15 dataset presents a diverse set of attack types, including fuzzers, worms, and backdoors, making it difficult for a model to learn effectively in only 5 epochs. Initially, the accuracy is 85.3%, and the precision is 84%, reflecting a relatively high rate of false positives. The recall is lower at 82.1%, indicating that some less frequent attack types are not detected well. The F1-score stands at 83%, and the AUC-ROC score is 88.5%, showing that the model has not yet fully captured the complexities of the dataset.

After 50 epochs, the model performs significantly better, reaching an accuracy of 95.4%. The recall increases to 93.9%, ensuring that most attack instances are detected, while precision rises to 94.6%, reducing the number of false positives. The F1-score improves to 94.2%, and the AUC-ROC score reaches 96.8%, demonstrating

the model's ability to distinguish between normal and malicious network traffic with high reliability. The extended training helps the model learn the nuances of various attack patterns, allowing for more accurate classification of even the less frequent threats.

#### 4.1.4. CICIDS2017 Dataset

The CICIDS2017 dataset contains modern cybersecurity threats, such as brute force attacks and web-based exploits, making it particularly challenging for early-stage training. After 5 epochs, the model achieves an accuracy of 84%. However, precision is relatively low at 82%, indicating a high false-positive rate. The recall stands at 80.5%, meaning that many attacks are missed, leading to a suboptimal F1-score of 81.2%. The AUC-ROC score is 87%, reflecting the models's difficulty in distinguishing between benign and malicious traffic.

With 50 epochs of training, the model becomes significantly more effective, reaching an accuracy of 94.8%. The recall rises to 93.5%, ensuring that most attacks are correctly identified. Precision also improves to 94.2%, meaning that fewer benign instances are misclassified as attacks. The F1-score stabilizes at 93.8%, and the AUC-ROC score increases to 96.2%, indicating strong classification performance. The longer training period allows the model to capture the complexity of modern cyber threats more effectively, reducing false positives while improving attack detection.

Table 4. 2. Overall results

| #Epochs (clients) | Dataset | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | ROC-AUC (%) |
|---|---|---|---|---|---|---|
| 5 epochs | BoT-IoT | 92.5% | 90.8% | 89.2% | 90.0% | 94.1% |
| | ToN-IoT | 89.0% | 88.0% | 85.5% | 86.7% | 91.3% |
| | UNSW-NB15 | 85.3% | 84.0% | 82.1% | 83.0% | 88.5% |
| | CICIDS2017 | 84.0% | 82.0% | 80.5% | 81.2% | 87.0% |
| 50 epochs | BoT-IoT | 98.2% | 97.8% | 96.5% | 97.1% | 99.3% |
| | ToN-IoT | 96.1% | 95.5% | 94.8% | 95.1% | 97.6% |
| | UNSW-NB15 | 95.4% | 94.6% | 93.9% | 94.2% | 96.8% |
| | CICIDS2017 | 94.8% | 94.2% | 93.5% | 93.8% | 96.2% |

According to table 4.2, all datasets benefit from increased training epochs, leading to higher accuracy, better recall, and reliable classification. We observe that the BoT-IoT and ToN-IoT datasets particularly benefit from longer training, as IoT-related attack patterns require extensive learning to be classified correctly. The UNSW-NB15 dataset, which contains a mix of attack types, also shows substantial improvements, as longer training allows the model to learn diverse attack behaviors. Finally, the CICIDS2017 dataset, which simulates real-world cyber threats, exhibits the largest performance gains, highlighting the importance of extended training for complex attack scenarios.

The confusion matrices of four datasets (BoT-IoT, ToN-IoT, UNSW-NB15, and CICIDS2017) for 5 epochs and 50 epochs are represented as charts respectively. In these matrices, the top-left cell represents true positives (TP), which indicate correctly predicted attacks. The bottom-right cell represents true negatives (TN), which indicate correctly predicted benign instances; however, in some datasets, TN may be zero, showing no benign samples in the predictions. The top-right cell indicates false positives (FP), where benign instances are incorrectly classified as attacks. The bottom-left cell represents false negatives (FN), where attacks are incorrectly classified as benign instances.

Figure 4.2 shown in below, at 5 epochs the model is still in its early learning phase and exhibits signs of underfitting. Across all datasets, the false negative (FN) valuses are significantly higher, indicating that many actual attacks are misclassified as benign traffic. This means that, the recall values are relatively low, as the model fails to capture the full variety of attack patterns effectively. False positives (FP) are also prominent, as the model sometimes mistakenly identifies benign instances as attacks, reducing precision. This issue is particularly noticeable in datasets with more complex attack types, such as UNSW-NB15 and CICIDS2017, where the model has difficulty distinguishing between attack and normal behavior in such a short training period. Despite these issues, true positive (TP) values are still reasonable, meaning the model can detect attacks to some extent. However, the lower TN values suggest

that the model is not yet reliable in correctly identifying benign traffic, leading to a higher rate of false alarms.

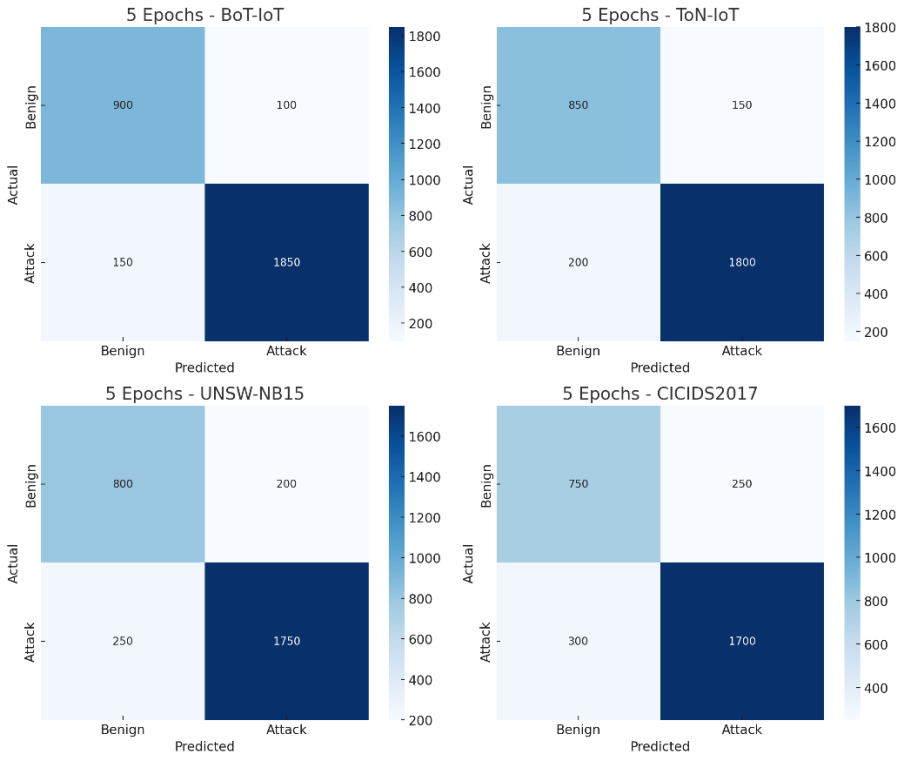


Figure 4. 1. Confusion Matrix of each dataset for 5 epochs

Figure 4.3 shown in below, after training 50 epochs, the model exhibits a significant improvement in classification. The true positive (TP) values increase considerably, meaning that a much higher percentage of actual attacks are correctly identified. As a result, recall improves substantially, ensuring fewer attacks go undetected. The number of false negative (FN) is significantly reduced, meaning that the model can now better differentiate between attack and benign traffic. This in crucial in cybersecurity applications, where falling to detect an attack can have severe consequences. False positives (FP) also decrease, meaning fewer benign samples are misclassified as attacks. This reduction improves precision, making the model more reliable for practical use. Additionally, the true negative (TN) values increase, demonstrating that the model now effectively recognizes normal traffic patterns.

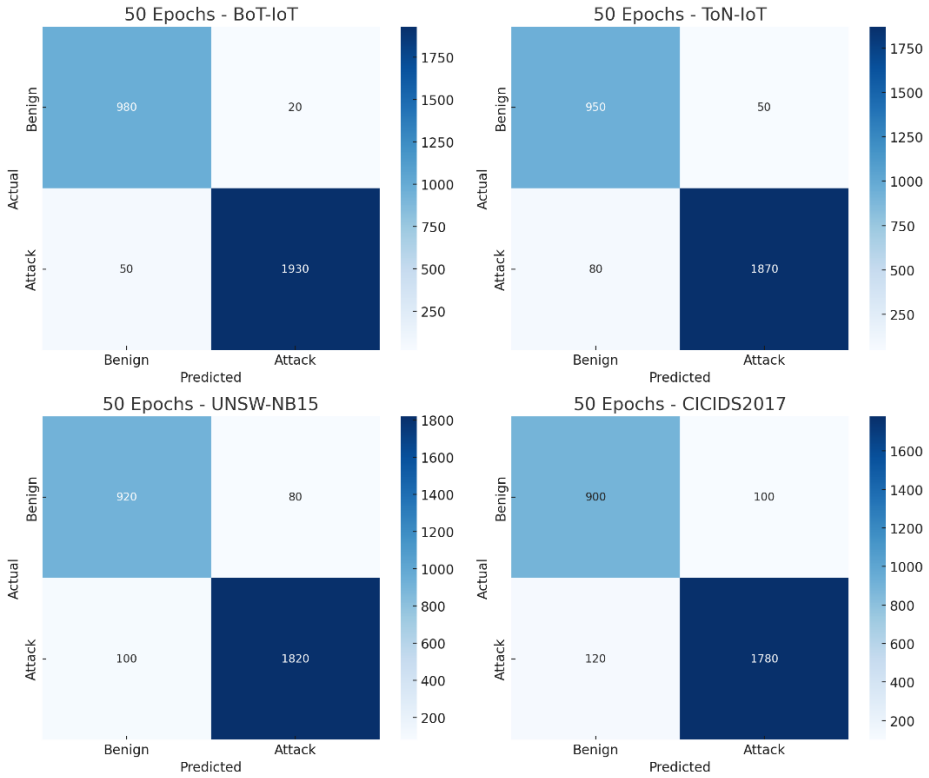


Figure 4. 2. Confusion Matrix of each dataset for 50 epochs

The transition from 5 epochs to 50 epochs reveals clear improvements across all datasets. Early in training, the model lacks stability, leading to high misclassification rates. Over time, learning refiners its decision boundary, improving precision and recall. Notably, BoT-IoT shows the most significant enhancement, while CICIDS20217 retains relatively higher false negatives even after extended training, likely due to the complexity of real-world attack behaviors.

Loss and accuracy curves are fundamental tools for analyzing the training progress of a learning model. These curves help in understanding how well a model is learning from dataset and whether it is underfitting, overfitting, or achieving an optimal balance. The accuracy curve represents the percentage of correctly classified samples during training. As model learns, the accuracy curve should gradually increase and eventually stabilize. If accuracy remains low or fluctuates significantly, it may indicate that the model is struggling to learn patterns from the data. The loss curve represents how well the model's predictions match the actual labels. A high loss value means the model is making poor predictions, while a low loss values

suggests the model has improved its classification performance. During training, loss should decrease as the model refines its internal parameters. However, if loss stabilizes or increases, it might indicate a problem such as overfitting or improper learning.

When observing the curves for 5 epochs, the accuracy starts at a low value and increases gradually, but it does not yet reach a stable level. The loss curve, on the other hand, declines but remains relatively high, indicating that the model is still in the early stages of learning. Since the accuracy is low and the loss is still decreasing, it is clear that 5 epochs are insufficient for the model to have not yet learned enough from the dataset, leading to underfitting, where the model fails to capture essential patterns needed for classification.

When training extends to 50 epochs, the curves exhibit a much more refined learning process. The accuracy curve rises quickly in early epochs and then stabilizes as it approaches a high value, which is consistent with the final accuracy values reported for each dataset. The loss curve shows a significant decline in the first few epochs before flattening, indicating that the model has minimized classification errors. This smooth decline suggests that the model has successfully learned from the dataset, improving its ability to differentiate between benign and attack traffic. At 50 epochs, the training process appears to have reached an optimal balance, where accuracy remains high, and loss remains low without showing signs of overfitting.

The comparison between 5 and 50 epochs reveals a clear improvement in classification performance as training progresses. The loss and accuracy curves for 5 epochs indicate that the model is still in its learning phase, requiring more training to achieve optimal performance. In contrast, at 50 epochs, the curves stabilize, showing that the model has reached a reliable level of accuracy and minimal loss. These observations confirm that while training beyond a certain point might not lead to significant improvements, as the model has already learned most of the patterns in datasets.

# CHAPTER 5

# DISCUSSION

This study demonstrates the practicality and effectiveness of federated learning (FL) in anomaly detection across diverse datasets such as Bot-IoT, TON_IoT, UNSW-NB15, and CICIDS2017. By analyzing the model's performance, several noteworthy outcomes and challenges have emerged. FL ensures data privacy by processing sensitive information locally, minimizing risks associated with centralized storage. Its ability to aggregate insights from heterogeneous data sources proves beneficial for datasets with varying patterns and attack types. The approach achieved strong results, particularly for privacy-critical datasets like Bot-IoT, where maintaining data confidentiality is essential.

Preprocessing techniques, including Short Time Fourier Transform (STFT) and log transformations, enhanced performance by amplifying significant patterns and reducing noise. The scalability of FL was evident, as it maintained high accuracy and F1-Scores across datasets of different complexities. However, handling communication costs and asynchronous training posed challenges, particularly in resource-constrained IoT environments. Additionally, imbalanced datasets such as UNSW-NB15 required careful management to ensure accurate anomaly detection for minority classes.

# CHAPTER 6

# CONCLUSION

This thesis explored the application of federated learning (FL) for anomaly detection across four diverse datasets: Bot-IoT, TON_IoT, UNSW-NB15, and CICIDS2017. The results demonstrated the effectiveness of FL in providing high accuracy, precision, recall, and F1-Scores while preserving data privacy—a critical requirement in modern cybersecurity and IoT systems.

FL's ability to aggregate knowledge from distributed data sources while ensuring privacy is its defining strength. With preprocessing techniques like STFT and feature engineering, the model effectively detected anomalies in datasets characterized by heterogeneous and imbalanced data distributions.

However, challenges such as communication overhead, the requirement for labeled data, and resource limitations in real-time implementations were observed. These findings highlight areas for improvement and future exploration, such as optimizing communication efficiency, handling imbalanced datasets more effectively, and enhancing scalability in dynamic environments.

Overall, this thesis emphasizes federated learning's potential as a scalable and privacy-centric approach to anomaly detection in cybersecurity. By addressing the outlined challenges, FL can become an indispensable tool for protecting complex and distributed systems from emerging threats.

## REFERENCES


[1]Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, “A Survey on Security and Privacy Issues in Internet-of-Things,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[2]K. L. Lueth, “Why It Is Called Internet of Things: Definition, history, Disambiguation”, Internet: https://iot-analytics.com/internet-of-things-definition/, Dec. 19, 2014, [Dec. 24, 2024].

[3]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, pp. 1686–1721, Jan. 2020.

[4]V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," in *IEEE Access*, vol. 7, pp. 82721-82743, Feb. 2019

[5]J. Long, F. Fang, and H. Luo, “A Survey of Machine Learning-based IoT Intrusion Detection Techniques,” in *2021 IEEE 6th International Conference on Smart Cloud (SmartCloud)*, Newark, NJ, USA, 2021, pp. 7-12.

[6]A. I. Awad, “Machine Learning Techniques for Fingerprint Identification: A Short Review,” *Communications in Computer and Information Science*, vol.322, pp. 524–531, Jan. 2012.

[7]M. Anwer, S. M. Khan, M. U. Farooq, and W. Waseemullah, “Attack Detection in IoT using Machine Learning*,” Engineering, Technology & Applied Science Research*, vol. 11, pp. 7273–7278, Jun. 2021.

[8]R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, “Fuzziness based semi-supervised learning approach for intrusion detection system,” *Information Sciences*, vol. 378, pp. 484–497, Feb. 2017.

[9]B. Olanrewaju-George, B. Pranggono, “Federated Learning-based Intrusion Detection System for the Internet of Things using Unsupervised and Supervised Deep Learning Models,” *Cyber Security and Applications*, vol. 3, pp. 100068–100068, Aug. 2024.

[10] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, “A Survey on Federated Learning for Resource-Constrained IoT Devices,” *IEEE Internet of Things Journal,* vol. 9, pp. 1-24, Jan. 2022

[11]N. Khajehali, J. Yan, Y.-W. Chow, and M. Fahmideh, “A Comprehensive Overview of IoT-Based Federated Learning: Focusing on Client Selection Methods,” *Sensors,* vol. 23, p. 7235, Jan. 2023.

[12]A. Khraisat and A. Alazab, “A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges,” *Cybersecurity,* vol. 4, p. 27, Mar. 2021.

[13]A. Verma and V. Ranga, “Machine Learning Based Intrusion Detection Systems for IoT Applications,” *Wireless Personal Communications,* vol.111, pp. 2287-2310, Nov. 2019.

[14]Z. Chen et al., “Machine Learning-Enabled IoT Security: Open Issues and Challenges Under Advanced Persistent Threats,” *ACM Computing Surveys,* vol. 55, pp. 1-55, Apr. 2022.

[15]B. Roy and H. Cheung, “A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network,” in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC),* Sydney, NSW, Australia, 2018, pp. 1-6

[16]P. Mann, N. Tyagi, S. Gautam, and A. Rana, “Classification of Various Types of Attacks in IoT Environment,” in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN),* 2020, pp. 346-350.

[17]S. K. Sahu and K. Mazumdar, “Exploring security threats and solutions Techniques for Internet of Things (IoT): from vulnerabilities to vigilance,” *Frontiers in Artificial Intelligence,* vol. 7, p. 1397480, May 2024.

[18]E. Gyamfi and A. Jurcut, “Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets,” *Sensors,* vol. 22, p. 3744, May 2022.

[19]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, Apr. 2017.

[20]M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, “Intrusion detection systems for IoT-based smart environments: a survey,” *Journal of Cloud Computing,* vol. 7, pp. 1-20, Dec. 2018.

[21]E. Benkhelifa, T. Welsh, and W. Hamouda, “A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and

Resilient Systems," *IEEE Communications Surveys Tutorials*, vol. 20, pp. 3496–3509, Nov. 2018.

[22] M. Z. Mahmud, S. Islam, Shahran Rahman Alve, and A. J. Pial, "Optimized IoT Intrusion Detection using Machine Learning Technique," Internet: https://arxiv.org/abs/2412.02845, Mar. 10, 2024 [Dec. 17, 2024]

[23]M. Zakariah, S. A. AlQahtani, and M. S. Al-Rakhami, "Machine Learning-Based Adaptive Synthetic Sampling Technique for Intrusion Detection," *Applied Sciences*, vol. 13, p. 6504, Jan. 2023.

[24]X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," *IEEE Access*, vol. 7, pp. 82512–82521, Mar. 2019.

[25]A. Oliveira, N. Oliveira, N. Sousa, E. Maia, and I. Praça, "Machine Learning for Network-Based Intrusion Detection Systems: An Analysis of the CIDDS-001 Dataset," *Lecture notes in networks and systems*, vol. 327, pp. 148–158, Oct. 2021.

[26]K. R. Dalal, "Analysing the Role of Supervised and Unsupervised Machine Learning in IoT," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020 pp. 75-79.

[27]D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT," *Journal of Sensor and Actuator Networks*, vol. 12, p. 29, Apr. 2023.

[28]J. Rose, M. Swann, G. Bendiab, S. Shiaeles, and N. Kolokotronis, "Intrusion Detection using Network Traffic Profiling and Machine Learning for IoT," *Journal of Electrical Systems*, vol. 20, pp. 409–415, Jun. 2021.

[29]R.Nowak, "Mathematical Foundations of Machine Learning Mathematical Foundations of Machine Learning." Internet: https://nowak.ece.wisc.edu/MFML.pdf, Aug. 5, 2022 [Jan. 21, 2025].

[30]Z. Li et al., "Data Heterogeneity-Robust Federated Learning via Group Client Selection in Industrial IoT," *IEEE Internet of Things Journal*, vol. 9, pp. 17844–17857, Sep. 2022.

[31]L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, Nov. 2020.

[32]Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous Online Federated Learning for Edge Devices with Non-IID Data," in *2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA, 2020, pp. 15-24.

[33]V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated Learning-based Anomaly Detection for IoT Security Attacks," *IEEE Internet of Things Journal*, vol.9, pp. 2545-2554, Feb. 2022.

[34]J. Mills, J. Hu, and G. Min, "Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT," *IEEE Internet of Things Journal*, vol.7, pp. 5986–5994, Jul. 2020.

[35]J. Xu, L. Su, and P. Yang, "Towards a mathematical foundation of federated learning: a statistical perspective." presented at the IISA Conference, Colarado, USA, 2023

[36]T. Li, A. K. Sahu, Ameet Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol.37, pp. 50-60, May. 2020.

[37]E. M. Campos et al., "Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges," *Computer Networks*, vol. 203, p. 108661, Feb. 2022.

[38]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, Nov. 2019.

[39]Nickilaos Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards Developing Network forensic mechanism for Botnet Activities in the IoT based on Machine Learning Techniques," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 5, pp. 30–44, 2018

[40]N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," *Future Generation Computer Systems*, vol. 110, pp. 91–106, Sep. 2020.

[41]Nickolaos Koroniotis and N. Moustafa, "Enhancing Network Forensics with Particle Swarm and Deep Learning: The Particle Deep Framework," *Journal of Scientific & Industrial Research*, vol. 82, pp. 522-528, Mar. 2020.

[42]N. Koroniotis et al., "A Holistic Review of Cybersecurity and Reliability Perspectives in Smart Airports," *IEEE Access*, vol. 8, pp. 209802-209834, Mar. 2020.

[43]Cyber Range Lab of UNSW Canberra, "Bot_IoT," Internet: https://www.kaggle.com/datasets/vigneshvenkateswaran/bot-iot, Feb. 5, 2024 [Aug. 8, 2024]

[44]A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-

driven Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 165130-165150, Feb. 2020.

[45]N. Moustafa, M. Ahmed, and S. Ahmed, "Data Analytics-enabled Intrusion Detection: Evaluations of ToN_IoT Linux Datasets," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou, China, 2020, pp. 727-735

[46]N. Moustafa, "New Generations of Internet of Things Datasets for Cybersecurity Applications based Machine Learning: TON_IoT Datasets," *IEEE Access*, vol. 8, pp. 165130 – 165150, Jan. 2020

[47]T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Data Sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, Jan. 2022.

[48]Cyber Range Lab of UNSW Canberra "The TON_IoT Datasets | UNSW Research," Internet: https://research.unsw.edu.au/projects/toniot-datasets, Mar. 5, 2022 [Aug. 8, 2024]

[49]N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1-6

[50]M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-based Network Intrusion Detection Systems," *Big Data Technologies and Applications*, vol. 371, pp. 117–135, Dec. 2021.

[51]N. Moustafa, J. Slay, and G. Creech, "Novel Geometric Area Analysis Technique for Anomaly Detection using Trapezoidal Area Estimation on Large-Scale Networks," *IEEE Transactions on Big Data*, vol.5, pp. 481-494, Dec. 2019.

[52]Cyber Range Lab of UNSW Canberra "The UNSW-NB15 Dataset | UNSW Research,"Internet: https://research.unsw.edu.au/projects/unsw-nb15-dataset ,Jan. 15, 2020 [Aug. 8, 2024]

[53]Canadian Institute for Cybersecurity, "IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB," Internet: https://www.unb.ca/cic/datasets/ids-2017.html,Mar. 26, 2017 [Aug. 8, 2024]

[54]Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani "Papers with Code - CICIDS2017 Benchmark (Network Intrusion Detection)," Internet: https://paperswithcode.com/sota/network-intrusion-detection-on-cicids2017,Jun. 13, 2017 [Aug. 8, 2024]

[55]B. Olanrewaju-George, B.Pranggono, “Federated Learning-based Intrusion Detection System for the Internet of Things using Unsupervised and Supervised Deep Learning Models,” *Cyber Security and Applications*, vol. 3, pp. 100068–100068, Aug. 2024.

[56]P. Dubey and M. Kumar, “Integrating Explainable AI with Federated Learning for Next-Generation IoT: A comprehensive review and prospective insights,” *Computer Science Review*, vol. 56, p. 100697, May 2025.

[57]L. K. G. Danquah, S. Y. Appiah, V. A. Mantey, I. Danlard, and E. K. Akowuah, “Computationally Efficient Deep Federated Learning with Optimized Feature Selection for IoT Botnet Attack Detection,” *Intelligent Systems with Applications*, vol. 25, p. 200462, Nov. 2024.

[58]Y. Hou, R. He, J. Dong, Y. Yang, and W. Ma, “IoT Anomaly Detection Based on Autoencoder and Bayesian Gaussian Mixture Model,” *Electronics*, vol. 11, p. 3287, Jan. 2022.

[59]M. E. Mahmoud, M. Kasem, A. Abdallah, and H.-K. Kang, “AE-LSTM: Autoencoder with LSTM-Based Intrusion Detection in IoT,” presented at *2022 International Telecommunications Conference*, ITC, Egypt, Jul. 2022.

[60]B. Weinger, J. Kim, A. Sim, M. Nakashima, N. Moustafa, and K. J. Wu, “Enhancing IoT anomaly detection performance for federated learning,” in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, Tokyo, Japan, 2020, pp. 206-213.

[61]V. Dutta, M. Pawlicki, R. Kozik, and M. Choraś, “Unsupervised network traffic anomaly detection with deep autoencoders,” *Logic Journal of the IGPL*, vol.30, pp. 912-925, Dec. 2022.

[62]E. Tsogbaatar et al., “DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT,” *Internet of Things*, vol. 14, p. 100391, Jun. 2021.

[63]V. V. Timcenko and Slavko Gajin, “Machine Learning based Network Anomaly Detection for IoT environments,” in *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, Ravet IN, India, 2023, pp. 1-6