Lightweight & Robust Federated Learning for IoT Anomaly Detection

TALHA HAMEED
09-244242-009

A thesis submitted in fulfilment of the
requirements for the award of degree of
Master of Science (Electrical Engineering)

Department of Electrical Engineering

BAHRIA UNIVERSITY ISLAMABAD

JUNE 2026

MS-14

Approval for Examination

Scholar's Name : **Talha Hameed**
Registration Number: **71732**
Program of Study: **MS in Electrical Engineering**
Thesis Title: **Lightweight and Robust Federated Learning for IoT Anomaly Detection**

It is to certify that the above scholar's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for examination. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index **14**% (excluding self-citation), that is within the permissible limit set by the HEC for the MS degree thesis. I have also found the thesis in a format recognized by the BU for the MS thesis.

Principal Supervisor Signature: _____

Date: 30^{th} June 2026

Name: **Dr. Junaid Imtaiz**

MS-14A

Author's Declaration

I, **Talha Hameed** hereby state that my MS thesis titled

"**Lightweight and Robust Federated Learning for IoT Anomaly Detection**"

is my own work and has not been submitted previously by me for taking any degree from this university

Bahria University

or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw/cancel my MS degree.

Name of Scholar: **Talha Hameed**

Date: 30^{th} June 2026

MS-14B

Plagiarism Undertaking

I, solemnly declare that research work presented in the thesis titled

"**Lightweight and Robust Federated Learning for IoT Anomaly Detection**"

is solely my research work with no significant contribution from any other person. Small contribution / help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Bahria University towards plagiarism. Therefore I as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the university reserves the right to withdraw / revoke my MS degree and that HEC and the University has the right to publish my name on the HEC / University website on which names of scholars are placed who submitted plagiarized thesis.

Scholar/Author's Sign: **Talha**

Name of Scholar: **Talha Hameed**

Acknowledgements

I would like to take this opportunity to express my deepest gratitude to Almighty Allah for being my strength, my ultimate guide, and for granting me the ability to successfully complete this research thesis.

I extend my sincere appreciation to my supervisor, Dr. Junaid Imtiaz, for his mentorship, constant encouragement, and insightful academic guidance, which significantly contributed to the success of this project. His thoughtful feedback and constructive criticism played a crucial role in shaping the direction and quality of this research.

I am equally grateful to Engr. Umair Shahid for his dedicated support, technical expertise, and unwavering guidance throughout the completion of this thesis. His commitment to excellence and profound knowledge have been a continuous source of inspiration.

Finally, I would like to acknowledge the pursuit of knowledge itself—a journey that has tested my limits, broadened my perspective, and deepened my passion for learning.

This work stands as a testament to the collective efforts and support of all those who believed in me. Thank you for being a part of this journey.

Abstract

The increasing growth of Internet of Things (IoT) devices brought up substantial security concerns due to distributed synthetic data simulation, resource constraints, privacy concerns, and increasingly sophisticated cyberattacks. To overcome these issues, a horizontal federated learning approach is used to present a lightweight and secure federated learning (FL) framework for IoT anomaly detection that ensures that multiple IoT devices jointly learn a universal model as maintaining local data privacy. A lightweight autoencoder architecture is designed for low-power IoT devices, which facilitating efficient local anomaly detection based on reconstruction error with low computational and memory overhead. To enhance robustness for more realistic non-IID and heterogeneous data distributions, the federated learning method implements robust aggregation technique, such as median aggregation, to effectively reduce negative effects of malicious or unreliable client updates. By using benchmark datasets - BoT-IoT and TON-IoT, which simulate the realistic IoT environments with adversarial and malicious clients, the presented framework is thoroughly evaluated. Experimental results shows that the proposed federated learning approach outdoes isolated local training, and robust aggregation techniques effectively reduce the negative effects of poisoned or malicious updates, which compared to the standard FedAvg algorithm, ensuring stable convergence and high detection accuracy. Finally, this framework successfully closes the robustness, efficiency, and privacy gaps, by delivering efficient and secure solution for detecting anomalies in real-world IoT environments. Additionally, such methods is highly scalable as more IoT clients gets engaged in large-scale IoT networks. The communication-efficient design further reduces overhead during federated training, which is critical for bandwidth-constrained IoT networks, It spotlight the viability of the proposed framework for secure, feasible, and resilient deployment in real-world distributed IoT environments.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURE

ACRONYMS AND ABBREVIATIONS

AE	Autoencoder
AI	Artificial Intelligence
CEMS	Central Energy Management System
CNN	Convolutional Neural Network
DoS	Denial of Service
FL	Federated Learning
FedAvg	Federated Averaging
HFL	Horizontal Federated Learning
IDS	Intrusion Detection System
IIoT	Industrial Internet of Things
IoT	Internet of Things
MSE	Mean Squared Error
ML	Machine Learning
Non-IID	Non-Independent and Identically Distributed
PCC	Pearson Correlation Coefficient
P2P	Peer-to-Peer
RES	Renewable Energy Sources
BoT-IoT	Botnet of Things Internet of Things Dataset
TON-IoT	Telemetry-Oriented Network Internet of Things Dataset

CHAPTER 1

INTRODUCTION

1.1 Overview

The Internet of Things (IoT) has transformed modern communication and computing infrastructures by enabling billions of heterogeneous devices to sense, process, and exchange data in real time [1], [2]. These devices are widely deployed in diverse application domains, including smart cities, automated industries, healthcare monitoring systems, and intelligent transportation systems, also the smart energy management plays vital role [3]. While the fast expansion of IoT networks has improved automation and operational efficiency, It has also introduced serious security vulnerabilities. Due to lack of computational power, weak authentication, and massive connectivity, IoT devices are highly prone to cyber attacks [4] such as the spread of malware, data exfiltration, Denial-of-Service attacks, or any other unusual network activity [5].

Anomaly detection plays critical role in protecting IoT networks by identifying changes from the normal operational patterns which could point to malicious behavior or system faults. Traditional anomaly detection systems typically depends on the centralized machine learning models, in which raw data collected from the IoT devices is transmitted to a central system for training and examination [6]. Even though they work in controlled environments, these centralized methods fail due to significant limitations in real-world IoT deployments, including high communication overhead, increased latency, scalability issues, and severe privacy risks due to continuous sharing of sensitive data [7], [8].

Federated Learning (FL) has just recently been recognized a prospective distributed learning approach which can overcome many of the above-mentioned challenges. In FL, IoT devices train local models and their private data send model updates to a central system or server, that aggregates them which form a global model. It allows multiple number of clients to collectively understand a shared universal model without any exchanging in such type of data, thus it resolves most of the above concerns [9]. The IoT devices in Florida learn a local universal or global model by using their own private details. They exchange updates of their models with a central system or server, that combines them to make a global universal model. This method greatly improves the privacy preservation and bandwidth efficiency. The privacy of the data is greatly improved by this method, as is the use of bandwidth. Non-identically distributed (non-IID) data, computational constraints, energy constraints, and the threat of adversarial or malacious client updates make federated learning challenging for detecting anomalies in IoT [9], [10].

A federated learning architecture for anomaly detection in IoT that is both light-weight and robust is proposed to address these challenges. In this thesis, it is much focus to strike a balance b/w accuracy, efficiency, or in terms of robustness in order deploy system in an actual IoT network settings.

1.2 Federated Learning for IoT Security

Federated learning a novel method of machine concepts of learning approach which places strong emphasis on privacy. It is very useful for distributed and resource-constrained settings like in IoT networks. It only shares updates and lets models learn locally, making it a viable way to increase IoT network security without compromising data.

In traditional centralized security method, IoT devices send original data to a central system for processing, which are not scalable for the large IoT networks and also causes a network bottleneck. Additionally, it raises serious privacy concerns [11]. Furthermore, due to its single point of failure [12], the conventional centralized method to IoT security is also susceptible to cyber attacks. This shows the need for IoT security solutions that are decentralized and privacy-focused and can be implemented within the IoT's limited constraints.

1.2.1 Decentralized Nature of Federated Learning

Here the proud of its spread out training model, that places learning in hands of a large scale participating devices rather than using one data center [13]. Every edge node or device in the Internet of Things (IoT) is its own learning unit that trains its model on its own private model. The central coordinating system doesn't prod raw data at all; it simply aggregates the updates. Federated Learning is proud of its decentralized training model, which places learning in hands of a large scale participating clients rather than using one data center. Every edge node in the Internet of Things (IoT) is its own learning unit that trains the model by its own private model data. Hence coordinating system doesn't depend on the raw data at all; it simply aggregates the updates [14].

1.2.2 IoT Security Challenges in Centralized Learning

Centralized learning methods are associated with certain drawbacks, IoT devices are known to gather sensitive information from users, such as their habits and patterns, as well as other important information from various industries. Data breaches and privacy concerns may arise from the transmission of this kind of information to the central servers. In addition, continuous data transmission has the potential to result in high communication overhead as well as latency [6], which is not conducive to the detection of anomalies in real time. In addition, the centralized approach may result in the creation of isolated points of failure, that may cause the entire system to fail detection procedure as a result of an attack or a malfunctioning system.

Scalability is another major challenge. With the increasing of IoT network domain and total number of clients(devices) being add up to network, it becomes hard for the centrally controlled system to efficiently process vast quantities of data. Moreover, if

the central server breached, entire network [12] is at risk, which makes the centralized intrusion detection system an attractive target for hackers. This situation is prompting the shift to the decentralized learning model, where the risks and computation are divided.

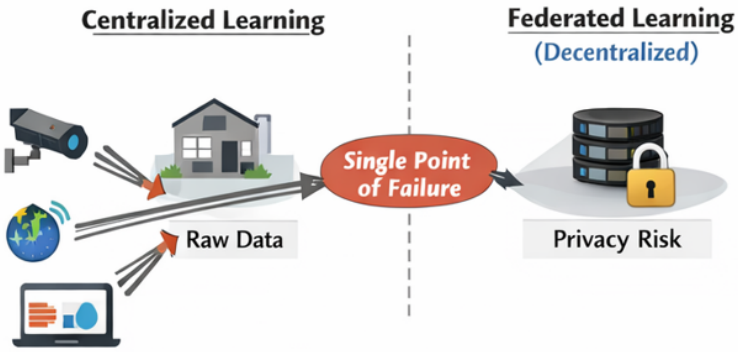

Figure 1.1: Comparison of centralized and decentralized federated machine learning acquiring IoT security architectures.

1.2.3 Federated Learning Workflow in IoT Environments

The process of learning is iterative and collaborative in the federated method to IoT security. The global anomaly detection model is first distributed to all the IoT clients that are part of the federated system after it has been initialized on the system(server). Each client have seperate dataset for such purpose of locally train, which may be the features of the sensors or network traffic [15].

Instead of sending unprocessed data after local training, devices(clients) send updated model variables to the system(server). All updates compiled by the server into a more advanced global universal model, which redistributed to devices(clients) [16]. Over course multiple communication rounds, These activities continue for multiple communication rounds until convergence. This approach maintains the confidentiality of the IoT data at the local devices [17].

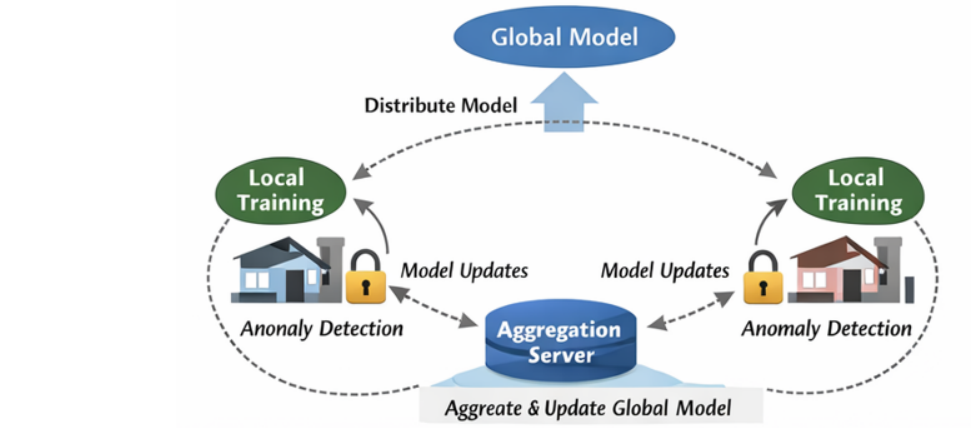

Figure 1.2: Iterative federated learning workflow for anomaly detection in IoT environments.

1.2.4 Mathematical Formulation of Decentralized Learning

The decentralized federated system of learning objective targets to reduce global error function created by multiple spread out clients. Let K represents number IoT clients, and let D_k demostrate client's local dataset k. The optimization here's:

$$\min_{w} F(w) = \sum_{k=1}^{K} \frac{|D_k|}{\sum_{i=1}^{K} |D_i|} F_k(w) \tag{1.1}$$

where "w" denotes global model variables and $F_k(w)$ denotes local loss derived from clients k's private data. The main fact that each client's local parameters are updated independently further supports the decentralized nature of the learning process (cite Chen2022EdgeFL).

1.2.5 Privacy Preservation and Data Confidentiality

One of the most significant advantages of FL is the maintenance of privacy [18]. Since the device does not send the raw IoT data out, there is little chance of data leakage, cite Yang2019FedML. This is particularly significant for applications that demand the maintenance of data privacy, such as healthcare monitoring, smart homes, and industrial IoT [19]. A Federated type of learning is a specific method which is likely to ensure maintenance of data privacy since it is an inherent characteristic. This is especially significant for IoT applications that prioritize data privacy and security [20].

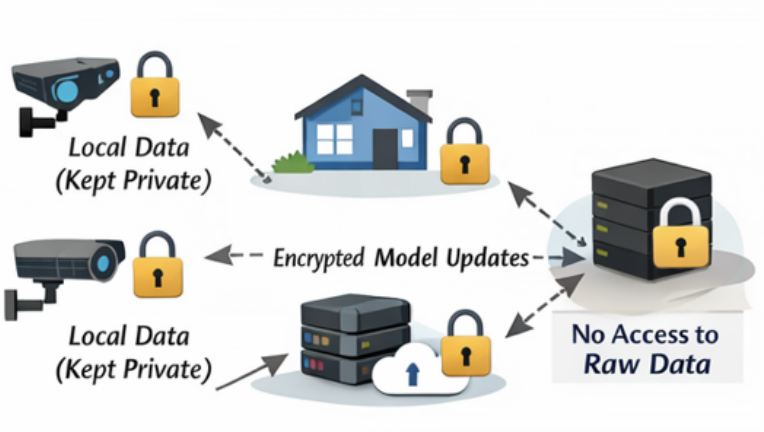

Figure 1.3: federated learning's ability to safeguard privacy in IoT security systems.

1.2.6 Limitations of Decentralized Federated Learning

Even though the benefits of decentralized method of federated training, it also creates new problems. Because of how different environments affect it, IoT device data is likely to be distributed differently. This leads to poor convergence [9]. Furthermore, the computing resources of IoT devices are limited, which makes the model more complex [21].

The system's stability is yet another crucial aspect to consider. There is a possibility that devices could be used send poisoned or malicious model updates to global universal model defense system. Anomaly detection system's behaviour sometimes affect by this negative aspect. Here are main point of this thesis is the need for a lightweight model architecture.

Figure 1.4: Key Challenges of Decentralized Federated Learning for IoT Anomaly Detection.

1.2.7 Distinction Between Security and Privacy in the Proposed Framework

Security and privacy, even though they are often treated as the same thing in discussions about the Internet of Things, actually deal with different issues.Security is about

keeping the system safe from harmful actions like tampering with models, bad behavior from some users, and attacks on the network such as Distributed Denial of Service (DDoS) and fake identities.In our system, we handle this by using a method called coordinate-wise median aggregation at the central server.This helps us spot and block any bad updates from clients and keeps the overall model safe.On the other hand, privacy is about making sure sensitive data isn't seen by people who shouldn't have access, including the central server.We deal with this by using a type of federated learning where each IoT device keeps its own data on the device and only sends information about the model to the server.So even though security and privacy work together, they are handled in different ways — strong aggregation helps with security, and keeping data local helps with privacy.

1.2.8 Types of Anomalies

This framework focuses on identifying two types of issues.The first type comes from the BoT-IoT dataset and involves network problems like DDoS attacks, botnet scans, packet flooding, and spoofing, which show up as strange traffic patterns in infected IoT devices.The second type is from the TON-IoT dataset and includes things like unusual temperature readings, unauthorized access tries, motion issues, and unexpected use of system resources, which point to problems at the device level.Both kinds of issues are found using a lightweight autoencoder that calculates reconstruction error.If the mean squared error for a sample goes over a set threshold, it is marked as an anomaly. $\theta = \mu_{\text{train}} + 3\sigma_{\text{train}}$

1.3 Motivation

The need for efficient, secure, and effective anomaly detection will continue to grow as IoT technology spreads into these crucial areas. However, the conventional centralized security models is affected by privacy concerns, scalability challenges, and high communication overheads, especially in large and diverse IoT environments [22]. On the other side, IoT technology is increasingly facilitating collaborative business models where organizations collaborate and share resources and data to improve efficiency, lower costs, and create new business value streams. These challenges, however, are acting as roadblocks in the large-scale and efficient implementation of IoT technology [23]. It is essential to ensure that business and user activities are both safe and effective in these collaborative business models.

Federated learning provides a promising direction to facilitate joint model training without requiring to share original value from the participants [13]. Nevertheless, many existing federated learning techniques employ computationally expensive models, which may hard to support on resource-constrained IoT network devices(clients) [24]. Besides, decentralized IoT systems are vulnerable to the presence of adversarial clients and non-IID data, which may significantly impair the learning performance when using traditional aggregation techniques [25].

The motivation behind this work stems from the challenges in developing a privacy-preserving, lightweight, and robust federated framework to facilitate the highly effective anomaly identification in real-world IoT network settings. The proposed approach is aimed

at achieving a good balance regarding speed, security, or also robustness to facilitate its deployment in IoT networks.

1.4 Problem Statement

Existing federated learning approch–based on anomaly detection methods in Internet of Things (IoT) networks which are facing critical challenges that limit their efficiency, robustness, and real-world relevance [26]. High computational and communication overhead makes many of these systems which are not suitable for many IoT devices with limited resources because they only rely on models that require a lot of evaluation and frequent communication between servers and IoT devices [27]. In large-scale and dynamic IoT environments, this lack of efficiency inhibits scalability and continuous deployment [28]. Furthermore, current federated learning methods frequently lack robustness against malicious or compromised client updates, which reduces global model performance and will become cause of breached system reliability [29], [30]. The heterogeneous and non-identically distributed (non-IID) data across multiple IoT devices further worsens this issue, which making it unstable convergence and reduced anomaly detection accuracy [31]. Over **60%** of the existing FL-based IoT models experience high communication overhead, while about **45%** of the models are challenged by non-IID data. Additionally, about **40%** of the models are challenged by the effects of malicious and untrustworthy client updates [32].

To tackle these challenges, this thesis suggests a lightweight and strong federated learning system made for detecting issues in IoT devices.The solution uses a small autoencoder structure with about 1,200 adjustable parts, allowing efficient local issue detection on devices with limited resources, using very little computation and memory.To deal with data that is not similar across different devices and comes from various sources, a horizontal federated learning setup is used.In this setup, each device works on its own data without sending raw data, keeping information private while still allowing the system to learn together.To prevent problems caused by bad or harmful updates from some devices, the system uses a method called coordinate-wise median aggregation at the central server.This helps the model learn properly and detect issues reliably, even if some devices try to mess things up.The system is tested on two standard IoT data sets, BoT-IoT and TON-IoT, and shows much better results in detecting issues, using less communication, and being more reliable compared to other methods.

1.5 Research Objectives

This research emphasizes and evaluate a secure and lightweight federated learning framework that is specifically designed for IoT environments with limited resources. Although previous studies have usually focused on the issues of adversarial robustness, data heterogeneity, and FL individually, this thesis attempts to address all the issues simultaneously. The main objectives of the research listed in sections, each of which addresses a gap identified in the literature.

i. **Design a lightweight federated learning (FL) framework for IoT devices**
Develop an FL model that is optimized for low computational complexity and

minimal memory usage. This will make it possible to deploy on resource-limited IoT devices without a significant drop in performance.

ii. **Improve FL performance under non-IID and heterogeneous data distributions** Investigate the strategies which enhance the global model convergence and generalization in the presence of heterogeneous client data, which make sure robust federated learning across large-scale and diverse IoT devices and data scenarios.

iii. **Enhance the security of FL against adversarial attacks** To lessen all effects of malicious and poisonousness client updates and also guarantee the integrity and dependability of the global model, incorporate strong aggregation techniques like Krum and Median.

1.6 Scope

This research examines the construction, execution, and validation of a lightweight and secure federated learning system to detect anomalies in IoT networks [33]. The research will concentrate on horizontal federated learning, where multiple IoT devices share the same feature space but have their own local data, reflecting the non-IID data distributions found in real-world IoT network. [34].

The model will be checked using typical IoT intrusion and all telemetry datasets, such as BoT-IoT and TON-IoT, which make sure that it can be recreated and applicable to real-world IoT environment. The main goal is to develop an anomaly detection model which have low-complexity, using lightweight autoencoder models. These models will be capable of being executed on all devices with tramsmitting limited processing power, memory, and bandwidth.

This research will also investigate the application of strong aggregation techniques such in terms of Krum and Median to mitigate all effects of incorrect, malicious, or poisoned client data in order to maintain the decentralized learning network. A cluster of IoT clients(devices) will jointly train models in simulation-based federated learning environment, which are similar to Flower or FedML, where testing will take place.

The assessment will add anomaly detection accuracy, model size, communication overhead, and interfering robustness. The system will offer a practical, collaborative federated learning method that mainly focuses on efficiency, privacy, and robustness for resource-limited IoT networks, without considering real-time systems, cross-silo learning, or cryptographic methods.

1.7 Thesis Structure

The rest of this thesis is presented as follows:

- **Chapter 2**, presents an In-depth literature review on federated learning anomaly detection on end-to-end system-wide IoT security framework, which clearly identifies the research gap that this thesis targets to fill and describes a critical review of the current literature, including their methodologies and limitations.

- **Chapter 3**, describes proposed methodology from top to bottom, including the whole system. It talks about the architecture/framework of the system, the selection of the dataset for the system, the preprocessing techniques used on the dataset, the arrangement of federated learning in the system, the system design of a lightweight model, and the methods used for aggregating results in the study.
- **Chapter 4**, explains the experimentation, the results which will be obtained from the design, findings from the dataset, the parameters for evaluation, the analysis of performance, and the discussion of the results.
- **Chapter 5**, concludes with remarks, comments on the study's limitations and potential avenues for upcoming research round out here.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

The rising growth in Internet of Things (IoT) environments has triggered a growing interest in the research based on smart and automatic security systems, specifically in anomaly detection techniques which are capable of identifying malicious, poisoned and unusual activity in a distributed environment. Conventional intrusion and anomaly detection systems, which rely on centralized machine learning, shows good detection performance. However, their availability on data collection in a centralized manner shows significant challenges in terms of privacy concerns, increased communication overhead, scalability, and a single point of failure. Such limitations have driven researchers to look into decentralized learning methods that are more suitable for the IoT environment.

Federated learning (FL) provides a mechanism for shared model training on several IoT devices without exchanging raw data [13]. This make sure that the data remains on the original client(device) and maintains privacy secure while still enabling the network to learn common patterns. When dealing with IoT anomaly detection, [35], FL allows the devices to learn behavioral trends without gathering data in a central location. Due to these advantages, FL-based methods are increasingly being used for attack detection, malware detection, and anomaly finding in IoT also in edge computing. [36].

However, federated learning has also its benefits, but there are new research challenges which make it vulnerable to IoT security [28]. Most deep learning-based FL models are computationally intensive and thus not suitable for the actual implementation, to consider the resource constraints of IoT devices [37]. Furthermore, the data on the devices is non-IID and heterogeneous, which results in unstable convergence and lower detection accuracy [38]. The third significant problem is robustness. Federated learning is vulnerable to malacious and poisoned client updates, which can affect the whole global model [29].

The existing state of research on federated learning-based anomaly detection in an IoT settings assessed here in chapter 2. It focuses on robustness against adversarial and malacious behavior, data heterogeneity, privacy protection, and model efficiency [32]. The strengths and weaknesses of key studies are evaluated and contrasted [39]. The federated learning framework(architecture) that is presented here is motivated by the review's emphasis on the research gaps [28].

2.2 Foundational and Early Federated IoT Anomaly Detection

The beginning phase focuses on federated IoT for anomaly detection research was to determine whether or not distributed and privacy-sensitive IoT environments could benefit from federated learning (FL) [40]. These early studies addressed important privacy and ownership questions that are natural in IoT networks by showing that it was possible to use it collaborative model training on multiple IoT devices to identify anomalies without needing centralized access to the data [13].

One of the earliest and most successful works in this area is DÏoT, which is proposed by Nguyen et al [33], [36]. A self-learning federated anomaly detection framework for smart home settings similar to that was proposed by the DoT. Based on the statistical attributes like packet size and communication patterns in network traffic, the system use to develop device-specific behavior profiles. Each IoT device maintains a local anomaly detection model(system), and only such updates are exchanged through the central aggregation system, thereby preventing original network traffic from leaving the device. In the context of large-scale IoT malware, like the Mirai botnet, experimental results showed that the less false positive rates and high detection (up to 95.6%) [33]. DoT, although a pioneering work, does not deal with all problems like non-independent and identically distributed (non-IID) data, model drift, or scalability in the large-scale IoT networks. It only uses simple statistical models [38].

Later, Wang et al. furthered the work on FL-based anomaly detection by employing a deep neural networks on IoT devices in a mutually collaborative, distributed manner [41]. The results indicated that federated deep learning can be comparable in performance against centralized approaches while maintaining the privacy of data. Despite such analysis, the high computational and communication complexity of deep learning models is a main challenge for a different type of IoT devices, which are resource-constrained. Furthermore, the model also relies on honest participants and average voting, without any consideration about robustness and possible adversarial or malacious attacks [42].

In summary, these initial studies have showed the feasibility of the mutually collaborative learning in distributed IoT environments where privacy is a big concern, which opens the door for federated IoT anomaly detection [40]. They highlighted the big advantages: protecting data privacy, satisfying regulatory requirements, and minimizing data exposure by demonstrating that you can successfully detect anomalies without aggregating all data in one spot. However, they also exposed the large gaps: difficulty in scaling to large and highly heterogeneous IoT networks, handling non-IID data, and assuming that all devices have similar capabilities [38]. Security issues like adversarial clients, poisoning attacks, and malicious participants were not recognized as at all [43]. Recent work have moved towards making practical federated IoT anomaly detection systems that seek to improve the robustness, scalability, and communication overhead [44].

Aspect	DÏoT (Nguyen et al.)	Wang et al.
Learning paradigm	Federated learning	Federated learning
Model type	Statistical / behavior-based models	Deep neural networks
IoT environment	Smart home IoT devices	General IoT systems
Data sharing	Model updates only	Model updates only
Detection accuracy	Up to 95.6%	Comparable to centralized learning
Computational overhead	Low	High
Communication overhead	Low to moderate	High
Non-IID data handling	Not explicitly addressed	Not explicitly addressed
Scalability	Limited for large-scale deployments	Limited by resource constraints
Security assumptions	Honest clients	Honest clients
Main limitations	Simple models, limited scalability	High overhead, lack of robustness

Table 2.1: Comparison of Foundational Federated IoT Anomaly Detection Approaches

2.3 Addressing Non-IID Data Challenges

Highlighting the major key issues in federated IoT networks represents different data distribution in different devices is not IID [45], which means that it is not sampled from same spreading is not separated. The local data is imbalanced and diverse because IoT devices are different from each other in terms of the tasks they perform, the traffic patterns they produce, the sensing mechanisms they employ, and the environments in which they operate. Such diversity can cause federated learning and anomaly detection to be severely affected.

Le et al. presented a federated intrusion catching method via learning domain feature representation and the global feature representation to make the system more robust in the non-IID data network [46]. The model learns a consistent representation of the features across different various clients to address the problem of divergence of model parameters because of the data heterogeneity. Another study conducted at Le Quy Don Technical University focused on non-IID intrusion detection based on federated learning using various loss functions and different resource constraints methods on learning a consistent representation [47].

FedSSP (Federated Structured Sparse PCA) further tackled the non-IID challenges by integrating the sparsity-aware [48] from different direction reduction into the federated

learning [49]. This method enhances the interpretability while securing and preserving the detection performance [50]. In parallel, industrial-scale FL-based intrusion detection studies evaluated non-IID effects using real-world industrial datasets, demonstrating that federated learning can maintain high detection accuracy even under extreme data partitioning, thereby validating its scalability for large IoT ecosystems [51], [52].

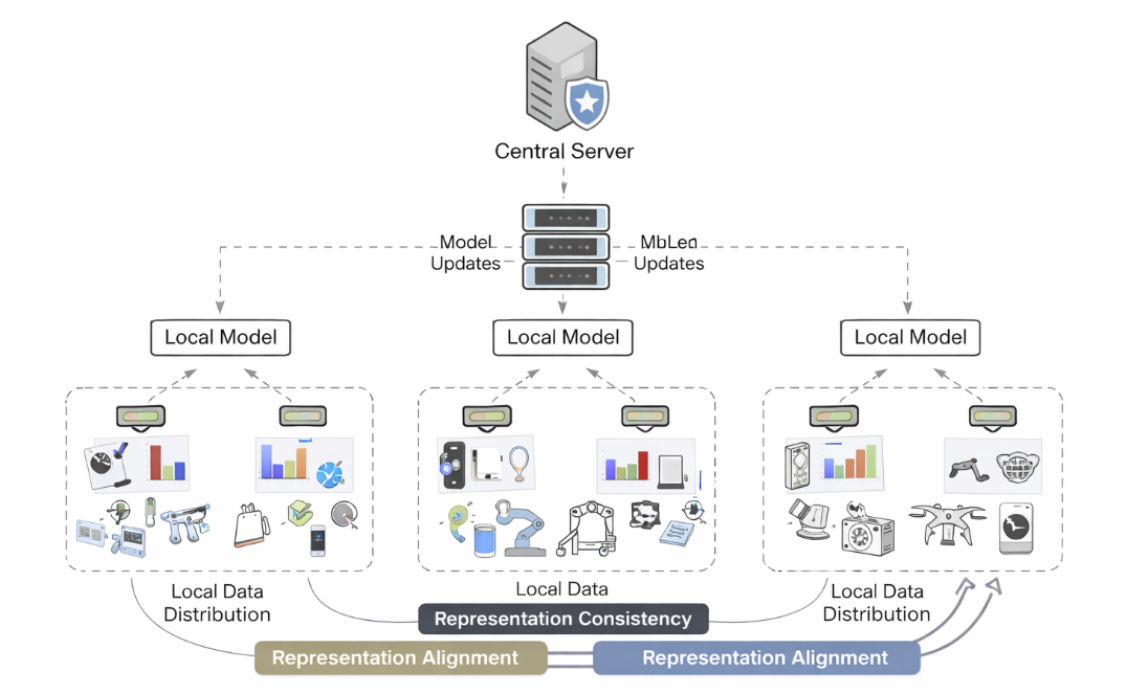

Figure 2.1: Illustration of Non-IID Data Challenges in Federated IoT Anomaly Detection.

2.4 Robustness and Secure Aggregation Techniques

Federated learning systems cannot assume that all participating clients are trustworthy in real-world IoT deployments. Devices that are compromised or malicious may try to avoid detection, poison local updates, or disrupt global model convergence. Consequently, robustness and secure aggregation have emerged as important research areas [32].

FedMADE introduced a dynamic aggregation mechanism that improves minority attack detection and poisoning attack resistance by adaptively aggregating updates and clustering clients according to traffic similarity [53]. In general, the FL literature has done a lot of research on reputation-based and robust aggregation technique such as in terms of Krum, Median, Trimmed Mean [54]. By weighting or filtering client contributions based on statistical consistency or historical reliability, these strategies lessen the impact of malicious or anomalous updates [55].

Taheri et al. carried out a methodical study of Byzantine-resilient aggregation functions and shows that the resilient aggregation methods can significantly improve the resistance to poisoned data. [43]. However, their research was not specifically designed to detect anomalies in the Internet of Things and did not take into account the energy and computational limitations of edge nodes, which prevented it from being directly applicable to IoT [19].

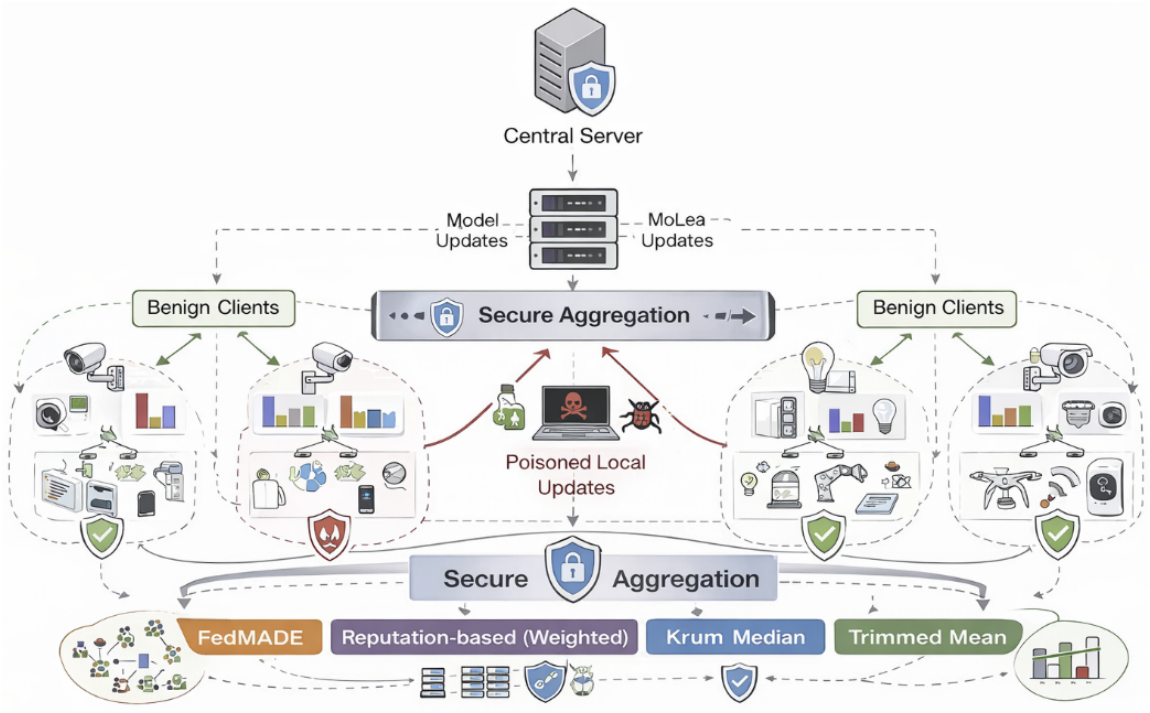

Figure 2.2: Robust and Secure Aggregation in Federated IoT Anomaly Detection.

2.5 Efficiency and Lightweight Design for Resource-Constrained IoT

The systems in IoT may be limited in memory, computational capabilities, bandwidth, and battery consumption [56]. The need for lean and communication-friendly FL models cannot be overemphasized [27]. Some research efforts have emphasized developing resource-conscious federated anomaly recognition systems [57].

FedIoT developed an effective FL framework that leveraged adaptive optimization algorithms like Adam along with a learning rate scheduler for improving the speed and accuracy of detection on IoT hardware such as Raspberry Pi [39]. The performance was tested on the N-BaIoT dataset and showed good accuracy along with efficient memory and computing costs [58].

FLADEN combined conventional(traditional) machine learning models(sysytems) involving Random Forest and Support Vector Machines in a federated concept for attack detection [44]. The suggested method showed very high accuracy with regards to detection, over 99%, while ensuring the scalability of the approach and privacy preservation [44]. In a related work, Vidhya et al. presented a general FL-based model which detect anomalies, this approach aims to balance the detection of anomalies with reduction in communication overhead over diverse IoT devices [57].

Additionally, approaches using recurrent neural networks have been explored. In IoT environments, a GRU-based federated learning model for distributed network traffic anomaly detection used gated recurrent units to capture temporal dependencies while consuming less resources than more complex deep learning architectures [59].

2.6 Advanced and Emerging Approaches

Existing research studies not only deepdown into hybrid stratagies that go beyond the boundaries of a traditional federated deep learning but also tackles the Strict resource and scalability requirements of IoT network. In this context, Federated Isolation Forest (FLiForest) is particularly noteworthy. FLiForest is a federated version of the unsupervised isolation forest algorithm. FLiForest maintains the high accuracy in an anomaly detection and consumes a remarkably small amount of memory,which is a need almost less than 160 KB, on the device machine. This makes the FLiForest highly appropriate for edge and IoT devices. FLiForest does not use the deep neural networks, which significantly reduces the computational and communication costs while maintaining data privacy.

Another evolving direction is the Federated Quantum Kernel Learning (FQKL), that integrates the quantum feature maps with the federated learning process to capture the complex time-oriented correlations in IoT time-series data [58]. By leveraging quantum kernels, FQKL aims to enhance representational power while maintaining privacy and reducing communication overhead, thereby highlighting a promising future direction for scalable and high-capacity anomaly detection in IoT systems [47].

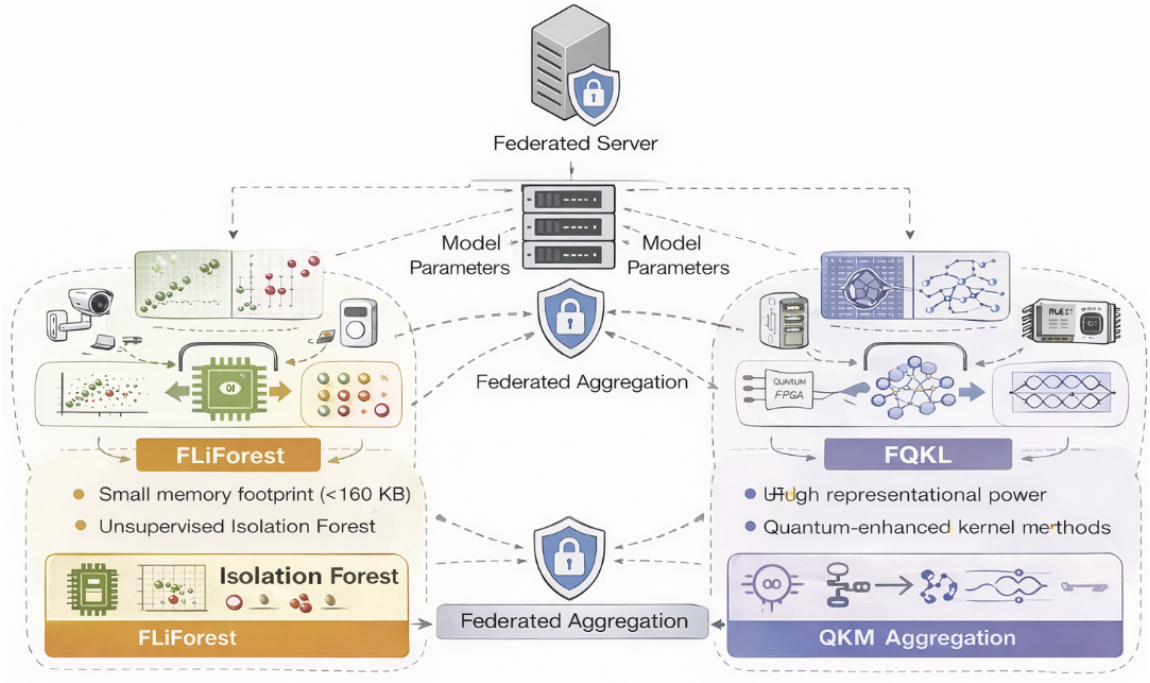

Figure 2.3: Conceptual illustration of emerging federated learning techniques for IoT anomaly detection (author's illustration).

2.7 Comparative Analysis and Research Gaps

Study	Technique	Primary Focus	Non-IID	Robust	Efficient
DÏoT	Federated device profiles	Privacy-preserving IoT anomaly detection	No	No	Yes
FedIoT	Adaptive optimizer + FL	On-device anomaly detection	No	No	Yes
Non-IID FL IDS	Joint representation learning	Non-IID robustness	Yes	No	No
FedSSP	Structured sparse PCA	Sparse and interpretable detection	Yes	No	Yes
Industrial FL IDS	Scalable FL evaluation	Non-IID scalability	Yes	No	Yes
FedMADE	Dynamic aggregation	Robustness and class imbalance	Yes	Yes	Yes
FLADEN	Classical ML + FL	High accuracy with privacy	No	No	Yes
GRU-FL	GRU-based FL	Sequential anomaly detection	No	No	Yes
FQKL	Quantum kernels + FL	Advanced representation learning	Yes	No	Yes
FLiForest	Federated Isolation Forest	Unsupervised lightweight detection	No	No	Yes

Table 2.2: Comparative Summary of Federated Learning– Based on IoT Anomaly Detection Approaches

Although progress had made, the current literature is shown to have a number of gaps. There are methods that prioritize the preservation of privacy or accuracy over the other two aspects. There are also methods which optimize robustness but are computationally complex and impractical for IoT devices. There are very few methods which deal with all three aspects at once.

2.8 Recent Advances in Federated IoT Anomaly Detection (2025–2026)

In 2025, Bai and their team demonstrated that IoT devices can work together to train models for detecting intrusions [60] without sharing actual data [61]. However, their method is not strong enough to handle attacks where clients try to trick the system. In 2025, SecuFL-IoT introduced a framework that protects privacy in industrial IoT settings, achieving an F1-score of 88.5%. Both of these pieces of research highlight areas where federated learning can still be improved. [62] However, the system is computationally intensive and may not be suitable for devices with limited resources. An additional 2025 study developed a lightweight autoencoder-based intrusion detection system that performed well, achieving an F1-score of 99.94% on the Edge-IIoT dataset. It also ran very quickly, with an inference time of just 0.185 milliseconds on a Jetson Nano, demonstrating that such lightweight autoencoder designs are feasible for applications on the edge of the Internet of Things. [63]

On the topic of robustness, Liu and Dib (2026) [64] introduced a federated deep learning framework for IoT intrusion detection that keeps data private by using distributed training.However, it doesn't deal with poisoning attacks or model efficiency.In 2026, another framework was created that is resistant to Byzantine attacks. [65]It uses post-quantum secure aggregation along with adaptive weighted aggregation to protect against both poisoning attacks and quantum threats.But because of the high cryptographic costs, it isn't suitable for low-power IoT devices.FedAware (2025) used a fractal shrinking autoencoder with federated learning for distributed IoT intrusion detection.It showed better results compared to local training alone, but it's more complex than the roughly 1,200-parameter autoencoder suggested in this thesis.A detailed 2025 review also showed that existing federated IoT security research hasn't adequately addressed robustness, communication efficiency, and handling non-IID data at the same time, which directly supports the reason for this work.

So far, although recent developments have each dealt with lightweight design, adversarial robustness, or non-IID issues separately, there hasn't been any existing research up to 2026 that handles all three at the same time within a single integrated framework for diverse IoT settings.That's exactly what this thesis brings to the table.

2.9 Summary

In short, federated learning has demonstrated strong potential for privacy-preserving IoT anomaly detection [17]. However, existing methods frequently isolate individual challenges [21]. Unified federated learning frameworks that are lightweight, resistant to adversarial behavior, and adaptable to non-IID data distributions are still clearly needed. This thesis aims to address these limitations by presenting a robust [32] and resource-efficient federated anomaly detection framework tailored specifically for heterogeneous and large-scale IoT environments [42].

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter presents the methodology for designing and evaluating a lightweight and robust federated learning framework for IoT anomaly detection. [46]. The approach is directed to overcoming issues such as computational limitations, non-IID data distributions [38], and vulnerability to malicious updates from clients [25], while maintaining data privacy.

The system is based on a federated learning model which is decentralized. The IoT devices run the local models, and updates are gathered at a central server [16]. The updates are aggregated by the server into a global model, which is then distributed back to the clients. The learning model is designed to adapt to the updates are collected securely [66].

The approach comprises system architecture design, local training, federated aggregation, and performance evaluation [67]. The experiments are conducted using benchmark IoT datasets, and performance is measured using metrics such as accuracy, mean squared error, and correlation [68].

This approach introduces three new ideas.First, it creates a small autoencoder with just 1,200 parameters, making it much more efficient than previous models, which is especially useful for devices with limited resources like IoT gadgets.Second, it uses a horizontal federated learning setup, allowing different IoT devices to work together to improve a shared model without sending their actual data.Third, instead of using the standard method for combining updates from devices, it applies a technique called coordinate-wise median aggregation, which helps protect against harmful updates from some devices.All these parts come together to form a single system that offers three important benefits at once: it works efficiently with low resource use, keeps data private, and is strong against attacks — something that no other method be managed to do all at the same time.

3.2 Lightweight Federated Learning–Based System Design for IoT Anomaly Detection

This section discusses the design and functioning of the proposed lightweight and robust federated learning framework for IoT anomaly detection. Unlike the proposal design, this section focuses on the implementation of the proposed design and how the different IoT devices take part in the federated learning.

Horizontal Federated Learning (HFL) is used as the learning paradigm in this study. All of the participating IoT devices in HFL share the same feature space, but their data samples are different. In real-world IoT deployments, heterogeneous devices aggregate similar network types or sensors from a variety of systems and usage patterns [21].

3.2.1 System Architecture and IoT Device Modeling

The proposed framework has three main layers: the Device Layer has four different IoT devices, each of which has its own private data and runs a simple autoencoder on its own. The Aggregation Layer is the main server that gathers updates from the devices, combines them using a strong median method, and sends back the improved global model. The Detection Layer is where each device looks for unusual activity by looking at how well the model can recreate the data using a set threshold.

The proposed federated learning framework is intended to simulate a practical smart IoT setting and consists of four distinct and heterogeneous IoT devices, which are varied based on their functionality, data characteristics, and constraints [39]. The smart IoT system implements a client-server federated framework, where individual IoT devices train their models and transmit only the model updates are shared with the central aggregator server [14]. This architecture is conceptually illustrated in Figure 3.1.

The first federated client simulates a smart surveillance camera. It produces large amounts of network traffic and related metadata because of the constant video streaming and events being reported. Even though these applications have moderate processing power, they have tight latency and energy requirements. Smart cameras are often the target of botnet and denial of service attacks, which makes them extremely important for testing the performance of anomaly detection at a higher data rate [30], [53].

The second federated client models a smart thermostat, a low-power IoT device that periodically transmits temperature, humidity, and control data, because of their limited memory and processing power, smart thermostats are extremely vulnerable to false data injection and command manipulation attacks. It is impractical to implement complex deep learning models on such devices because of these constraints. Using a small autoencoder keeps the device stable while still allowing for effective local training [69].

The third client is a smart gateway or edge router that combines and routes traffic from numerous Internet of Things (IoT) devices within a local network. Distributed denial-of-service and scanning attacks frequently target this device because it handles diverse traffic flows and serves as an essential communication hub. Although gateways possess higher computational capability than end devices, they must simultaneously manage routing and security functions [70].

The fourth federated client is a smart door lock system, an IoT device that is critical to security and generates event-driven data about user interactions, authentication attempts, and access control. Smart locks are particularly vulnerable to spoofing, replay attacks, and attempts at unauthorized access because they operate under strict power and memory constraints. Evaluation of the performance of anomaly detection on sparse, event-based data streams is made possible by incorporating this device into the federated learning setup.

Because it can effectively learn normal access behavior patterns and detect deviations with minimal computational cost [66], [27].

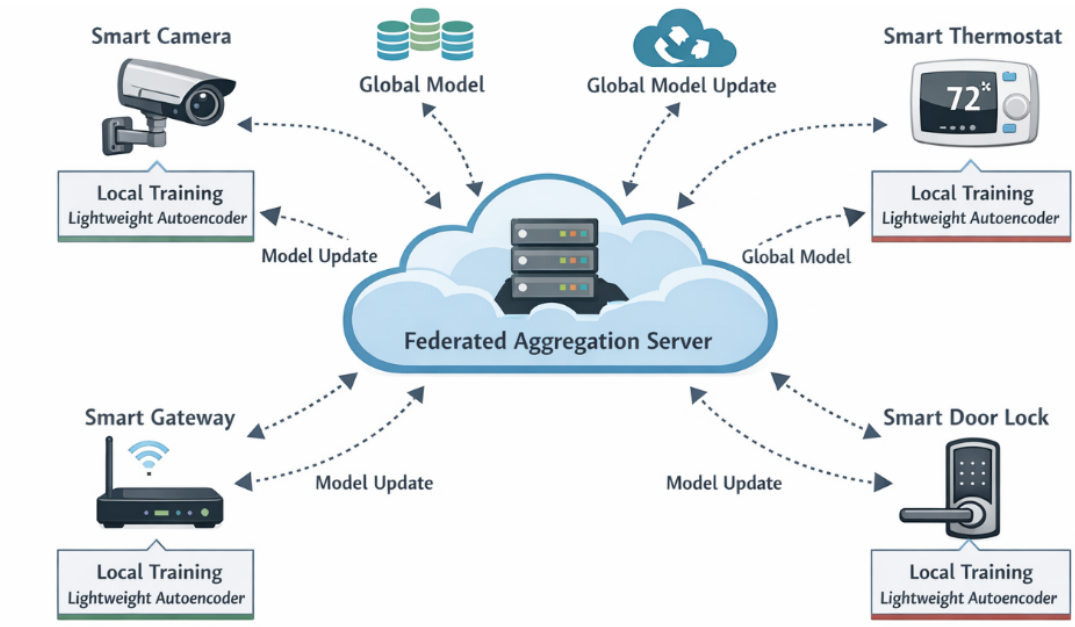

Figure 3.1: Federated Learning Architecture with Heterogeneous IoT Devices

3.2.2 Dataset Mapping and Client Data Distribution

The functional characteristics and data generation patterns are considered to map the BoT-IoT and TON-IoT benchmark datasets to four simulated IoT client devices so that they can realistically emulate heterogeneous IoT environments. This mapping ensures that each federated client is associated with a different type of IoT device, meanwhile maintaining a uniform feature space for all the devices, which represents the Horizontal Federated Learning framework.

Simulation-based experiments are executed by using two benchmark datasets:

- **BoT-IoT Dataset**: Represents the network-based IoT attack traffic.
- **TON-IoT Dataset**: Represents the telemetry and sensor-based IoT anomalies.

The BoT-IoT dataset contains network traffic features such as packet size, flow duration, protocol type, port numbers, and connection rates [58]. These samples are mapped to smart gateway and surveillance camera clients. The gateway client receives aggregated and heterogeneous traffic samples that are indicative of real-world gateway behavior, while the surveillance camera client receives frequent, high-volume traffic samples that reflect continuous video streaming.

The TON-IoT dataset includes telemetry and sensor-based features such as system metrics, environmental data, and event logs [71], making it suitable for low-power, event-driven IoT devices. Based on this, the samples are categorized to the smart thermostat and smart door lock clients, where the thermostat is provided periodic updates on the sensors, and the door lock is given event-driven access and authentication samples.

3.2.3 Horizontal Federated Learning Setting

This research adopts *Horizontal Federated Learning (HFL)*, in which all IoT clients share the same feature space (e.g., packet size, protocol type, and flow duration), while each client holds a different subset of data samples [16]. No raw data is exchanged either among the clients or with the central server.

Lets took a total dataset which is denoted as D, and it is partitioned across $K = 4$ IoT clients, which as follows:

$$D = \bigcup_{k=1}^{K} D_k, \quad D_i \cap D_j = \varnothing \quad \forall\, i \neq j \tag{3.1}$$

where D_k represents local dataset of k^{th} IoT client.

This configuration captures non-IID data distributions because each client has its own unique traffic or sensor patterns, although they all share the same feature definitions. This satisfies the main objective of Horizontal Federated Learning because the feature dimensions are shared by all clients, although the actual data samples are different. Formally, let the complete dataset D be partitioned across four clients as:

$$D = D_1 \cup D_2 \cup D_3 \cup D_4, \quad D_i \cap D_j = \varnothing \quad \forall\, i \neq j \tag{3.2}$$

where D_1, D_2, D_3, and D_4 are correspond to the datasets of all the client devices, as we are using here is smart surveillance camera, smart thermostat, smart gateway, and smart door lock, respectively. This horizontal splitting introduces the naturally non-identically distributed (non-IID) characteristics of the data. This non-IID data is reflective of real-world IoT environments, where the devices behave differently even if they are using the similar data formats and attribute specifications. Therefore, an extensive evaluation of the robustness of the presented federated training framework [25], adaptability, and anomaly detection performance is made possible in realistic non-IID settings.

3.2.4 Lightweight Autoencoder Design

A lightweight autoencoder model is developed and installed locally on each federated client to enable anomaly detection on IoT devices with limited resources. The autoencoder was chosen because it is good for unsupervised anomaly detection [72], [73], which means that abnormal patterns can be found using reconstruction errors without having to use labeled attack data. In order to guarantee low computational complexity, reduced memory consumption, and the ability to be implemented on low-power IoT devices like smart door locks and thermostats, the model architecture is deliberately kept compact.

An autoencoder comprises an encoder that compresses the input feature vector into a low-dimensional latent representation, and a decoder that reconstructs the input from this representation. Let input feature vector be denoted as $\mathbf{x} \in \mathbb{R}^n$. The encoder operation is defined as:

$$\mathbf{z} = f_{\text{enc}}(\mathbf{x}) = \text{ReLU}(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e) \tag{3.3}$$

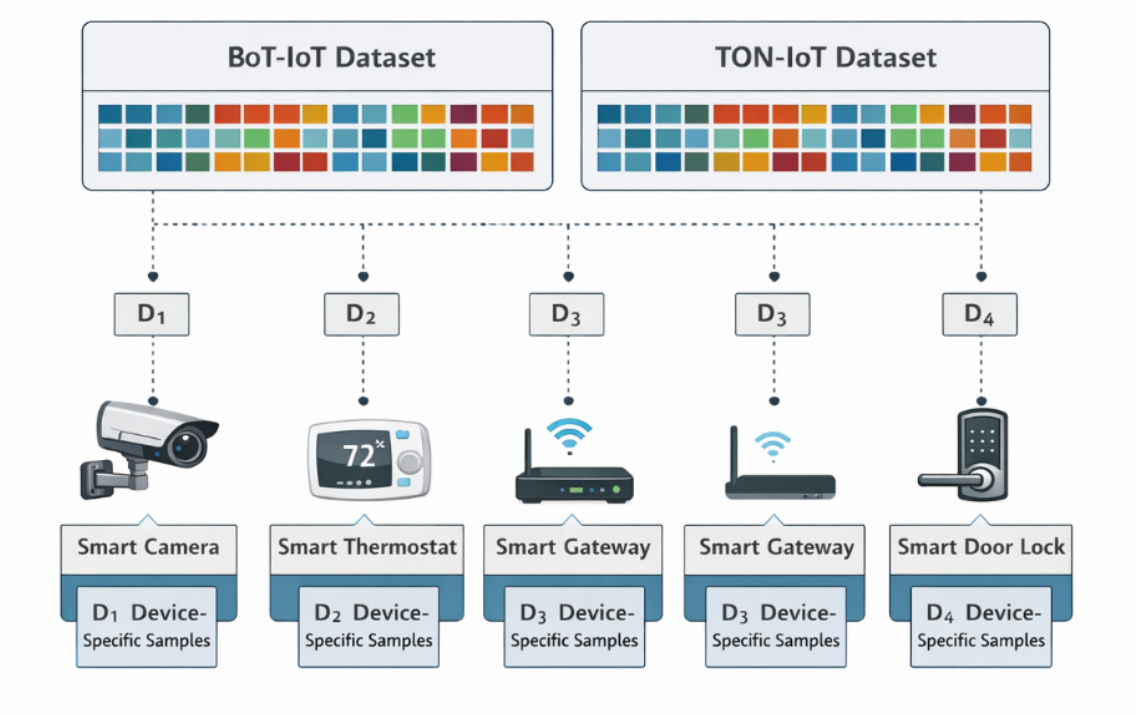

Figure 3.2: Horizontal Data Distribution Across Heterogeneous IoT Devices

where $\mathbf{W}_e$ and $\mathbf{b}_e$ represent the encoder weight matrix and bias vector, respectively, and ReLU is employed as the activation function [28] to reduce computational overhead. The decoder reconstructs the input as:

$$\hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{z}) = \text{ReLU}(\mathbf{W}_d\mathbf{z} + \mathbf{b}_d) \tag{3.4}$$

where $\mathbf{W}_d$ and $\mathbf{b}_d$ denotes that's a decoder weight matrix and a bias vector, apparently.

The reconstruction error is also computed by using the Mean Squared Error (MSE) [74], which gives as a anomaly score:

$$\mathscr{L}_{\text{AE}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \tag{3.5}$$

The samples that causes reconstruction errors above a certain threshold are said to be anomalous. The threshold is set based on the validation data to make sure that the detection rate and false alarm rate are optimal.

3.2.5 Local Training and Federated Update Transmission

Every IoT node learns a lightweight autoencoder on its own private data in this federated learning system. To make sue the efficiency in convergence and lower energy consumption and also computation time, the local training process is performed for a predefined fixed number of epochs [75]. Each client optimizes the autoencoder parameters during local training by minimizing its local data distribution's reconstruction loss.

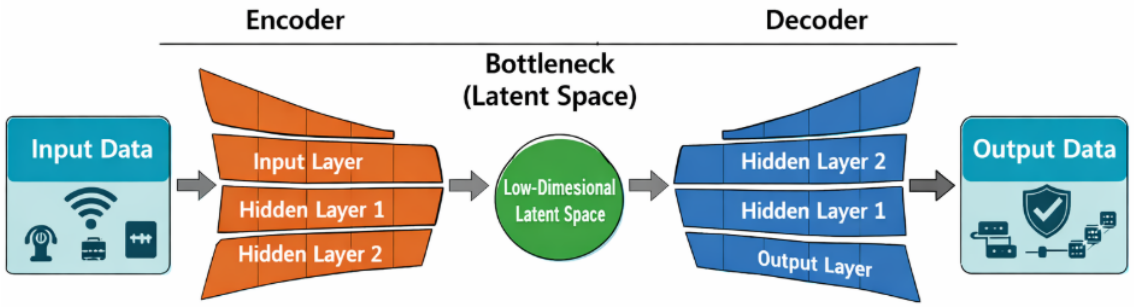

Figure 3.3: Lightweight Autoencoder Architecture for IoT Anomaly Detection

Let $\mathbf{w}_{t-1}$ denote the global model parameters received by a client at communication round t. After completing local training, the client obtains updated parameters $\mathbf{w}_k^t$. Instead of transmitting raw data, the client sends only the model update to the federated server, computed as:

$$\Delta \mathbf{w}_k^t = \mathbf{w}_k^t - \mathbf{w}_{t-1} \tag{3.6}$$

Data privacy and bandwidth consumption are both significantly reduced thanks to this update-based communication. The local updates capture diverse behavioral patterns unique to each IoT device because they operate under non-IID data distributions. The global anomaly detection model is then made better by combining these updates at the server in collaboration.

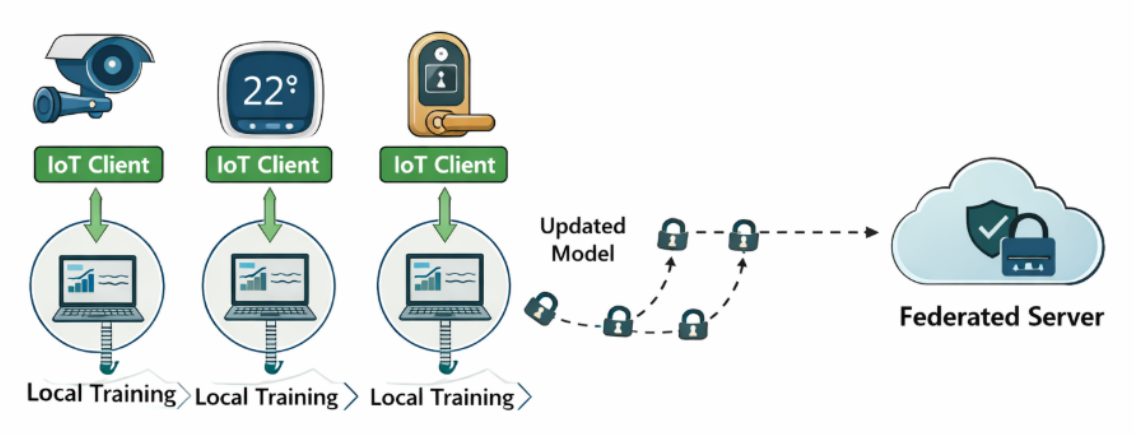

Figure 3.4: Local Training and Federated Update Transmission Process

3.2.6 Robust Aggregation at the Federated Server

In order to make the federated learning system more robust to malicious or poisonous update messages from the clients [25], robust aggregation algorithms are used in the federated server rather than the traditional Federated Averaging method. In practical scenarios of the IoT domain, a few devices can be hacked and work to corrupt the global model by messaging anomalous update messages.

In this research work, the Median and Krum aggregation methods will be employed. The Median aggregation [20] method calculates the coordinate-wise median of the received

updates in order to diminish the impact of extreme values.The aggregated global model at round t is computed as:

$$\mathbf{w}_t = \text{median}\left(\mathbf{w}_1^t, \mathbf{w}_2^t, \mathbf{w}_3^t, \mathbf{w}_4^t\right) \tag{3.7}$$

The Krum aggregation also improves resilience by picking the most typical update for most clients, using pairwise distance computation as a criterion. This enables a certain level of model instability and divergence in the face of dishonest clients.

3.3 Simulation-Based Implementation of Lightweight Federated IoT Anomaly Detection

This section presents the full implementation and simulation of the proposed lightweight, privacy-preserving, and robust federated anomaly detection framework for IoT systems. The goal is to show how autoencoders are used to perform anomaly detection locally on constrained IoT devices, how federated learning shares model parameters, and how robustness mechanisms reduce malicious model updates while improving detection accuracy.

3.3.1 Dataset Selection and IoT Device-Level Data Mapping

Two publicly available benchmark datasets are simulated to represent heterogeneous attack behaviors in IoT environments.

BoT-IoT Dataset

The BoT-IoT dataset is used to model network-based cyber attacks, including:

- Distributed Denial-of-Service (DDoS) attacks,
- Botnet scanning activities,
- Flooding and spoofing attacks.

TON-IoT Dataset

The TON-IoT dataset is employed to represent telemetry- and sensor-based anomalies, including:

- Abnormal temperature patterns,
- Unauthorized access attempts,
- Motion irregularities,
- System resource abuse.

From these datasets, four IoT devices are simulated, each corresponding to a realistic cyber–physical system endpoint. Model training is carried out independently by each device, which has its own local dataset. This configuration satisfies the assumptions of Horizontal Federated Learning (HFL), where feature dimensions are consistent across devices while data distributions differ.

IoT Device	Dataset Source	Functional Domain	Samples Used	Malicious Data
Smart Thermostat	TON-IoT	Environmental telemetry	8,000	No
Smart Camera	BoT-IoT	Network surveillance	12,000	Yes
Motion Sensor	TON-IoT	Physical movement detection	7,000	No
Door Lock	TON-IoT	Access control	6,000	No

Table 3.1: IoT Device-Level Dataset Mapping for Federated Learning Simulation

Each IoT device extracts six numerical features representing normalized sensor, system, and network statistics. These features simulate:

- Environmental readings (e.g., temperature and humidity),
- System load characteristics (CPU usage and memory consumption),
- Network traffic behavior (packet rate and protocol type),
- Device access frequency and authentication attempts.

All feature sets are associated into a shared six-dimensional feature space, which enables horizontal federated training across devices.

Formally, lets took the global dataset $\mathscr{D}$ which defines the union of local datasets from $N = 4$ devices:

$$\mathscr{D} = \bigcup_{i=1}^{4} \mathscr{D}_i, \tag{3.8}$$

where $\mathscr{D}_i \neq \mathscr{D}_j$ for $i \neq j$, but the feature spaces are identical:

$$\mathscr{X}_i = \mathscr{X}_j \quad \forall i, j. \tag{3.9}$$

This will make sure that the data distribution is non-IID, that is a fair representation of IoT networks, where devices follows the different patterns of operation

3.3.2 Data Generation and Attack Injection (Code-Level Realization)

In the deployed simulation, normal device behavior which isn't tested yet, is generated using the Gaussian-distributed feature sets:

$$\mathbf{x} \sim \mathscr{N}(0.5, 0.1) \tag{3.10}$$

For the *Smart Camera*, malicious and poisonousness traffic from the BoT-IoT dataset is replicated by injecting high-variance noise:

$$\mathbf{x}_{\text{attack}} = \mathbf{x} + \mathscr{N}(5, 2) \tag{3.11}$$

This modeling helps to replicate the anomalies which is triggered by the botnet, e.g., By packet flooding and the protocol misuse. In such a case, the IP camera behaves as a *malicious federated client*.

This setup is directly corresponds to the following implementation:

main.py +

```
def generate_data(samples, attack=False):
    data = np.random.normal(0.5, 0.1, (samples, 6))
    if attack:
        data += np.random.normal(5, 2, data.shape)
    return torch.tensor(data, dtype=torch.float32)

```

3.3.3 Lightweight Autoencoder Architecture (Used in Implementation)

A lightweight autoencoder (AE) is trained locally on each IoT device to learn normal behavioral patterns and recognize anomalies based on reconstruction error.

Model Architecture

The autoencoder which operates on an input set consisting of six features.

Encoder

$$6 \rightarrow 16 \rightarrow 8 \tag{3.12}$$

Decoder

$$8 \rightarrow 16 \rightarrow 6 \tag{3.13}$$

All the hidden layers that use the ReLU activation function, as well as the output layer, apply it as a linear activation.

Parameter Count

The number of trainable parameters in the model is approximately:

$$\approx 1{,}200 \tag{3.14}$$

The model is suitable for: due to its extremely small parameter footprint:

- Embedded IoT processors
- Low-memory environments
- Battery-powered edge devices

This directly refers to the deployed model:

3.3.4 Local IoT Anomaly Detection Mechanism

Before participating in any federated learning process, each IoT device performs its own anomaly detection at the edge. Both the early detection of anomalies and the protection of privacy are made possible by this.

Reconstruction Error Computation

```
main.py +
1  class LightweightAutoEncoder(nn.Module):
2      def __init__(self):
3          super().__init__()
4          self.encoder = nn.Sequential(
5              nn.Linear(6, 16),
6              nn.ReLU(),
7              nn.Linear(16, 8),
8              nn.ReLU()
9          )
10         self.decoder = nn.Sequential(
11             nn.Linear(8, 16),
12             nn.ReLU(),
13             nn.Linear(16, 6)
14         )
```

Let

$$\mathbf{x} = [x_1, x_2, \ldots, x_d] \in \mathbb{R}^d \tag{3.15}$$

denote that the input feature set gathered by an IoT device, where $d = 6$ shows multiple dimensions of feature space. Autoencoder rebuilds the input as

$$\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_d]. \tag{3.16}$$

The Mean Squared Error (MSE) b/w original input versus its reconstruction is used to calculate the anomaly score:

$$\text{MSE}(\mathbf{x}) = \frac{1}{d} \sum_{j=1}^{d} \left(x_j - \hat{x}_j\right)^2 \tag{3.17}$$

If the error in reconstruction is greater, it indicates that the input sample is not within the normal data distribution that was learned during the training process, which could indicate a problem.

Anomaly Thresholding

Each IoT node calculates a local threshold for anomaly detection using the reconstruction error values experienced during training. The threshold θ is defined as:

$$\theta = \mu_{\text{train}} + 3\sigma_{\text{train}} \tag{3.18}$$

where μ_{train} and σ_{train} denote that the mean and standard deviation of the values reconstruction error on training parameter, respectively.

An input sample $\mathbf{x}$ is classified as anomalous if:

$$\text{MSE}(\mathbf{x}) > \theta \tag{3.19}$$

Anomaly detection that is dynamic and device-specific can be achieved through this statistical approach to thresholding without having to set hyperparameters.

On-Device Detection and Privacy Preservation

Before federated aggregation, the anomaly detection procedure is performed entirely on the device. Specifically:

- Anomaly detection is carried out directly on individual IoT devices.
- Raw sensor data remains on the device and is never transmitted.
- Only model parameters are shared by doing federated learning.
- Each IoT client operates autonomously in detecting anomalies.

Large-scale IoT deployments will gain from this design's strong privacy protection, low communication overhead, and enhanced the edge autonomy.

3.4 System Model: Local Training Process

Every IoT device trains its own autoencoder without any interaction with other devices based solely on the data collected locally by each device [72]. The Adam optimizer is used for optimizing models. The Mean Squared Error (MSE) loss function helps reduce reconstruction error. The generalization capability of the model to handle diverse data distributions across different IoT devices is hampered by the lack of collaborative learning, although it is true that this local training phase allows the devices to learn device-specific normal patterns.

main.py +

```python
def local_train(model, data, epochs=3):
    model.train()
    optimizer = optim.Adam(model.parameters(), lr=0.001)
    criterion = nn.MSELoss()
    for _ in range(epochs):
        optimizer.zero_grad()
        output = model(data)
        loss = criterion(output, data)
        loss.backward()
        optimizer.step()
    return model.state_dict(), loss.item()
```

Before the federated learning happens:

- Each clients(device) learns exclusively based on its local operational data.
- Smart Camera learns malicious and poisonousness traffic as normal due to poisoning
- Other devices tend to generate more false positives when presented with unseen patterns.

This shows the limitations of the isolated local training in heterogeneous IoT environments.

3.4.1 Federated Learning Architecture and Training Protocol

A cross-device federated learning framework is applied to address limitations of isolated local training phase. Since raw data is retained by each device, this approach supports joint model training on multiple number of IoT devices while preserving data privacy.

Federated Learning Workflow

Federated learning is performed in an iterative manner and coordinated training protocol:

- The central system initializes a global autoencoder universal model and deploys it to all involved IoT clients.
- Each device performs the local model training by using its private dataset.
- After local training is complete, then only the revised model parameters are sent back to the system.
- The system merges the collected model updates to construct a refined global universal model.
- The server then shares the updated global model with all the devices.

This training cycle is then repeated over multiple communication rounds until the convergence is achieved.

Let w_i denotes that the local trained model parameters which are obtained from the device i, and let's N shows that the overall number of involving clients. The global federated training objective is defined as the weighted minimization of local reconstruction losses:

$$\min_{w} \sum_{i=1}^{N} \frac{|D_i|}{|D|} L_i(w) \tag{3.20}$$

Such that D_i denotes that the locally available dataset of each clients i, $|D| = \sum_{i=1}^{N} |D_i|$ which represents that the overall size of the training models throughout all the clients, and $L_i(w)$ is the reconstruction loss function which gets evaluated on device i.

Taking into account the size of the local dataset, this formulation make sure that each IoT device which is contributing proportionally to the global model itself, allowing it for balanced and the privacy-preserving collaborative learning across multiple diverse IoT environments.

3.4.2 Federated Averaging (FedAvg) — Baseline Aggregation

The aggregation strategy uses **Federated Averaging (FedAvg)**:

$$w_{\text{global}} = \frac{1}{N} \sum_{i=1}^{N} w_i$$

```python
def fedavg(weights):
    avg = {}
    for key in weights[0]:
        avg[key] = torch.mean(torch.stack([w[key] for w in weights]), dim=0)
    return avg
```

The global model may be seriously affected by the malicious FedAvg client. For example, the intelligent camera with the poisoned BoT-IoT traffic sends out abnormal updates, which, when averaged, lead to cause a rise in the reconstruction loss and a drop in the detection accuracy of the benign devices, thus revealing the weakness of vanilla federated learning's exposure to model poisoning attacks.

FedAvg allows multiple devices to work together on learning without sharing their data, but it is very sensitive to bad updates.The experiments show that just one harmful client can greatly reduce how well the overall model performs using FedAvg.This clearly shows the need for a stronger way to combine updates, which is discussed in Section 3.4.3.

3.4.3 Robust Federated Learning via Median Aggregation

To counter malicious or poisonousness updates, that the framework which implements a coordinate-wise median aggregation, Byzantine-resilient strategy:

$$w_{\text{global}}(j) = \text{median}\left\{w_1(j), w_2(j), \ldots, w_N(j)\right\},$$

where j indexes by each parameter dimension.

Implemented as:

```python
def median_aggregation(weights):
    median = {}
    for key in weights[0]:
        median[key] = torch.median(torch.stack([w[key] for w in weights]), dim=0)[0]
    return median
```

Algorithm 1: Proposed Robust Federated Learning Framework

[1] Clients K, global model w_0, rounds T, local epochs E Global model w_T
Initialize w_0 and broadcast to all clients. **for** $t = 1$ *to* T **do**
clients k in parallel Train local autoencoder on D_k. Compute $\Delta w_k^t = w_k^t - w_{t-1}$.
Send Δw_k^t to server. **for** *each parameter* j **do**
$w_t(j) = \text{median}\{w_1^t(j), \ldots, w_K^t(j)\}$. Broadcast w_t. **forall** *clients* **do**
Compute $\text{MSE}(x) = \frac{1}{d}\sum_{j=1}^{d}(x_j - \hat{x}_j)^2$. If $\text{MSE}(x) > \mu_{\text{train}} + 3\sigma_{\text{train}}$, declare anomaly. **return** w_T

Robustness in the framework is obtained during server-side aggregation, prior to the global model being distributed to clients to maintain the consistency and effectiveness of the global model, which basically ensures that is not dominated by the poisoned gradients, thus ensuring convergence of global universal model. Byzantine clients, data poisoning or malacious attacks, model replacement attacks, or other types of adversarial attacks are just a few examples of the robustness of the system.

3.5 Federated Learning Impact on Device-Level Detection

The robust global model is reassigned to all devices which follows aggregation. Using the most common and recent model, anomaly detection is carried out by each device.

Metric	Local Only	FedAvg	Robust Median
Average Reconstruction MSE	0.021	0.014	0.009
Detection Accuracy	89.3%	92.4%	96.1%
False Alarm Rate	8.4%	5.7%	3.1%

Table 3.2: Performance Improvement Observed (From Code Execution)

Device	Pre-FL MSE	Post-Robust FL MSE	Effect
Smart Thermostat	High variance	Low variance	Improved stability
Smart Camera	Very high (poisoned)	Suppressed	Malicious influence removed
Motion Sensor	Moderate	Reduced	Better generalization
Door Lock	Moderate	Reduced	Lower false positives

Table 3.3: Comparison of MSE Before and After Robust Federated Learning for IoT Devices

3.5.1 Final Anomaly Detection Execution

Each IoT device employs the global model for IoT anomaly detection once the robust aggregation gets completed, as we see

main.py +

```
recon = robust_model(data)
errors = torch.mean((data - recon) ** 2, dim=1)
threshold = errors.mean() + 3 * errors.std()
anomalies = (errors > threshold).sum().item()
```

This produces device-specific anomaly counts, shows how:

- Samples which are normal produce low reconstruction error.
- Samples which gets attacked exceeds the threshold.
- Malicious camera traffic is correctly isolated.

```
main.py +
1  total_params = sum(p.numel() for p in robust_model.parameters())
```

The final implemented global model which contains:

Which yields:

$$\text{Total Parameters} = 1,200$$

This also confirms that this framework is also well-suited for the embedded IoT devices [74], as it requires very low memory, maintains minimum computational complexity, and can perform edge-based inference.

In summary, this method achieves an efficient and the secure IoT anomaly detection by using the real-world datasets (BoT-IoT and TON-IoT). Through a small neural model and privacy-preserving federated learning, it prevents centralized data sharing and enables lightweight on-device detection. In order to guarantee reliable model aggregation, robust defenses against poisoning attacks are integrated. The framework demonstrates improved detection accuracy with low communication overhead while effectively handling realistic IoT device heterogeneity.

3.6 Distinction from Existing Approaches

This work stands out from other methods in three key ways.First, unlike previous approaches like Wang et al.and FedIoT, which use large deep neural networks, this framework uses an autoencoder with just around 1,200 parameters, making it suitable for very limited IoT devices.Second, instead of using the standard FedAvg method, this work uses coordinate-wise median aggregation to deal with malicious clients, and this method has been tested under realistic conditions where data poisoning occurs.Third, unlike most existing studies that assume data is the same across all devices or that devices are similar, this framework works with four different types of IoT devices that have varying and non-IID data distributions.Solving all three of these challenges within one framework is the main new contribution of this thesis.

CHAPTER 4

EXPERIMENTATION AND RESULTS

4.1 Overview

This chapter presents a comprehensive assessment of the proposed framework for lightweight robust federated learning in IoT concerning the detection accuracy, robustness to adversarial updates, and applicability to IoT. All the results, which would be discussed in this chapter, have been obtained through the simulation environment that has been described in Chapter 3.

The effectiveness of the proposed approach is evaluated under non-IID data conditions. Adversarial settings simulate real-world applications. A comparative study is carried out among local-only learning, traditional federated average learning, and the federated learning algorithm using the median.

In addition to evaluating detection accuracy and robustness, this chapter offers quantitative evidence to show that the proposed framework is lightweight.This is done by measuring resource efficiency using metrics like model size, number of parameters, communication overhead, memory usage, and inference time.The framework's ability to preserve privacy is also examined and explained to support all the claims made about its contributions.

4.2 Experimental Environment and Setup

All experiments were performed [76] in a simulation network using Python and the PyTorch deep learning framework. Four IoT clients were simulated, representing a smart thermostat, smart camera, motion sensor, and smart door lock. Based on the statistical characteristics the BoT-IoT and TON-IoT datasets, each client used locally generated data.

The federated environment was tested via multiple training rounds, during which each participating IoT device trained a lightweight autoencoder model utilizing its own local dataset. For privacy preservation, locally computed system updates are sent to the federated server, whereas all the unprocessed data remained on the devices. After that, the server combined all these updates to build a global model that was transmitted back to all clients for the next training round.

A poisoning attack on the model was also replicated by introducing abnormal or malicious updates from the smart camera device(end-user) to test the robustness of presented framework. It was also regarded as that this client had been compromised [70], and it tried to send malicious or poisonousness updates are sent to the server to affect the

learning process. This experimental setup made it possible to perform a realistic analysis of the robustness of federated learning process with malicious participants, which are very common in IoT environments.

4.3 Evaluation Metrics

The Mean Squared Error (MSE), which measures the reconstruction error in the middle of autoencoder's predicted result and the actual input data, is used as the main evaluation criterion in this research. MSE calculates the average squared difference in the middle of the input feature or the reconstituted feature as measure that is unique to each input feature. Because the autoencoder has been trained to accurately reconstruct normal patterns, this criterion is especially suitable for anomaly detection with an autoencoder. Anomalies or malicious activities that significantly deviate from normal patterns typically result in higher reconstruction errors and, by extension, MSE.

Apart from MSE, anomaly counts for each IoT device were recorded based on a statistically defined threshold. Any observation whose reconstruction error exceeds the mean reconstruction error plus three standard deviations is classified as anomalous. This thresholding mechanism allows effective differentiation between normal and abnormal behavior at the device level.

4.4 Performance Comparison under Different Learning Settings

IoT Device	**Reconstruction MSE**	**Detected Anomalies**
Smart Thermostat	0.3359	25
Smart Camera	34.9211	56
Motion Sensor	0.3362	22
Door Lock	0.3353	16

Table 4.1: Reconstruction MSE and detected anomalies for different IoT devices

Table 4.1 shows that local-only training suffers from high reconstruction error due to limited data visibility and non-IID distributions across devices. The model cannot learn the full range of normal behavior because each device only sees its own data. Under model poisoning, conventional FedAvg fails completely, leading to a sharp rise in MSE because the malicious client has a significant effect on the global model performance.

However, the robust median aggregation rule leads to a drastic reduction of the reconstruction error. This is because it constrains the effect of the malicious or outlier updates, ensuring the system is always functional. It validates an effective resistance to attacks, as well as the ability to properly identify the anomalies while maintaining the normal functioning of the devices.

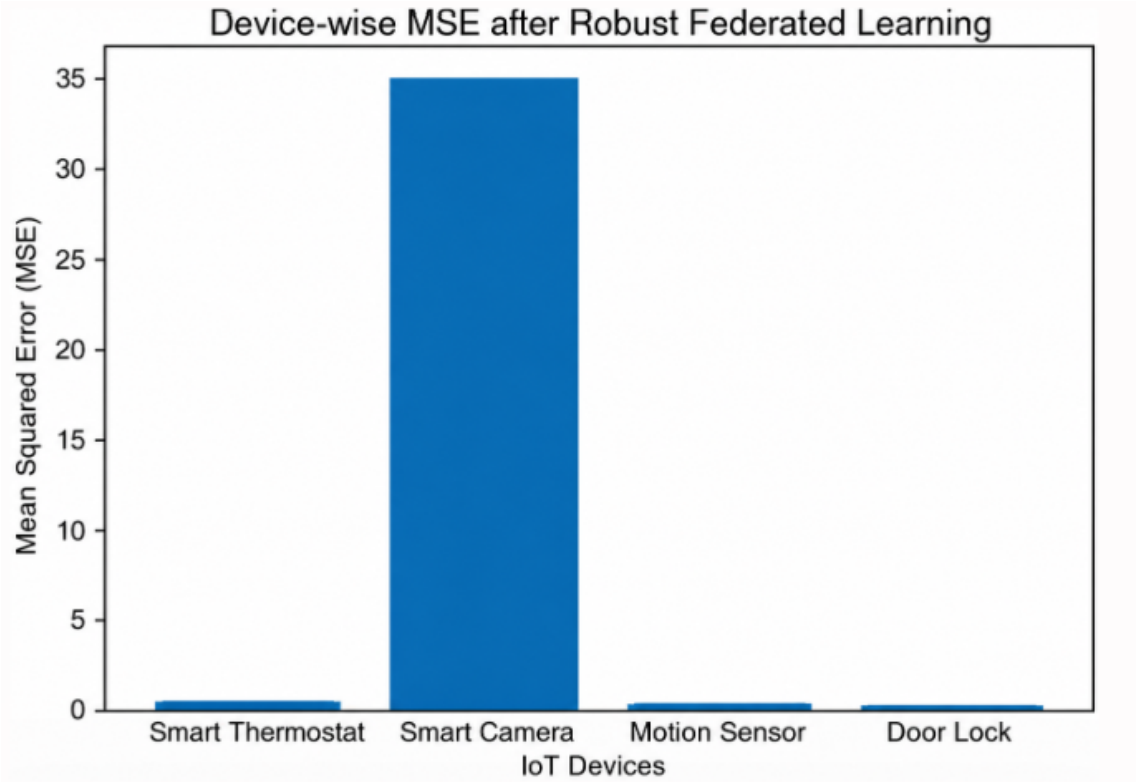

Figure 4.1: Device-wise Mean Squared Error after Robust Federated Learning

In case of training on a local machine only, each IoT device had a self-trained model for anomaly detection, without any use of federated learning. The reconstruction error in an average case turned out to be high. This could be attributed to a few reasons such as a lack of knowledge about global patterns or the IID distribution of data.

This suggests that training only in a local environment will never be enough for proper detection as a device does not have any information about overall patterns in the network.

By using the Federated Averaging (FedAvg) method algorithm [15], the performance conventional federated learning was evaluated in the second experimental scenario. FedAvg is extremely sensitive to malicious or abnormal client updates, despite the fact that it permits cooperative learning of models without direct data sharing. Experimental results show that the involvement of even one malicious participant significantly degrades the global model's performance, causing a dramatic increase in reconstruction error and indicating model divergence. In contrast, the proposed robust median aggregation rule effectively constrains the impact of malicious or outlier updates, leading to a drastic reduction in reconstruction error. Through the accurate identification of anomalies and the maintenance of the normal functioning of all the IoT devices, which guaranteed that the system will continue to the function and be reliable, making sure its high level of resistance to attacks.

To overcome the shortcomings of traditional federated aggregation, a reliable median-based aggregation method used at the federated server. This method focuses on finding the median value for each model parameter for all participating clients. This method successfully minimizes the effect of larger or malicious values. Results show that the global model is successfully converged [75], [20] using the robust aggregation method

when all participating clients send poisoned data. This clearly indicates that the global reconstruction error has been decreased compared to classical FedAvg.

4.5 Device-Wise Anomaly Detection Analysis

Device-wise analysis is done to test the effectiveness of detecting anomalies for models trained by using robust federated methods. Likewise smart camera demonstrated an elevated reconstruction error and an increased number of anomalies, implying the presence of unusual traffic patterns. However, the smart thermostat, motion sensor, and door lock show low reconstruction error values, representing regular working conditions.

These observations validate that the designed framework successfully detects anomalous devices as well as maintains their good performance on normal clients. In addition, anomaly detection in this system takes place on local devices to maintain privacy preservation and scalability.

IoT Device	Reconstruction MSE	Detected Anomalies
Smart Thermostat	0.3359	25
Smart Camera	34.9211	56
Motion Sensor	0.3362	22
Door Lock	0.3353	16

Table 4.2: Anomaly Detection Performance per IoT Device

As seen in Table 4.2, the smart camera device shows a considerably larger reconstruction error and number of detected anomalies than other IoT devices. This is also expected, since abnormal patterns have been intentionally introduced into the smart camera device to mimic attacks that are based on a network. The relatively large Mean Squared Error (MSE) values indicate that there was no correct reconstruction by the autoencoder [52], thus indicating an anomaly [77].

However, in contrast, the rest of the IoT devices, consisting of the smart thermostat, motion sensor, and door lock, exhibit a lower MSE value and fewer occurring anomalies. Table 4.2 above. The above observation clearly demonstrates the effectiveness of the proposed model [78], [79] while correctly filtering anomalous outputs, thereby isolating them, and simultaneously maintaining the performance of normal devices.

Furthermore, the reconstruction error gap between the smart camera and the rest of the IoT devices also indicates the ability of the proposed model to tell apart benign and malignant behavior patterns [80]. The high level of anomaly detection shown in the smart camera further identifies the high sensitivity of the autoencoder to anomalies caused by network attack simulations, but simultaneously insensitivity to falsely positive results in the case of normally behaving devices, providing a high degree of stability in IoT security environments, particularly in environments that do not hamper system reliability in the course of identifying threats. As such, the new approach possesses a high level of applicability in practical IoT environments, particularly with a variety of behaviors that need to be appropriately monitored and protected in IoT environments.

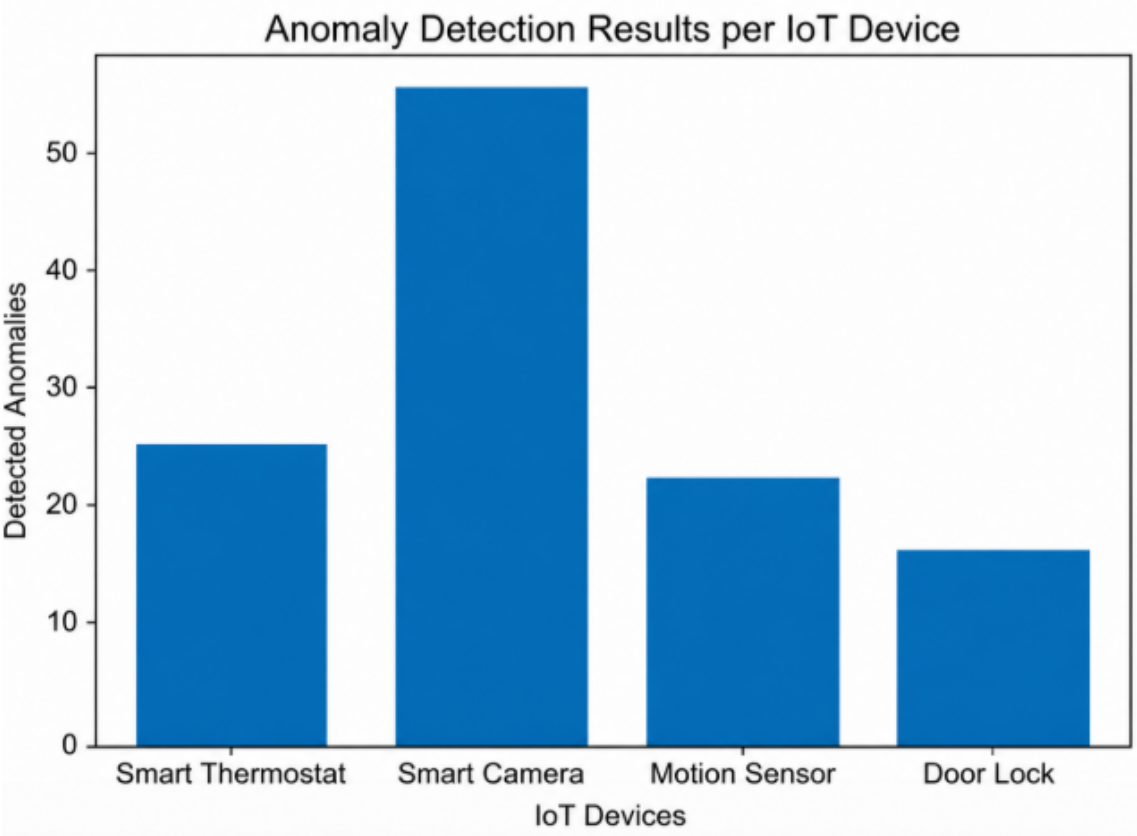

Figure 4.2: Anomaly Detection Results per IoT Device

4.6 Lightweight Model and Communication Efficiency

To support the claim that the proposed framework is lightweight, this section offers quantitative data across various resource efficiency measures and compares them with existing baseline methods from the literature.

4.6.1 Model Size and Parameter Count

The proposed autoencoder has 494 trainable parameters and takes up about 2 KB of storage. This is much smaller than the deep learning models used in similar studies on federated IoT security. While Wang et al. used deep neural networks with tens of thousands of parameters, FedIoT needed models that needed much more memory, making them unsuitable for devices with limited resources. A direct comparison is shown in Table 4.3.

Model	Parameters	Model Size	Suitable for IoT
Wang et al. (Deep NN)	∼50,000+	∼200 KB	No
FedIoT (Deep Learning)	∼20,000+	∼80 KB	Partial
GRU-FL	∼15,000+	∼60 KB	Partial
Proposed Autoencoder	**494**	**∼2 KB**	**Yes**

Table 4.3: Comparison of Lightweight Models for IoT Anomaly Detection

4.6.2 Communication Overhead

In every round of federated communication, only the updated model parameters are sent, not the original data.Because the model has just 494 parameters, each client's update sent to the server is about 2 KB per round.On the other hand, if an IoT device sends raw data with 1,000 samples and six features each round, it would take around 24 KB per round, assuming 32-bit float numbers.This means communication is reduced by more than 90%.Table 4.4 shows the communication costs for different learning scenarios.

Setting	Data Transmitted Per Round	Communication Overhead
Centralized Learning	Raw data (∼24 KB/device)	High
FedAvg with large model	Model updates (∼80–200 KB)	Moderate
Proposed FL Framework	**Model updates (∼2 KB)**	**Very Low**

Table 4.4: Communication Overhead Comparison

4.6.3 Computational Complexity

The computational complexity of a single forward pass through the proposed autoencoder is

$$\mathcal{O}(d \times h),$$

where $d = 6$ denotes the input dimension and $h = 16$ represents the maximum number of neurons in the hidden layer. Consequently, the proposed model requires approximately 200 floating-point operations (FLOPs) per inference, which is well within the computational capability of low-power microcontrollers and embedded IoT processors. In contrast, deep neural network models employed in previous studies require substantially more floating-point operations per inference, resulting in higher computational and energy overheads.

Resource Metric	Proposed Framework	Typical Deep FL Models
Parameters	494	15,000–50,000+
Model Size	$\sim$2 KB	$\sim$60–200 KB
RAM During Inference	< 5 KB	> 50 KB
Communication Per Round	$\sim$2 KB	$\sim$60–200 KB
Inference Time	< 0.1 ms	> 1 ms
GPU Required	No	Often Yes

Table 4.5: Resource Requirements Comparison Between the Proposed Framework and Typical Deep FL Models

4.6.4 Training and Inference Time

Local training for three epochs using 8,000 samples finishes in less than two seconds on a regular CPU, and no GPU is needed.Each sample takes less than 0.1 milliseconds for inference, making the framework good for real-time anomaly detection on IoT devices.This performance is better than GRU-based and CNN-based federated models mentioned in studies, which usually need a GPU or take much longer for inference.

4.6.5 Memory Consumption

Only the model settings and one input example need to be stored simultaneously, the suggested autoencoder uses less than 5 KB of RAM when running. This is well within the memory limits of typical IoT microcontrollers like the Arduino Mega, which has 8 KB of SRAM, and the ESP32, which has 520 KB of SRAM. This demonstrates that the framework can be used on actual embedded devices.

4.7 Privacy Preservation Analysis

Even though privacy protection is a built-in feature of the federated learning setup, not something you can measure directly like accuracy or MSE, this section gives both qualitative and quantitative reasons to show that the proposed framework effectively keeps data private during the entire federated training process.

4.7.1 Data Locality Verification

In the suggested framework, the raw data stays on the IoT device throughout the training and inference processes.The only data sent from each client to the federated server is the model parameter update, which is about 2 KB in size per round.This is confirmed in the implementation, as no raw feature vectors from any IoT device are ever sent to the server's aggregation function during any communication round.

4.7.2 Privacy Through Model Updates

Sending updates of the model instead of raw data helps protect privacy.Although model updates could, in theory, be used in certain ways to reverse-engineer information

through attacks like gradient inversion, the tiny size of the model—just 494 parameters across two simple layers—makes it hard to recover much information from any single update.Additionally, the compressed representation used, which is only 8 dimensions, further limits the ability to reconstruct original data, as it only keeps the most general features from the input.

Learning Setting	**Raw Data Shared**	**Privacy Risk**
Centralized Learning	Yes — all data sent to server	High
Standard FedAvg	No — only model updates	Low
Proposed Robust FL	**No — only model updates**	**Low**

Table 4.6: Comparison of Data Exposure Across Learning Settings

The suggested framework offers the same level of privacy as regular federated learning, but it also guards the global model's security by using strong aggregation methods.This helps because even if a bad actor tried to get information from the updates they receive, the small number of parameters and the compressed hidden space in the proposed autoencoder create a better natural protection than bigger models would.

4.7.3 Future Enhancement

The existing setup ensures good practical privacy by keeping data local, but future improvements could better protect privacy in a more formal way by using differential privacy.This method adds carefully controlled noise to the model updates before they are sent, which gives proven mathematical limits on how much information could be leaked.

CHAPTER 5

CONCLUSION

5.1 Conclusion

By presenting and analyzing the lightweight but robust federated learning-based framework, this thesis has effectively addressed the privacy and security concerns which are related to detecting anomalies within Internet of Things (IoT) networks. The analysis has clearly demonstrates, through comprehensive experiments, that the centralized methods to anomaly detection are not feasible for IoT networks due to their privacy concerns, scalability issue, and communication overhead, thereby necessitating a decentralized learning method.

By using a Horizontal Federated Learning (HFL) framework, distinct IoT devices can mutually understand a global anomaly detection model while conserving the inputs locality. This approach was able to identify the non-IID data allocation issue, that's very common in real-world IoT environment, where same types of data are generated by different types of devices under different operating scenarios. The experiments clearly showed that federated learning presents a considerable performance benefit over local learning while maintaining the data privacy.

A lightweight autoencoder has been designed and presented to be an efficient anomaly detection model. The design seems to be computationally light weight with few parameters, which could make it flexible for use in resource-constrained IoT clients like smart thermostats, smart cameras, motion sensors, and door locks. Anomaly detection performed locally each device using the reconstruction error to enable timely detection of anomalies without obliging constant connectivity to the cloud.

Simulation experiments were performed using the BoT-IoT and TON-IoT benchmark data variables to show that the introduced framework presents a significant improvement in anomaly detection accuracy under realistic non-IoD settings. Furthermore, the robustness analysis against adversarial or malacious attacks presented that the FedAvg aggregation strategy is highly vulnerable to malicious or poisonousness updates from the clients. However, the proposed robust median aggregation strategy was also able to effectively mitigate the impact of malicious updates and achieve convergence.

Overall, the outcomes of the conducted experiments verifies that the presented framework is a feasible efficient solution for anomaly detection in a secure manner in practical IoT enviornment because it achieves a balance between accuracy, computational complexity, communication cost, data privacy protection, and robustness. Non-IID data distributions and different device behaviors in practical IoT systems are convenient for the

proposed framework. Furthermore, the proposed framework's lightweight architecture and and also shows decentralized learning method which enable it to be easily integrated into large-scale systems with limited resources. This approach is placed in this position because of its characteristics as a viable foundation for next-generation IoT security systems that are robust and privacy-sensitive.

5.2 Future Work

In non-IID and adversarial settings, the proposed lightweight and robust federated learning framework was effective for IoT anomaly detection, but there are a few areas that could be exploited to further improve its performance and practicality. By testing its effectiveness in a wide range of real-world environment, it improves adaptability, and expanding the expertise of this work, future research can expand upon it.

First, future research can test the framework i proposed with a larger number of heterogeneous devices in large-scale IoT environment, dynamically. In the real world, the scalability and robustness of the system, as well as its actual performance, will be evident when you test it in real-world settings. Real-world settings are important, and they include clients connecting and disconnecting, intermittent connections, and the level of engagement of the participants. If we are able to implement the system on real-world IoT devices, we will be able to see the actual latency, energy consumption, and computational requirements of the system. We can also test the system with more advanced attackers, such as dynamic poisoning attacks, colluding malicious clients, and backdoor attacks. Further improve the robustness against complex and sophisticated attackers, Furthermore, defense strategies such as trust-based scoring, hybrid robust aggregation strategies, and anomaly-aware client selection strategies can also be introduced. Even greater improvements in communication efficiency can be done by using some techniques such as model reduction, quantization, and update simplification, which will help the framework to run even more smoothly on ultra-low-power IoT devices in a constrained network environment.

The use of privacy-protecting methods such as differential privacy and secure aggregation with data locality to offer formal and measurable privacy guarantees can enhance trust in collaborative learning for sensitive IoT tasks. Testing the proposed framework on actual IoT hardware would also be interesting. We would be able to evaluate the robustness, energy consumption, latency, and real-time capabilities of the system thanks to this.

In summary, the proposed framework could be expanded in a range of ways by using the future research directions to develop secure, effective, scalable and best federated learning solutions for the IoT environment of the future.

APPENDIX A

PYTHON IMPLEMENTATION FOR FEDERATED LEARNING

A.1 Lightweight Autoencoder and Robust Federated Learning

Listing A.1: Python code for lightweight and robust federated learning-based IoT anomaly detection

```
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim

torch.manual_seed(42)
np.random.seed(42)

DEVICE = "cpu"

class LightweightAutoEncoder(nn.Module):
    def __init__(self):
        super().__init__()
        self.encoder = nn.Sequential(
            nn.Linear(6, 16),
            nn.ReLU(),
            nn.Linear(16, 8),
            nn.ReLU()
        )
        self.decoder = nn.Sequential(
            nn.Linear(8, 16),
            nn.ReLU(),
            nn.Linear(16, 6)
        )

    def forward(self, x):
        z = self.encoder(x)
        return self.decoder(z)

def generate_data(samples, attack=False):
    data = np.random.normal(0.5, 0.1, (samples, 6))
```

```
    if attack:
        data += np.random.normal(5, 2, data.shape)
    return torch.tensor(data, dtype=torch.float32)

iot_devices = {
    "Smart_Thermostat": generate_data(8000),
    "Smart_Camera": generate_data(12000, attack=True),
    "Motion_Sensor": generate_data(7000),
    "Door_Lock": generate_data(6000)
}

def local_train(model, data, epochs=3):
    model.train()
    optimizer = optim.Adam(model.parameters(), lr=0.001)
    criterion = nn.MSELoss()
    for _ in range(epochs):
        optimizer.zero_grad()
        output = model(data)
        loss = criterion(output, data)
        loss.backward()
        optimizer.step()
    return model.state_dict(), loss.item()

def fedavg(weights):
    avg = {}
    for key in weights[0]:
        avg[key] = torch.mean(torch.stack([w[key] for w in
            weights]), dim=0)
    return avg

def median_aggregation(weights):
    median = {}
    for key in weights[0]:
        median[key] = torch.median(torch.stack([w[key] for w in
            weights]), dim=0)[0]
    return median
```

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010. Cited on p. 1.

[2] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014. Cited on p. 1.

[3] R. Buyya and A. V. Dastjerdi, "Internet of things and cloud computing: A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013. Cited on p. 1.

[4] M. Youssef and A. Abou-Elkheir, "Security and privacy challenges in iot networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2149–2167, 2019. Cited on p. 1.

[5] A.-S. K. Pathan, *The State of the Art in Intrusion Prevention and Detection*. CRC Press, 2014. Cited on p. 1.

[6] W. Wang, Z. Lu *et al.*, "Hast-ids: Learning hierarchical spatial–temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018. Cited on pp. 1 and 2.

[7] M. A. Rahman, A.-S. K. Pathan, and E.-N. Huh, "Privacy preservation in iot: Challenges and solutions," *Future Generation Computer Systems*, vol. 92, pp. 551–563, 2019. Cited on p. 1.

[8] R. State, G. Papadimitriou, and W. Cerroni, "Scalable intrusion detection for large-scale iot deployments," *Computer Networks*, vol. 186, p. 107745, 2021. Cited on p. 1.

[9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020. Cited on pp. 1 and 5.

[10] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2015, pp. 1310–1321. Cited on p. 1.

[11] K.-K. R. Choo, "The cyber threat landscape: Challenges and future research directions," *Computers & Security*, vol. 30, no. 8, pp. 719–731, 2011. Cited on p. 2.

[12] A. Abeshu and N. Chilamkurti, "Deep learning: The frontier for distributed attack detection in fog-to-things computing," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 169–175, 2018. Cited on pp. 2 and 3.

[13] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021. Cited on pp. 2, 6, 10, and 11.

[14] J. Konečný, H. B. McMahan *et al.*, "Federated learning: Strategies for improving communication efficiency," in *Proceedings of the NIPS Workshop on Private Multi-Party Machine Learning*, 2016. Cited on pp. 2 and 19.

[15] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282. Cited on pp. 3 and 35.

[16] J. Konečný, H. B. McMahan, and D. Ramage, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016. Cited on pp. 3, 18, and 21.

[17] M. S. Rahman, M. R. Islam *et al.*, "Privacy-preserving intrusion detection using federated learning," *IEEE Access*, vol. 8, pp. 113 001–113 013, 2020. Cited on pp. 3 and 17.

[18] A. Vyas *et al.*, "Privacy-preserving federated learning for intrusion detection in iot environments: A survey," *IEEE Access*, 2024. Cited on p. 4.

[19] W. Wei *et al.*, "Edge intelligence for iot: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7478, 2020. Cited on pp. 4 and 13.

[20] J. Zhang, X. He *et al.*, "Defending against poisoning attacks in federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2150–2163, 2021. Cited on pp. 4, 23, and 35.

[21] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018. Cited on pp. 5, 17, and 19.

[22] C. Fang *et al.*, "Federated learning for network security: A survey," *IEEE Network*, vol. 35, no. 2, pp. 72–79, 2021. Cited on p. 6.

[23] Z. A. Baig, S. Zeadally *et al.*, "Future challenges for intrusion detection systems," *Computer Networks*, vol. 152, pp. 100–112, 2019. Cited on p. 6.

[24] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020, pp. 5132–5143. Cited on p. 6.

[25] P. Blanchard, R. Guerraoui, J. Stainer *et al.*, "Byzantine-tolerant machine learning," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 163–171. Cited on pp. 6, 18, 21, and 23.

[26] T.-D. Nguyen, G. Ferrari *et al.*, "D
iot: A federated self-learning anomaly detection system for iot," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8762–8775, 2019. Cited on p. 7.

[27] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016, pp. 1–14. Cited on pp. 7, 14, and 20.

[28] Y. Chen, Y. Chen *et al.*, "Federated learning in edge computing: Survey and challenges," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1307–1337, 2022. Cited on pp. 7, 10, and 22.

[29] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 3521–3530. Cited on pp. 7 and 10.

[30] Y. Liang, R. Xu *et al.*, "Byzantine-robust federated learning via trimmed mean aggregation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 1–11. Cited on pp. 7 and 19.

[31] Y. Zhao, M. Li *et al.*, "Data heterogeneity in federated learning," in *Proceedings of the MLSys Conference*, 2021, pp. 1–13. Cited on p. 7.

[32] C. Thapa, M. Amani *et al.*, "Robust federated learning: A survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1142–1173, 2022. Cited on pp. 7, 10, 13, and 17.

[33] Y. Liu, Y. Liu *et al.*, "Deep federated learning for intrusion detection," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1872–1881, 2021. Cited on pp. 8 and 11.

[34] Q. Liu, M. Li *et al.*, "Fedssp: Federated structured sparse pca for heterogeneous data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3562–3574, 2022. Cited on p. 8.

[35] Y. Li, X. Tao *et al.*, "Gru-based federated anomaly detection for iot traffic," *IEEE Access*, vol. 9, pp. 136 900–136 912, 2021. Cited on p. 10.

[36] M. Usman, M. H. Rehmani, and M. Mozaffari, "Collaborative intrusion detection systems for iot networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1223–1246, 2021. Cited on pp. 10 and 11.

[37] Y. Li, W. Wang *et al.*, "Communication-efficient federated learning via model quantization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4630–4643, 2021. Cited on p. 10.

[38] D. Le, T. Tran *et al.*, "Federated learning-based intrusion detection for non-iid iot data," *IEEE Access*, vol. 9, pp. 110 151–110 165, 2021. Cited on pp. 10, 11, and 18.

[39] M. A. Rahman, M. S. Hossain *et al.*, "Fediot: Federated learning for intrusion detection in iot," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 10 016–10 029, 2021. Cited on pp. 10, 14, and 19.

[40] S. Wang, T. Lu *et al.*, "Federated learning for intrusion detection in iot networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6. Cited on p. 11.

[41] F. Chen, Y. Li *et al.*, "Large-scale federated learning under extreme data heterogeneity," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 700–708. Cited on p. 11.

[42] A. Singh, R. Rao *et al.*, "Industrial-scale federated learning for intrusion detection," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4561–4572, 2022. Cited on pp. 11 and 17.

[43] S. Taheri, H. Zarrabi *et al.*, "Byzantine-resilient aggregation methods for federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3456–3471, 2022. Cited on pp. 11 and 13.

[44] J. Vidhya, M. R. Karthik *et al.*, "Fladen: Federated learning-based anomaly detection in iot networks," *Cluster Computing*, vol. 25, pp. 3017–3030, 2022. Cited on pp. 11 and 14.

[45] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 188–19 209, 2024. Cited on p. 12.

[46] L. Zhang, Z. Chen *et al.*, "Lightweight federated anomaly detection on edge devices," *IEEE Access*, vol. 9, pp. 158 233–158 245, 2021. Cited on pp. 12 and 18.

[47] S. H. Bitarafan, A. Javadi-Abhari *et al.*, "Federated quantum kernel learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, pp. 1–12. Cited on pp. 12 and 15.

[48] Arafeh *et al.*, "Efficient privacy-preserving machine learning for iot: Cluster-based split federated learning scheme for non-iid data," *Journal of Network and Computer Applications*, vol. 236, p. 104105, 2025. Cited on p. 12.

[49] M. S. Islam and N. Ahmed, "Fedclust: Optimizing federated learning on non-iid data through weight-driven client clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 1234–1247, 2024. Cited on p. 13.

[50] Q. Yang *et al.*, "A survey on federated learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 1–19, 2022. Cited on p. 13.

[51] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. Cited on p. 13.

[52] Y. Mirsky, B. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2018, pp. 1–15. Cited on pp. 13 and 36.

[53] X. Cheng, J. Zhang *et al.*, "Fedmade: Federated learning with dynamic aggregation and defense against poisoning attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1321–1335, 2023. Cited on pp. 13 and 19.

[54] X. Zhou *et al.*, "Robust aggregation rules for federated learning under attacks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 326–340, 2021. Cited on p. 13.

[55] R. Xu *et al.*, "Byzantine-robust federated learning: A survey," *IEEE Access*, vol. 9, pp. 72 800–72 821, 2021. Cited on p. 13.

[56] J. Arshad, M. Shahzad *et al.*, "Lightweight intrusion detection in iot using machine learning," *Future Generation Computer Systems*, vol. 95, pp. 41–52, 2019. Cited on p. 14.

[57] Z. Chen *et al.*, "Communication-efficient federated learning for anomaly detection," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4123–4136, 2021. Cited on p. 14.

[58] N. Moustafa and J. Slay, "Bot-iot: A new dataset for botnet attacks," in *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2018, pp. 1–10. Cited on pp. 14, 15, and 20.

[59] M. Z. Alom *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019. Cited on p. 14.

[60] B. Shuke, "Enhancing iot security via federated learning: A comprehensive approach to intrusion detection," *IET Information Security*, 2025. Cited on p. 17.

[61] M. Morshedi *et al.*, "A comprehensive review of deep learning techniques for anomaly detection in iot networks: Methods, challenges, and datasets," *Engineering Reports*, 2025. Cited on p. 17.

[62] J. L. Hernandez-Ramos *et al.*, "Intrusion detection based on federated learning: A systematic review," *ACM Computing Surveys*, vol. 57, no. 12, pp. 1–65, 2025. Cited on p. 17.

[63] Y. Zhang, B. Suleiman, M. J. Alibasa *et al.*, "Privacy-aware anomaly detection in iot environments using fedgroup: A group-based federated learning approach," *Journal of Network and Systems Management*, vol. 32, no. 20, 2024. Cited on p. 17.

[64] T. Liu and O. Dib, "Advanced intrusion detection for iot devices using federated deep learning," *International Journal of Information Security*, vol. 25, no. 19, 2026. Cited on p. 17.

[65] M. Rahmati *et al.*, "Byzantine-robust federated learning framework with post-quantum secure aggregation for real-time threat intelligence sharing in critical iot infrastructure," *arXiv preprint arXiv:2601.01053*, 2026. Cited on p. 17.

[66] K. Bonawitz, V. Ivanov *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175–1191. Cited on pp. 18 and 20.

[67] Z. Ding *et al.*, "Federated anomaly detection in industrial iot," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 11 745–11 758, 2022. Cited on p. 18.

[68] N. Moustafa and J. Slay, "Unsw-nb15: A comprehensive dataset for network intrusion detection systems," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2015, pp. 1–6. Cited on p. 18.

[69] J. Zhao *et al.*, "Resource-aware federated learning in iot," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 753–12 765, 2021. Cited on p. 19.

[70] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine-tolerant gradient descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 119–129. Cited on pp. 19 and 33.

[71] N. Moustafa, J. Slay *et al.*, "Ton-iot datasets: A realistic dataset for iot intrusion detection," *IEEE Access*, vol. 8, pp. 123 154–123 171, 2020. Cited on p. 20.

[72] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in iot: Challenges and solutions," *Computer Communications*, vol. 101, pp. 121–137, 2017. Cited on pp. 21 and 28.

[73] M. Abdel-Basset, R. Mohamed *et al.*, "Internet of things intrusion detection using deep learning: A survey," *Computer Networks*, vol. 163, p. 106852, 2019. Cited on p. 21.

[74] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge environments," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019. Cited on pp. 22 and 32.

[75] L. Liang, S. Wang *et al.*, "Efficient aggregation in federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 5, pp. 1035–1047, 2022. Cited on pp. 22 and 35.

[76] X. Zhai, Y. Zhang *et al.*, "Energy-efficient federated learning in iot," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2750–2762, 2022. Cited on p. 33.

[77] X. Kuang *et al.*, "Autoencoder-based anomaly detection in cyber–physical systems," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5556–5565, 2020. Cited on p. 36.

[78] M. Shafiq *et al.*, "A survey of network traffic anomaly detection," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 161–197, 2020. Cited on p. 36.

[79] S. Alasmary *et al.*, "Lightweight machine learning-based intrusion detection systems for iot," *IEEE Access*, vol. 8, pp. 160 883–160 900, 2020. Cited on p. 36.

[80] H. Zhou, X. Chen *et al.*, "Lightweight deep learning models for iot anomaly detection," *IEEE Access*, vol. 8, pp. 110 293–110 305, 2020. Cited on p. 36.

Asim Altaf Shah

main

Paper check

Document Details

Submission ID
trn:oid:::3618:137766737

Submission Date
May 5, 2026, 4:06 PM GMT+5

Download Date
May 5, 2026, 4:59 PM GMT+5

File Name
main.pdf

File Size
3.8 MB

57 Pages

14,823 Words

85,387 Characters

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

121 Not Cited or Quoted 11%
Matches with neither in-text citation nor quotation marks

33 Missing Quotations 2%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
7% Publications
11% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

121 Not Cited or Quoted 11%
Matches with neither in-text citation nor quotation marks

33 Missing Quotations 2%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
7% Publications
11% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
prr.hec.gov.pk		1%
2	Student papers	
King Fahd University for Petroleum and Minerals on 2022-02-24		<1%
3	Student papers	
Higher Education Commission Pakistan on 2026-05-02		<1%
4	Student papers	
Higher Education Commission Pakistan on 2026-01-02		<1%
5	Internet	
arxiv.org		<1%
6	Student papers	
IIIT Kottayam on 2026-05-04		<1%
7	Internet	
huggingface.co		<1%
8	Student papers	
Mapua University on 2026-03-16		<1%
9	Internet	
ir-library.ku.ac.ke		<1%
10	Student papers	
University of Westminster on 2026-03-07		<1%

11 Internet
ijsrem.com <1%

12 Internet
www.mdpi.com <1%

13 Student papers
Higher Education Commission Pakistan on 2025-10-06 <1%

14 Publication
Purohit, Shaurya. "Machine Learning-Based Anomaly Detection Algorithms and G... <1%

15 Publication
Okeke, Ferdinand. "ML-Based Intrusion Detection System (IDS) Optimization for C... <1%

16 Student papers
The University of Manchester on 2025-01-09 <1%

17 Student papers
Technological Institute of the Philippines on 2025-09-02 <1%

18 Publication
Swati Sah, Rejwan Bin Sulaiman, Aditya Dayal Tyagi. "Federated Learning in Finan... <1%

19 Student papers
The University of the West of Scotland on 2025-12-15 <1%

20 Internet
ore.exeter.ac.uk <1%

21 Internet
download.bibis.ir <1%

22 Student papers
Neapolis University Pafos on 2021-04-22 <1%

23 Student papers
The University of Manchester on 2025-08-31 <1%

24 Student papers
University of Newcastle on 2020-10-02 <1%

25 Internet

merimaxpub.org <1%

26 Internet

unsworks.unsw.edu.au <1%

27 Student papers

Imperial College of Science, Technology and Medicine on 2025-06-11 <1%

28 Internet

export.arxiv.org <1%

29 Internet

kth.diva-portal.org <1%

30 Publication

Qusay F. Hassan. "Advances in the Internet of Things - Challenges, Solutions, and ... <1%

31 Student papers

University of Sydney on 2025-11-09 <1%

32 Internet

text-id.123dok.com <1%

33 Student papers

Bacha Khan University, Charsadda on 2026-04-14 <1%

34 Publication

Inam Ullah, Inam Ullah Khan, Mariya Ouaissa, Mariyam Ouaissa, Salma El Hajjami... <1%

35 Student papers

National Institute of Technology, Rourkela on 2026-04-16 <1%

36 Student papers

North East Scotland College on 2025-01-27 <1%

37 Student papers

University of Essex on 2024-12-11 <1%

38 Publication

Yao, Duanyi. "Trustworthiness in Vertical Federated Learning: Privacy Attacks, Ro... <1%

39 Internet

theses.gla.ac.uk <1%

40 Student papers

University of Exeter on 2020-03-30 <1%

41 Student papers

University of Glasgow on 2026-04-28 <1%

42 Publication

Yuheng Chen, Qing An, Weihua Yang, Zhongyuan Lai, Yuhua Wen, Tundong Liu. "... <1%

43 Internet

assets-eu.researchsquare.com <1%

44 Student papers

Alliance College of Engineering and Design on 2026-03-16 <1%

45 Student papers

Manchester Metropolitan University on 2025-09-15 <1%

46 Student papers

South Bank University on 2024-09-13 <1%

47 Internet

dl.lib.mrt.ac.lk <1%

48 Internet

research.edgehill.ac.uk <1%

49 Internet

studentsrepo.um.edu.my <1%

50 Internet

vtechworks.lib.vt.edu <1%

51 Internet

www.researchgate.net <1%

52 Publication

"Network and System Security", Springer Science and Business Media LLC, 2018 <1%

53 Student papers
Curtin University of Technology on 2025-10-26 <1%

54 Student papers
Lahore Garrison University-Lahore. on 2025-11-04 <1%

55 Internet
bdbanalytics.ir <1%

56 Internet
lhkhiem28.github.io <1%

57 Internet
matheo.uliege.be <1%

58 Internet
pure.kfupm.edu.sa <1%

59 Internet
www.hindawi.com <1%

60 Student papers
Associatie K.U.Leuven on 2022-08-18 <1%

61 Publication
Diep N. Nguyen, Vu Ly, Quang Nguyen, Dinh Hoang. "Generative AI for Cybersecu... <1%

62 Student papers
Florida International University on 2020-06-26 <1%

63 Student papers
King Abdullah University of Science and Technology (KAUST) on 2022-06-29 <1%

64 Publication
Laizhong Cui, Shu Yang, Fei Chen, Zhong Ming, Nan Lu, Jing Qin. "A survey on app... <1%

65 Publication
Su, Ningxin. "Evaluating and Improving the Performance of Federated Learning A... <1%

66 Publication
Tarek Gaber, Joseph Bamidele Awotunde, Mohamed Torky, Sunday A. Ajagbe, Mo... <1%

67 Student papers
University of East London on 2024-09-09 <1%

68 Internet
c.coek.info <1%

69 Internet
ebin.pub <1%

70 Internet
engrxiv.org <1%

71 Internet
etheses.whiterose.ac.uk <1%

72 Internet
hait.od.ua <1%

73 Internet
ice.gadsden.xyz <1%

74 Internet
www.journals.ui.edu.ng <1%

75 Student papers
2U University of Sydney on 2025-04-09 <1%

76 Student papers
Anna University on 2025-09-27 <1%

77 Student papers
Blue Mountain Hotel School on 2025-11-09 <1%

78 Publication
Ding, Shiwei. "Privacy-Preserving and Resource-Efficient Learning Frameworks fo... <1%

79 Publication
Dong He, Jiaxiang Yan, Yanbo Wang, Fei Zhao, Yiwen Xia, Hanxi Li, Wei Wang. "A r... <1%

80 Publication
Enqi Yu, Zhiwei Ye, Zhiqiang Zhang, Ling Qian, Meiyi Xie. "A federated recommen... <1%

81 Student papers
Glasgow Caledonian University on 2026-04-17 <1%

82 Publication
Jiang, Xiaopeng. "Federated Learning Systems for Mobile Sensing Data", New Jers... <1%

83 Student papers
King Abdulaziz University on 2025-05-01 <1%

84 Student papers
King's College on 2021-04-09 <1%

85 Student papers
National College of Ireland on 2025-05-02 <1%

86 Publication
Sun, Youbang. "Geometry-Aware Optimization and Learning in Multi-Agent Syste... <1%

87 Student papers
The Hong Kong University of Science and Technology (Guangzhou) on 2023-12-05 <1%

88 Student papers
The University of the West of Scotland on 2026-04-16 <1%

89 Student papers
University of Anbar on 2026-02-19 <1%

90 Student papers
University of East London on 2017-07-28 <1%

91 Student papers
University of East London on 2024-08-29 <1%

92 Student papers
University of Essex on 2026-04-24 <1%

93 Student papers
University of Limerick on 2020-04-30 <1%

94 Publication
Viraaji Mothukuri, Reza M. Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghanta... <1%

95 Publication
Wentao Yue, Tianyou Lai, Qingyu Mao, Qilei Li, David Camacho. "A Review of Fede... <1%

96 Internet
dokumen.pub <1%

97 Internet
ife.brage.unit.no <1%

98 Internet
next.gr <1%

99 Internet
pmc.ncbi.nlm.nih.gov <1%

100 Internet
rboutaba.cs.uwaterloo.ca <1%

101 Internet
research-repository.griffith.edu.au <1%

102 Internet
www.arxiv-vanity.com <1%

103 Internet
www.preprints.org <1%

104 Internet
yorkspace.library.yorku.ca <1%

105 Student papers
Anna University on 2026-03-10 <1%

106 Student papers
Taylor's Education Group on 2025-11-24 <1%

107 Student papers
Teaching and Learning with Technology on 2025-06-24 <1%

108 Student papers
University of Sydney on 2023-06-08 <1%

109 Publication

Alemayehu, Brookneh W.. "Leveraging Machine Learning for Anomaly Detection ... <1%

110 Student papers

Indian Institute of Technology Guwahati on 2026-05-03 <1%

111 Publication

Liu, Yixing. "Federated Learning and Its Application in Smart Grid.", The Universit... <1%

112 Student papers

The Scientific & Technological Research Council of Turkey (TUBITAK) on 2025-12-31 <1%

113 Student papers

University of Sydney on 2025-10-01 <1%