Metric Annealed Federated Learning (MAFL) - An Efficient and Hierarchical Approach towards NIDS for IoT Edge Devices

by

Andalib Iftakher
22101348
Md. Minhazul Islam Royel
22101683
Tahmid Bin Haque
22101395
Yeasin Ibne Kadir
22101048
Farhan Sadi Enan
22101134

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
BRAC University
October 2025.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Andalib Iftakher
22101348

Md.Minhazul Islam Royel
22101683

Tahmid Bin Haque
22101395

Yeasin Ibne Kadir
22101048

Farhan Sadi Enan
22101134

Approval

The thesis titled "Metric Annealed Federated Learning (MAFL) - An Efficient and Hierarchical Approach towards NIDS for IoT Edge Devices" submitted by

1. Andalib Iftakher (22101348)
2. Md. Minhazul Islam Royel (22101683)
3. Tahmid Bin Haque (22101395)
4. Yeasin Ibne Kadir (22101048)
5. Farhan Sadi Enan (22101134)

of Summer, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science.

Examining Committee:

Supervisor:
(Member)

Amitabha Chakrabarty, PhD

Professor
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

The Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

The proliferation of Internet of Things (IoT) devices has created unprecedented security challenges for network intrusion detection systems. Starting from simple implementation, the usage of IoT has expanded in many industries to increase their robustness. Unfortunately, the drawbacks of such distribution are also emerging at an alarming rate. Gradually attackers are developing new methods to gain unauthorized access to an IoT network. Over the past few years, the frequency of attacks like Distributed Denial of Service (DDoS) has significantly risen. The importance of protecting network and data has risen sharply. Current state-of-the-art approaches remain insufficient to provide a comprehensive solution in this field. Numerous studies have been attempted using deep learning architectures and supervised machine learning methodologies for binary and multiclass classification tasks. However, these efforts have been consistently hindered by class imbalance issues within the available datasets, resulting in suboptimal model performance and limiting the practical applicability of these approaches. Moreover, literature on compromised node detection are faintly in number and light-weight machine learning models had a false positive rate of around 30 percent overall. In this meticulous study, Deep Learning models perform with high accuracy (99%) in binary classification although they struggle with multi classifications. Additionally, ubiquitous data-driven approaches are parameter heavy and executed in a controlled environment. Given the scarcity of real-world dataset the study requires distributed simulation. Hence, a new Metric Annealed (MA) Federated Learning is proposed with Flower Framework to represent real-life scenarios and prioritize security. To address the resource constraints, hierarchical CNN-BiLSTM client model is introduced with an adaptive probabilistic client selected MA averaging aggregation strategy. The hierarchical model offers an optimized number of parameters enabling the simultaneous training of binary and multi-classification heads.

The proposed model achieved an accuracy of 98.8% in binary classification and 81.1% in multiclass classification while having low false positive rate, achieving efficient performance and resource utilization compared to state-of-the-art DL and FL models in precision, recall, and F1-score. Hyperparameter tuning through systematic tuning identified optimal temperature decay and weight parameters for energy-based evaluation . Comparative results demonstrate that the MAFL model offers superior scalability, privacy preservation, and detection performance compared to centralized IDS approaches. The research establishes Metric Annealed Federated Learning (MAFL) as a viable and efficient framework for secure intrusion detection in distributed IoT networks.

Keywords: MAFL, Federated Learning, NIDS, Simulated annealing, IoT

Nomenclature

- **FL** - Federated Learning
- **IoT** - Internet of Things
- **IDS** - Intrusion Detection System
- **FedAvg** - Federated Averaging
- **FedProx** - Federated Proximal
- **SA** - Simulated Annealing
- **DWA** - Dynamic Weighted Aggregation
- **SVM** - Support Vector Machine
- **ML** - Machine Learning
- **DL** - Deep Learning
- **RNN** - Recurrent Neural Networks
- **LSTM** - Long Short-Term Memory
- **CNN** - Convolutional Neural Networks
- **PCA** - Principal Component Analysis
- **AE** - Auto-Encoder
- **PCC-CNN** - Principal Component Clustering Convolutional Neural Network
- **CIFG** - Coupled Input-Forget Gate
- **MSE** - Mean Squared Error
- **FPR** - False Positive Rate
- **FNR** - False Negative Rate
- **AUC** - Area Under Curve
- **TP** - True Positive
- **FP** - False Positive
- **TN** - True Negative
- **FN** - False Negative
- **DDoS** - Distributed Denial of Service
- **DoS** - Denial of Service
- **XSS** - Cross-Site Scripting

- **SQL Injection** - Structured Query Language Injection
- **BOT** - Botnet Attack
- **UNSW-NB15** - UNSW Network Benchmark 2015
- **CICIDS2017** - Canadian Institute for Cybersecurity Intrusion Detection Dataset 2017
- **NSL-KDD** - NSL Knowledge Discovery in Databases
- **KDD** - Knowledge Discovery and Data Mining
- **HDD** - Hard Disk Drive
- **GPU** - Graphics Processing Unit
- **FLOP** - Floating Point Operations
- **FLWR** - Flower
- **MAFL** - Metric Annealed Federated Learning
- **SHAP** - Shapley Additive Explanations
- **LIME** - Local Interpretable Model-agnostic Explanations

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The adoption of Internet of Things (IoT) technology has expanded rapidly over the past decade. Kevin Ashton first used the term "Internet of Things," or IoT, in his presentation, and from there, IoT has spread into various sectors. Various applications in schools, offices, health care systems, multiple industries, and transportation field have been noted by Vishwakarma et al. [48]. As of 2024, the global industrial Internet of Things (IoT) market was valued at $194.4 billion and is projected to reach $286.3 billion by 2029, growing at an annual rate of 8.1% [59]. Its distributed nature promotes efficiency, scalability, and robustness of operations, makes it an essential component of the modern ecosystem. However, the proliferation of IoT and related distributed systems has amplified security risks, as the frequency and sophistication of cyber-attacks continue to escalate. These attacks include downloading and spreading malware, command injection [42], Denial of Service (DoS) [12], key extraction attacks, buffer overflow, runtime hardware trojans, IoT Routing Attacks, Bruteforce, and Ransomware. Through these attacks, sensitive data is being compromised, which can affect individuals both socially and financially.

Primarily, IoT networks are vulnerable because they rely heavily on resource-constrained devices, dynamic growth and contain a wide variety of heterogeneous technologies; all of which make it hard to maintain strong, uniform security. Thus, preventive measures, such as Intrusion Detection Systems (IDS), have become a vital necessity. IDS can be divided into two groups: real-time IDS, which detects intruders as they occur, as explained by Liu et al. [12], and Post Intrusion Detection Systems (Post-IDS), which detect intruders after they have already gained access to the network [42]. However, Liu et al. [12] also presented several challenges faced by conventional IDS, including scalability, resource limitations of IoT devices, centralized data processing approaches, and the poor performance of signature-based detection in evolving attack patterns. Moreover, Zipperle et al. [44] demonstrated that with growing network complexity, the exponential increase in data volume makes it difficult to process due to the computational scarcity of IoT devices.

Moreover, conventional IDS utilizes centralized data processing techniques which further strains the already limited resources of IoT devices. In [12], it was further

stated that with evolving attack patterns signature-based IDS performs very poorly. To address these challenges Federated Learning has emerged as a promising solution. Its ability to enable decentralized, privacy-preserving model training makes FL a particularly attractive approach for developing scalable and efficient intrusion detection systems in IoT networks. FL has emerged as a viable solution for decentralized learning. However, it also presents its own limitations that require further investigation.

1.2 Rationale of the Study or Motivation

Ensuring data security and privacy has become critical priorities in today's interconnected world. Although Intrusion Detection System (IDS) have advanced significantly, they continue to face limitations when deployed in IoT environments. This has established IDS research as a critical focus area, particularly in IoT edge devices. Despite numerous experiments which have been conducted, the progress in cyber-attacks detection with conventional IDS fails to identify the types of attacks accurately [12], [44]. Current Data-Driven approaches show high accuracy in binary detection. However, the privacy concern, type detection, and other IoT limitations still remain. The introduction of Federated Learning (FL) in data driven field opens new door for IDS to address these issues. Researchers have been tirelessly exploring alternative approaches, such as incorporating lightweight Machine Learning (ML) models for diverse threat detection combining Federated Learning (FL) to focus on the resource constraints and data privacy issues [54]. Nevertheless, new challenges emerge with every advancement, including the aggregation of parameters from training on non-Identical and Independent Distributed (non-IID) data and the increased data dimensionality, necessitating data compression techniques [27], [46].

Additionally, the need to transmit model parameters between central servers and edge devices results in high communication overhead [8], [33], [37], [46]. Furthermore, studies [7] showed that, specific tweaking and adjustments to make the system more efficient. The authors used structured and sketched updates to reduce communication expenses by 98% compared to traditional approaches. Although several experimental studies approached to lessen the tradeoff between accuracy and computational cost, none came up with a concrete solution or addressed any limitations. This highlights the pressing necessity for optimized, resource-aware IDS solutions that can effectively balance detection accuracy and classification performance.

1.3 Problem Statement

The widespread adoption of IoT has significantly expanded the attack surface of digital systems, resulting in a sharp increase in the frequency and sophistication of cyber-attacks. Threats such as Distributed Denial of Service (DDoS), malware injection, brute force - not only compromise sensitive data but also jeopardize the stability of critical infrastructures. This makes intrusion detection a crucial research focus in IoT edge environments. Conventional Intrusion Detection Systems (IDS) rely primarily on centralized data processing and signature-based methods. While effective against known threats, these approaches face critical limitations when de-

ployed in IoT networks due to scalability challenges, constrained computational resources, and poor adaptability to evolving attack patterns. Additionally, although recent machine learning (ML) and deep learning (DL) techniques have achieved high accuracy in binary intrusion detection, they perform poorly in multiclass classification tasks. These limitations are compounded by challenges such as class imbalance in datasets, high computational costs, and the inability of IoT edge devices to process large-scale intrusion datasets efficiently.

Federated Learning (FL) has emerged as a promising paradigm to mitigate some of these challenges by providing decentralized model training while preserving user privacy. However, FL itself introduces new barriers, including performance degradation on non-Identical and Independent Distributed (non-IID) data, communication overhead from frequent parameter exchanges, and the need for model optimization to operate within the limited resources of IoT devices. All of these lead to the following research questions:

1. What is the state-of-the-art performance of IDS in IoT networks for both binary and multi classification tasks?
2. What strategies can be applied to effectively address the data imbalance and scarcity in IoT network intrusion datasets?
3. To what extent can the Federated Learning aggregation techniques mitigate the resource constraints inherent in IoT-based IDS?
4. Are data driven solutions reliable for detecting and enhancing data security of IoT edge devices?
5. Developing Federated Learning-based Network Intrusion Detection Systems for IoT edge devices
   (a) Architectural inefficiency requiring separate models for binary and multiclass classification.
   (b) High client overhead due to uniform participation strategies
   (c) Lack of fairness in client selection leads to model bias and straggler issues.

   These limitations hinder practical deployment in resource-constrained IoT environments.

1.4 Objective

The study aims to overcome the limitations of existing IDS approaches by enhancing scalability, minimizing computational and communication overhead, and improving detection accuracy in both binary and multiclass classification tasks. The specific objectives of this research are as follows:

1. Derive thorough data insights by visualizing and exploring benchmark IDS datasets and to apply pre-processing techniques that enable reliable model training while addressing issues of data scarcity and class imbalance.

2. Designing a two-headed IDS model capable of jointly performing binary and multiclass intrusion detection tasks, with attention to reducing false positives and improving compromised node detection.

3. Develop a Federated Learning (FL) client-side deep learning model optimized for resource-constrained IoT devices.

4. Benchmarking the proposed IDS framework against state-of-the-art deep learning (DL) and machine learning (ML) models.

5. To develop and evaluate a novel client selection method based on simulated annealing to enhance training efficiency, particularly under non-Identical data conditions.

The foremost objective of this study is to come up with an efficient and lightweight Network Intruder Detection System along with data insights. Particular attention is placed on aiming to identify ML algorithms that effectively detect anomalies while maintaining low false positive rates. An efficient approach is sought to minimize resource consumption, with an emphasis on scalability, communication overhead reduction, and data processing efficiency. To address privacy concerns, FL will be employed to avoid sharing sensitive data. Additionally, the study aims to tackle non-IID data distribution challenges, ensuring the proposed IDS solution is applicable in such scenarios. Achieving more than half of the targeted objectives will be considered as a success of this study.

1.5 Methodology in Brief

This experimental research was conducted in three phases. Firstly, a comprehensive qualitative analysis of state-of-the-art NIDS research study was conducted to find out gaps, limitations, and challenges. Secondly, an in depth work done on benchmarking the IDS datasets available for past decade. They were pre-processed and features are studied meticulously. Finally, a novel FL-based NIDS approach is proposed and its performances are been compared against state-of-the-art machine learning and deep learning methods.
The first phase of this study was dedicated to a comprehensive review and qualitative analysis of existing Intrusion Detection System (IDS) research in the context of IoT. This ph ase focused on identifying the research gaps, limitations, and challenges associated with conventional IDS. After examining the strengths and weaknesses of state-of-the-art methods, this analysis provided the foundation for designing an improved framework that addresses unresolved gaps. The second phase focused an intricate study on benchmarking NIDS datasets that have been widely used over the past decade. These datasets went through an extensive pre-processing, including feature selection, normalization, and class balancing. Furthermore, careful data analysis was conducted to observe insights of attacks and its distributions. Based on the insights several class balancing methods were explored in order to address these gaps. This phase ensured the reliability of training data and enabled a robust experimental setup for model development.
The final phase introduced a novel Federated Learning (FL) based Network Intrusion Detection framework designed to meet the privacy, scalability, and efficiency require-

ments of IoT edge devices. A lightweight deep learning model with a two-headed architecture was developed to jointly handle binary and multiclass classification tasks. Additionally, a client selection mechanism based on simulated annealing (SA) was incorporated to improve training efficiency in federated environments. The proposed approach was systematically bench-marked against state-of-the-art machine learning (ML), deep learning (DL) and aggregation techniques of federated learning using standard evaluation metrics, including accuracy, precision, recall, F1-score, false positive rate and false negative rate.

1.6 Scopes and Challenges

This study primarily focuses on enhancing Intrusion Detection Systems (IDS) for IoT edge devices and distributed systems by addressing key challenges such as resource efficiency, privacy protection, and scalability. The implementation of Federated Learning (FL) in the context of Internet of Things (IoT) environments faces numerous technical challenges, including the management of high-dimensional and heterogeneous datasets, and the lack of optimization in communication. To address these challenges, the study explores optimizing FL-based frameworks and evaluates various Machine Learning (ML) models, including Recurrent Neural Networks (RNN), Auto-Encoders (AE), Support Vector Machine (SVM), Long-Short Term Memory (LSTM), Principal Component Clustering Convolutional Neural Network (PCC-CNN) etc. Critical issues of security and privacy call for strong defenses against hostile attacks on FL models and sensitive data. For optimization of computational and resource efficiency, this study additionally focuses on effective parameters and aggregation strategies of Federated Learning. Another challenge is scalability, especially when implementing IDS across a large number of IoT devices while maintaining steady performance as data quantities increase. Auto-encoders (AE), which can give reduced feature representations and remove variable data length, are used to address the heterogeneity of IoT data, which arises from a variety of device kinds, connection protocols, and environmental factors. Although we did find some validated datasets that could prove useful for our study such as the Darpa, and ISCX-2012, these datasets turn out to be very old. But overtime new datasets have emerged that effectively captures the real-world scenario for IoT. Effective communication protocols essential for maintaining low-latency and secure data exchange in FL-based IDS are thoroughly analyzed to ensure robust and scalable solutions tailored for real-world IoT environments. Validation and real-world testing are complex due to difficulties in replicating dynamic environments and realistic attack scenarios.

Chapter 2

Related Work

2.1 Preliminaries

2.1.1 Federated Learning (FL)

Federated Learning (FL) is a decentralized machine learning paradigm where multiple clients collaboratively train a global model without sharing their raw data. Each client computes local model updates using its private data, which are then aggregated at a central server to update the global model. This approach enhances data privacy, and is particularly suitable for heterogeneous and distributed environments such as the Internet of Things (IoT). FL faces challenges including handling non-IID data distributions, limited client resources, and ensuring fairness in client participation [43] and faces communication overhead issue.

2.1.2 CNN

Convolutional Neural Networks (CNNs) are deep learning architectures designed for spatial data analysis like feature maping. Its convolutional layers extract hierarchical features following pooling layers for dimensionality reduction, and use fully connected layers for classification. In the context of FL and IoT, CNNs are widely used for anomaly detection and intrusion detection, as they can efficiently capture spatial correlations and local patterns in network traffic or sensor data.

2.1.3 Bi-LSTM

Bi-LSTM networks extend traditional LSTMs by processing sequential data in both forward and backward directions, capturing past and future dependencies simultaneously. This property of Bi-LSTMs is highly effective for time-series analysis, such as modeling network traffic patterns or user behavior in IoT systems. In intrusion detection, Bi-LSTM can learn temporal correlations in attacks, improving prediction accuracy over standard unidirectional models.

2.1.4 Intrusion

Intrusion detection involves identifying unauthorized or malicious activity within a network or system. Traditional approaches rely on signature-based detection, while modern methods use machine learning to detect anomalous patterns in network

traffic or device behavior. Combining FL with deep learning models like CNN and Bi-LSTM enables privacy-preserving, distributed intrusion detection, which is critical in large-scale IoT networks with heterogeneous devices.

2.2 Review of Existing Research

Numerous studies on IoT and distributed systems have been conducted in recent times as the growing uses of tinyML and Artificial Intelligence revolutionize the era. A survey conducted by Vishwakarma [48], presented various advancements in IoT and its application in multiple services with the incorporation of blockchain, wireless sensor network (WSN), machine learning (ML), and big data analysis. General working process of each platform and their unique strengths and weaknesses. Emerging technologies like, AWS IoT provide secure, bidirectional communication between devices and the cloud whereas, Google Cloud IoT uses its diverse machine learning capability to provide IoT-related solutions which prove to be highly efficient. Promising communication protocols like IPv6, 6LoWPAN, MQTT, CoAP, Bluetooth, ZigBee, and Wi-Fi play an important role in this field. Current challenges like resource and energy constraints, interoperability, privacy and security risks, and scalability were provided.

The network environment changes and attack variants quickly, and novel attacks emerge constantly. IDSs have been facing false alarm issues for a long time. Many DL effective models have been used as they can detect unknown attacks. Liu et al. provided an extensive study of several representations of different attacks and the perspective of learning methods along with the different aspects of ML, and DL [12]. Unsupervised ML models like ANN, SVM, KNN, NAiVE BAYES, LR, Decision Tree, K-means, Ensemble Learning, etc are deployed on the datasets. These basic models found computational, non-linearity problems in IDSs, as distributed systems and IoTs are IID, continuous real-time data. Study conducted by Kim et al [27] on One-Class SVM, Isolation Forest (IF), and Local Outlier Factor (LOF) applied to the ADFA-LD dataset. Various feature extraction methods, including Doc2Vec and RNN-based AE, were used to convert variable-length logs into fixed-size vectors. IF with RNN-DAE achieved the highest AUROC accuracy (0.8708), but the study was limited to system call logs, signifying a requirement for broader features like API calls and network data.

Another approach using the Data Provenance-based Intrusion Detection Systems (PIDS) was proposed by Micheal et al. [44], to detect Tcpdump logs [42]. Out of 10 million logs, they found that direct access accounted for 39.1%, downloading activities accounted for 34.2%, and binary injections for 21%. Attackers exploited BusyBox to download malware, accessed the shell to execute commands, and injected binary codes into writable files to establish backdoors and unauthorized communication. However, the study focuses on, commonly observed behaviors rather than less common Persistent Threats (APT), where traditional IDSs fail due to high false positive rates and the inability to detect stealthy attacks. They proposed graph-based model representation which shows the flow of information and uses Data Provenance Capture (DPC) for system-level data collection. Despite the advantages, DPC faces challenges such as runtime overhead, availability, privacy, and fault tolerance. To reduce overhead, techniques like bit-compression, simplification,

policy-based reduction, and edge-grouping were used, but excessive compression may result in the loss of critical information. Scalability becomes an obstacle, with the increasing number of nodes.
Traditional anomaly detection techniques are widely studied by the researchers, including LSTM, Replicator Neural Networks (RNN), Recurrent Neural Networks (ReNN) combined with Feed-Forward Neural Networks (FFNN), and clustering-based approaches.

2.2.1 Dataset

Existing benchmark datasets such as LANL and CERT fail to reflect real-world environments. In addition, privacy concerns limit the sharing of datasets sharing progress in IDS classification. IDS methods and Data sources are separately divided into two parts [12]. Starting with misuse-based, which achieve low false positives but struggle with scalability and novel attacks. On the other hand, anomaly-based, which can detect new threats but often generate high false positives. Data sources are either host-based, relying on system logs and OS data but blind to network traffic, or network-based, which analyze traffic but depend on the reliability of network infrastructure. DOS works well with flow data, as flow data is familiar with

Table 2.1: Considered Dataset for Clients

Dataset
DARPA 1998 [1]
KDD99 [3]
NSL-KDD [4]
UNSW-NB15 [35]
CTU-13 [5]
CIC-IDS2017 [11]
ToN-IoT [24]
IoT-23 [17]
N-BaIoT [10]
Edge-IIoT [31]

the network environment. Flow data scores are poor in terms of heterogeneity. On the other hand, packet-based data does not work well with DOS as it involves a large volume of data, but it has advantages like precisely detecting the source of the attack, with communication parameters for U2L and R2L attacks; most importantly, it is applicable in real-time applications. Applying SVM on HTTP data shows a precision of 99.6% [12]. Additionally, flow data sometimes acts as white noise. CNN shows 99.13% accuracy with payload analysis data. Payload data is unstructured and struggles with encrypted data. A combination of CNN, LSTM, and RNN showed 98.44% precision [15], [44].

2.2.2 Existing Data Driven Approaches

Traditional Machine Learning models are unable to deal with data flow. To address this, simple and efficient feature engineering can be used. However, considering the networking protocol's parameters leads to a high false alarm rate. Traditional classifiers like SVM and KNN on average show 99.45% accuracy and a false rate of 1.57% on the KDD99 dataset. Additionally, feature reduction methods such as PCA can help improve the performance. However, feature engineering depends on domain knowledge and can become a bottleneck for detection efficiency. Deep learning models learn features directly from the data. A three-layer CNN model outperforms ResNet and GoogLeNet, achieving 91.14% accuracy. However, deep learning tends to have poor performance in the case of unbalanced datasets like KDD99 [15]. A survey by Merlino and Allegra [47] shows that, most of the traditional ML algorithms perform poorly as the number of features increases.Frameworks like Transfer Learning and Federated Learning, which can communicate from one device to another, have promising architectures in optimizing the use of limited resources.

Energy-driven data featuring client activity analysis serves as notable results in the detection of attacks such as unauthorized access, compromised devices, or malicious activities [47]. Notable power consumption of the attacked client places significant spotlight on a novel approach to IoT security. As the power sensor is a time series dataset, it has warping costs in Deep Neural Network fields. Energy consumption data, in the form of time series logs, plays a critical role in IoTbased IDS. To improve functionality and lower time complexity, several researches have been done on Dynamic Time Warping (DTW) in time series classification. Mobarakeh [25] proposed a kernel layers at the beginning of the artificial neural networks and trained them in every kernel based on DTW alignment. It improves non-linearity and enhances classification performance. The study showed that where signals behave similarly but in different situations, Minkowski distance can solve this issue. With the ECG5000 and ECG200 datasets, they achieved 91% accuracy, outperforming Inception, ResNet, and typical non-deep methods [25] . DTW does not require equal-length data, which helps with IoT heterogeneity from various devices and improves model training and classification. Incorporating feature reduction methods like AutoEncoders (AE) and Principal Component Analysis (PCA) can further improve the efficiency of processing large time series datasets. Other studies on AutoEncoders and PCA have been seen as beneficial approaches as they show significant reductions in computational processing [15], [47]. Higgins et al. [36]explored IDS from a different angle, pointing out a preventive measure using Moving Target Defense (MTD) against False Data Injection Attacks (FDIA) on IEEE-14 Bus test systems. The study demonstrates how dynamically modifying network topology, such as switching transmission paths and incorporating admittance perturbation, can prevent intrusion. Attackers unaware of these changes cannot effectively carry out attacks. While promising, the study is based on simulations rather than real-world implementations. Additionally, applying MTD consumes significant resources, which limits its feasibility in resource-constrained environments.

2.2.3 Federated Learning in IDS

Intrusion Detection Systems (IDS) are progressively utilizing Federated Learning (FL) methodologies to elevate security strategies in decentralized and resource deficient settings. A revised Federated Averaging (FedAvg) was introduced to improve the communication efficiency of FL. Enabling multiple local model updates before transmitting aggregated parameters to a central server, this algorithm reduced communication costs significantly. It is compared with Federated Stochastic Gradient Descent (FedSGD) which computes and sends gradients after one global iteration. FedAvg reduces communication rounds by 10–100×, achieving 99% accuracy on MNIST in 300 rounds compared to over 1000 for FedSGD [7]. It also proved robust to non-independent and Identically Distributed (non-IID) data and scalable across datasets, supporting thousands of clients. This minimizes privacy risks by keeping data on the devices and only sharing model updates with the server. Despite its effectiveness in general FL scenarios, the study did not address the challenges posed by high-dimensional network traffic data in IDS applications.

Isolation Forest is used in real-time anomaly detection. Designing a common iForest for all clients is crucial for IID datasets. Xiang et al modeled FLiForest, an FL integration of simple iForest to provide real-time security to an IoT-edge continuum. [49]. Locally, iForest has been trained based on collecting large amounts of multimedia data from the end devices. Transferring the global weights in each iteration, every client creates a complete isolation tree. With multimedia datasets, FLiForest outperforms other isolation forests, indicating that FL integration is impactful in anomaly detection with privacy. Lu et al. [46] discussed some solutions for privacy and security in FLForest, which utilizes isolation forests to detect and exclude malicious updates, enhancing Byzantine-robustness and trustworthiness in FL models. Blockchain-Based FL enables secure global data sharing and privacy protection using blockchain for decentralized FL model training. 2DP-FL incorporates differential privacy by adding noise during model training and global distribution to ensure privacy and convergence. Sophisticated systems or attacks like DDoS have more diverse features, hence the global model faces challenges like sparse trees and dimensionality.

Increasing identical nodes in the IoT system are considered as the same category which makes the FedAVG vulnerable. Therefore, Zhang et al. [43] proposed FedGroup to detect attack attempts for IoT devices. Consisting of 10 devices dataset for their research was particularly considered for IoT parameters also no real-time learning has been used. To maintain the identical nodes, a group master handles the group and focuses on the same type of features, only the updated weights from the group masters are transferred to the central server. Compared with several local and central models of Traditional ML and FL, FedAVG, and FedGroup show the same accuracy, but still FedGroup costs less computation than FedAVG and shows equivalent accuracy (99.91%) where Traditional ML surpluses the cost.
A wide range of studies [6], [8], [26], [33], [37], [43] on communication overhead reduction and resource constraints using various parametric optimizations were carried out utilizing scarcity techniques such as minimizing redundant data transmission, reducing overall resource usage [6], [46] put forward a quantized representation of model weights using fewer bits.[29] used Ternary Federated Averaging (T-FedAvg), a quantization-based protocol to optimize communication in FL, low-rank projec-

tion, and the Communication-Efficient Federated Optimization (CEFO) method to compress model parameters and reduce communication bandwidth. It also included some model performance challenges like performance bias and imbalance, generalization, convergence and stability, and knowledge sharing. [26], [54] aggregated local updates at intermediate cluster nodes before transmission to the central server, reducing communication overhead while optimizing energy usage. [37] Dynamic Weighted Aggregation Federated Learning (DAFL), a dynamic weighted aggregation method to prioritize local models based on detection accuracy and sample size, filtering out underperforming models to prevent degradation of the global model. [33] also used a dynamic weighted aggregation method. [8] used a Coupled Input-Forget Gate (CIFG) recurrent neural network, which reduced the number of parameters per cell by 25% compared to traditional LSTMs.

Study [8] on Federated CIFG achieved 15.8% top-1 recall on client caches versus 15.0% for server-trained CIFG and 12.5% for the n-gram baseline. Server CIFG is trained on centrally collected and anonymized server data which requires data logging and centralized storage meaning more resources. Fed CIFG improved live user impression recall by 1% over the server-trained model. The study incorporated 1.5 million clients in the training process, collectively processing 600 million sentences over 3000 training rounds ensuring scalability with significant improvements in model performance and user experience. Another study [33], the novel approach performs to a 90% reduction in communication costs, validated through experiments on CIFAR-10 datasets [45] used the NSL-KDD Dataset (Network traffic data focusing on realistic attack scenarios), and a Car-Hacking Dataset (Simulates real-world automotive attacks, such as spoofing and DoS, for autonomous vehicles). It used two test beds with NVIDIA Jetson Xavier (server) and Raspberry Pi 4 (clients). Firstly, the Logistic Regression model was used which resulted in an accuracy of 96.82% (Federated) vs. 46.54% (Centralized) in the NSL-KDD dataset and an accuracy of 93.56% in the Car-Hacking dataset. Then it was tested with the PCC-CNN model which had an accuracy of 97.08% in the NSL-KDD dataset and 99.93% accuracy in the Car-Hacking dataset.

Table 2.2: Comparison of Federated Learning Methods

FL Methods	**Advantages**	**Disadvantages**
FedAVG [7]	Reduces communication rounds, scalable, some privacy features	Requires more computation on clients, not ideal for resource-limited devices, slow convergence in heterogeneous data
FLForest / FLiForest [46] [49]	Enhancing Byzantine robustness and trustworthiness, Handles Non-IID Data, Low Overhead	Dependency on Dimensionality Reduction, False Positives in Detection, Requires Threshold Tuning, Complexity of Implementation
FedGroup [43][46]	Reduces instability caused by non-IID data, better model performance, 20% better than FedAVG	High communication and computational overhead from clustering, scalability issues with large-scale systems, complex deployment requirements
FedNorm [46]	Improved performance in heterogeneous data, enhanced generalization, applicability to real-world scenarios	Limited to single-modality clients, increased computational complexity, dependency on modality information
T-FedAVG [46]	Optimized communication, better heterogeneous data performance, reduced weight divergence	Higher computational overhead, limited scalability, potential loss of specificity
Blockchain-Based FL [46]	Enables secure global data sharing and privacy protection using blockchain, Enhanced Security and Trust, Incentive Mechanism	High Computational and Communication Overhead, Consensus Delays, Storage Requirements, Energy Consumption, Complex Implementation, Vulnerable to Blockchain-Specific Attack
2DP-FL [46]	Privacy Preservation, Resilience to Attacks, Experimental Validation, Low Overhead	Increased Complexity, Dependence on trusted shuffler which if compromised could lead to privacy breaches, Limited Scope of Attack Mitigation
DAFL [37]	Improved Global Model Performance, Communication Overhead Reduction, Better Adaptation to Non-IID Data	Potential Bias in Global Model, Limited Performance in Low-Quality Data Scenarios, Threshold Sensitivity

Table 2.3: Comparison of Federated Learning Aggregation Strategies

Feature / Issue	**FedAvg [38]**	**FedProx**	**[29]FedAdam / FedYogi / FedGrad**
Handling high α	Bad with high α (degree of data heterogeneity). [29]	Handles high α	Handle high α and robust / Slow
Client drops	Do not drop clients	Do not drop clients	Do not drop clients
Non-convex problems	Bad	Better than Adam	Good with client heterogeneity
Default learning rate	1.0 [38]	Learning rate follows variance	Tuned penalty coefficient
Gradient privacy	Gradient can leak info		
Non-IID data	Provides biased AVG, low generalization	Handles variance based on feature importance	Partial / controlled update on heterogeneous clients [29]
Client drift	Have client drift and adaptability [20]	Good with client heterogeneity	Client penalized for drifting [29]
Computation	Better than SGD, less computation	Better convergence, simpler than FedAvg	Limited computation power, cross-device
Assumptions	L-smooth, bounded variance and gradient [29]		Not all clients solve local subproblems

2.3 Summary of Key Findings

Throughout the study, various advancements of FL-IDS have been found. Yet, many challenges are left to be addressed and future in-depth works are required.

2.3.1 Lackings of IDS and Data Driven Approaches

1. Traditional IDS have high false positive rate, can't detect novel attack patterns, signature based often can't generalize.

2. Even with decentralized learning, resource constraint in the IoT environment remains a core factor, further study is needed to find a balance between security and resource utilization. Due to unbalanced dataset, model training proves difficult.

3. ML/DL models (SVM, KNN, ANN, RNN, CNN, AutoEncoders) improve anomaly detection but face issues with non-linearity, real-time data, and dataset imbalance.

4. Dataset availability is a bottleneck privacy restrictions hinder reproducibility in IDS research.

5. Non-IID data distributions in FL can be challenging as aggregation of model parameters needs acute adjustments for seamless merging

6. FL introduced a better privacy preserving method of training where only model parameters are shared instead of the actual data. Also, it challenges include high-dimensional traffic data, communication overhead, statistical heterogeneity, and resource constraints.

7. Reconstruction attacks on the shared parameters allow for the reverse-engineering of the actual data from the parameters being sent by the clients to the server for aggregation, given the attacker has enough context.

2.3.2 Major Challenges

1. Ensuring data security

2. False positives remain a critical issue for anomaly-based IDS.

3. Data scarcity and imbalance limit training effectiveness.

4. FL still struggles with non-IID distributions, convergence stability, and communication costs.

5. There is a gap in hybrid approaches that combine feature engineering efficiency with DL's adaptability.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

Hardware Configuration:

- Processor: Intel Core i7 14th Gen (20 cores, 28 threads)
- GPU: NVIDIA GeForce RTX 4070 Ti Super
- Development Environment: Python 3.11 with PyTorch 2.5.1+cu121
- Cloud Resources: Kaggle GPU runtime for pre-processing
- FL Framework: Flower v1.18 for distributed training

3.2 Societal Impact

The rapid adoption of IoT into day to day life makes cyber security a matter of significant societal concern. A robust Network Intrusion Detection System (IDS) not only protects data but also ensures trust in the service providers. The impacts of this study can be analyzed across several social dimensions. At the personal level, individuals increasingly rely on IoT devices in homes and workplaces. A compromised IoT edge device can lead to identity theft, financial fraud or unauthorized surveillance. By enhancing Network IDS performance, this research contributes to safeguarding personal privacy and social trust in technology-driven ecosystems.
The primary societal benefit of FL-NIDS lies in its ability to enable collaborative cybersecurity while preserving data privacy. According to recent surveys, federated learning allows multiple organizations to jointly train intrusion detection models without sharing raw network traffic data [19]. This addresses growing privacy concerns from both regulatory bodies and civil society groups advocating for stronger data protection [41]. The technology enables what was previously impossible: organizations can benefit from collective intelligence about cyber threats while maintaining complete control over their sensitive data.
Small and medium enterprises (SMEs) particularly benefit from this democratization of cybersecurity capabilities. Traditional centralized NIDS require substantial

infrastructure and expertise that many SMEs cannot afford. FL-NIDS enables these organizations to participate in and benefit from enterprise-grade threat detection without the associated costs or privacy risks [14]. This helps level the playing field in cybersecurity, where previously only large corporations could afford comprehensive protection. FL-NIDS naturally aligns with emerging data protection regulations such as GDPR and CCPA. By keeping data localized and only sharing model parameters, organizations can maintain compliance while still benefiting from collaborative learning [28]. This is particularly important for sectors like healthcare and finance, where data sharing is heavily regulated. The technology enables cross-border collaboration in cybersecurity without violating data localization laws, which is crucial as cyber threats are increasingly global in nature [22].
In the healthcare domain, particularly in Internet of Medical Things (IoMT), institutes involved with personal health data face severe problems. Such as, need of strong data security, the collaborative learning of Machine Learning in order to serve the data scarcity in the health sector. Due to this, Federated Learning (FL) emerges as a great adoption to ensure data sharing with privacy. The U.S. Department of Health and Human Services reported breaches affecting 500+ individuals, with the majority originating from network servers[58]. Studies show that healthcare data is one of the most frequently targeted sectors, with breaches caused by both internal and trends indicate that both the frequency and financial impact of such breaches are expected to rise in the future[23], [60], [61].
FL-NIDS faces new societal challenges like distributed mechanism of FL makes it difficult to establish trust among participants. Stakeholders are heavily relied on the integrity of other participants without being able to verify their data or intentions directly [16], [30]. It requires trusted governance frameworks and trust mechanisms. This is particularly important in critical infrastructure where false positives could disrupt essential services. The proposed IDS framework strengthens security in these sectors by reducing susceptibility to sophisticated cyber threats, ensuring more secure operations in healthcare and other IoT-enabled services. At bottom, offers a foundation for more resilient IoT infrastructures for sustainable, secure integration of IoT technologies into daily life.

3.3 Environmental Impact

The environmental impact of FL-NIDS is not that simple to understand and solely depends on the very details of the implementation process. Given Federated Learning is often considered as a more sustainable and scalable training model compared to existing centralized training, recent studies show alternative perspective, when efficiency and optimization while designing the system are not considered.

Carbon Footprint Analysis
A comprehensive study of federated learning's carbon footprint, conducted by Qiu et al.[41],where they discovered that poorly configured FL systems can generate carbon emissions several orders of magnitude higher than centralized training. However, their research also revealed that FL can outperform centralized methods, particularly when deployed on energy-efficient edge devices powered by renewable energy sources. The key factors determining environmental impact include hardware efficiency, number of communication rounds, and the carbon intensity of the local

power grid. Edge devices typically consume only 7.5-15W compared to 250-300W for data center GPUs [21], representing a dramatic 95% reduction in power consumption per device. This substantial difference translates to meaningful energy savings across distributed networks. Experiments[40] show that communication energy can dominate emissions for small datasets like CIFAR10, while for larger datasets like ImageNet, the energy cost of training outweighs communication.

Sustainability Strategies
Recent work on "Green Federated Learning" proposes several strategies to minimize environmental impact [13]. These include adaptive client selection based on energy availability, model compression techniques to reduce communication overhead, and scheduling training during periods of renewable energy surplus. The BEFL framework demonstrates a 28.2% decrease in overall energy consumption compared to baseline FL approaches while maintaining model accuracy [32]. Edge devices in FL systems typically don't require dedicated cooling, providing an inherent advantage. However, the distributed nature of FL means that the aggregate energy consumption across all participating devices must be considered. Overall, FL offers a more sustainable alternative although hardware efficiency and aggregation strategies are essential to optimize CO2 emissions.

3.4 Ethical Issues

This study acknowledges several ethical considerations associated with developing a FL-based IDS. First, data privacy must be strictly maintained to ensure that no sensitive information is leaked through gradient or weight inspection during model training. Fairness is also a key concern, requiring the system to avoid bias against clients or devices with lower computational capabilities or limited data quality. Maintaining transparency is essential so that model decisions remain interpretable within critical security contexts, supporting trust and accountability. Given that Network Intrusion Detection Systems (NIDS) could potentially be misused for surveillance or unauthorized monitoring, ethical safeguards and responsible deployment guidelines are necessary. Finally, accountability must be ensured by clearly defining responsibility for model errors, including false positives and false negatives.

3.5 Project Management Plan

The research was conducted over twelve months in three phases. The initial phase focused on planning, paper selection, and a comprehensive literature review to identify research gaps, define the problem scope, and formulate hypotheses. This was followed by the submission of the Pre-Thesis 1 report. In the next stage, relevant benchmark datasets such as UNSW-NB15 and CIC-IDS2017 were collected, pre-processed, and analyzed to establish a reliable experimental foundation. Data pre-processing included feature selection, normalization, and class balancing. Subsequently, an experimental setup was created to test and benchmark traditional IDS models and to design a taxonomy of existing IDS approaches. Implementation proceeded in two stages: Stage 1 focused on centralized IDS model development and baseline evaluation, while Stage 2 extended the model using Federated Learning to

enable decentralized and privacy-preserving training. Model training, optimization, and evaluation were conducted using standard performance metrics. The final phase focused on final hyperparameter tuning, result and evaluation, compilation documentation, and preparation of the final thesis report. This structured plan ensured continuous progress and integrity among the team members.

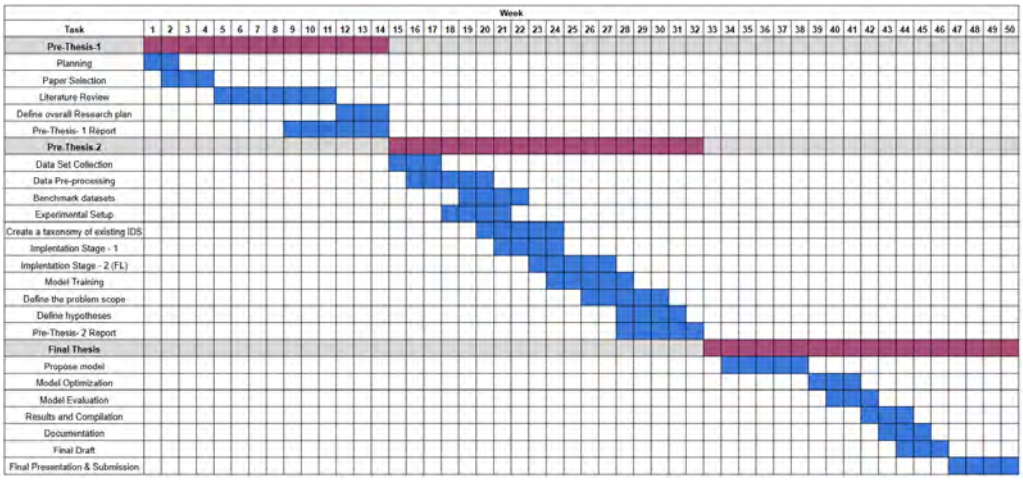

Figure 3.1: Planned workflow and research timeline

3.6 Risk Management

The development of the proposed IDS with Federated Learning introduces several risks that must be carefully managed to ensure successful outcomes. Each risk has been evaluated in terms of its potential impact and likelihood, with corresponding mitigation strategies outlined below.

Table 3.1: Risk Assessment and Mitigation Strategies

Risk	**Impact**	**Likelihood**	**Mitigation Strategy**
Model not converging	High	Medium	Training process begins with a baseline Federated Averaging (FedAvg) method, with complexity gradually increased to ensure stable convergence.
Insufficient computing resources	High	Medium	Utilize institutional GPU resources and cloud-based credits where necessary, optimize batch sizes to ensure efficient resource usage.
Poor performance on imbalanced data	High	Medium	Use weighted loss functions and resampling techniques such as SMOTE to improve classification across underrepresented classes.
Aggregation algorithm instability	Medium	Medium	Manage through extensive hyperparameter tuning and ablation studies to ensure robustness and stability of the aggregation process.
Timeline delays	Medium	Medium	Buffer period incorporated into project plan, core research objectives prioritized to minimize effect of unforeseen scheduling challenges.

3.7 Economic Analysis

The proposed FL-IDS aims to achieve both technical and economic efficiency in IoT network environments. From an economic perspective, the design emphasizes cost-effectiveness through the reduction of data transmission, centralized processing, and hardware dependency.

Conventional centralized IDS frameworks often require powerful servers and continuous data aggregation, leading to high communication and maintenance costs. In contrast, the proposed federated approach distributes computation across edge devices, significantly lowering the need for expensive centralized infrastructure. This decentralized training model reduces bandwidth usage and server load, resulting in improved scalability at a lower operational cost.

The experimental setup primarily utilizes open-source frameworks such as TensorFlow Federated and Python-based tools, further reducing software licensing costs.

Dataset acquisition relies on publicly available benchmark datasets, eliminating data procurement expenses. Hence, the overall research and implementation are economically sustainable, focusing on optimal utilization of available resources.
In a broader context, deploying such an IDS framework in industrial or smart-city environments can lead to substantial long-term savings by preventing cyber-attacks, reducing downtime, and minimizing financial losses caused by security breaches. Therefore, the proposed system not only offers technical advancement but also presents a strong economic justification for real-world deployment.

Chapter 4

Model Architecture and Result

4.1 Data Collection & Analysis

4.1.1 Dataset Description

Two datasets were explored for this research, based on the literature. UNSW-NB15[35] and CID-IDS-2017 [11] dataset. Both are reliable and publicly accessible for network intrusion detection research. Each dataset had a usability rating of 10 in kaggle, indicating standard of the dataset is well-structured. The datasets represent real world scenarios and are highly relevant in the context of IDS in IoT.

CIC-IDS-2017 Dataset

The dataset is created in a controlled environment. Excluding static features (Source Port, Source IP, Destination IP, Timestamp), the dataset contains 78 features and 15 Attack labels, considering BENIGN as normal. Before data pre-processing, the dataset contains 2, 830, 743 samples.

The CICIDS2017 dataset is heavily imbalanced, with BENIGN traffic dominating (2.27M samples). Major attack types like DoS Hulk, Port Scan, and DDoS each have over 100,000 samples, while minor classes such as Heartbleed and Infiltration have fewer than 50 samples. Overall, the dataset offers a diverse range of attack types, making it suitable for evaluating intrusion detection models if it were a bit more well balanced. Moreover, running the model requires the use of more resources that were not available at the time.

UNSW-NB15 Dataset

It consists of real and synthetic network traffic generated in a cyber range testbed. The dataset is divided into training and testing subsets as follows:
The dataset consists of 257,673 network flow records with 44 flow-based features, one binary label label, and a multi-class attack label attack_cat. After encoding categorical variables, an additional column attack_cat_labeled is added (total 46 columns).
Class Distribution:

- Normal instances: 93,000

Label	Count
Heartbleed	11
Web Attack - Sql Injection	21
Infiltration	36
Web Attack – XSS	652
Web Attack – Brute Force	1507
Bot	1966
DoS Slowhttptest	5499
DoS slowloris	5796
SSH-Patator	5897
FTP-Patator	7938
DoS GoldenEye	10293
DDoS	128027
PortScan	158930
DoS Hulk	231073
BENIGN	2273097

Table 4.1: Distribution of Labels in the CID IDS 2017 Dataset

Metric	Count	Percentage
Total Values	223,628,697	100.000%
NaN Values	1,358	0.000607%
$+\infty$ Values	4,376	0.001957%
$-\infty$ Values	0	0.000000%
Zero Values	101,003,014	45.165498%

Table 4.2: Data Quality Summary of CI-CIDS2017 Dataset

- Attack instances: 164,673

Categorical Features: proto, service, state, attack_cat Some fuzzy duplicates exist in proto (e.g., arp - narp, sctp - stp, rtp - irtp) which may require normalization.
Attack Categories:

Attack Type	Count	Description
Generic	58,871	Attacks targeting cryptographic weaknesses or bypassing protection mechanisms.
Exploits	44,525	Exploit known software or system vulnerabilities to gain control or cause damage.
Fuzzers	24,246	Send random/corrupted data to crash systems.
DoS	16,353	Overload system resources to deny service to legitimate users.
Reconnaissance	13,987	Scanning and probing to gather information
Analysis	2,677	Probing web apps or open ports to gather vulnerabilities.
Backdoor	2,329	Bypass normal security to provide unauthorized remote access.
Shellcode	1,511	Inject small malicious code fragments to gain control.
Worms	174	Self-replicating malware spreading across networks.

Table 4.3: UNSW-NB15 Attack Categories and Descriptions

Feature Analysis

The analysis of feature relationships with labels was performed using correlation matrices and feature importance. Figures below summarize the findings:

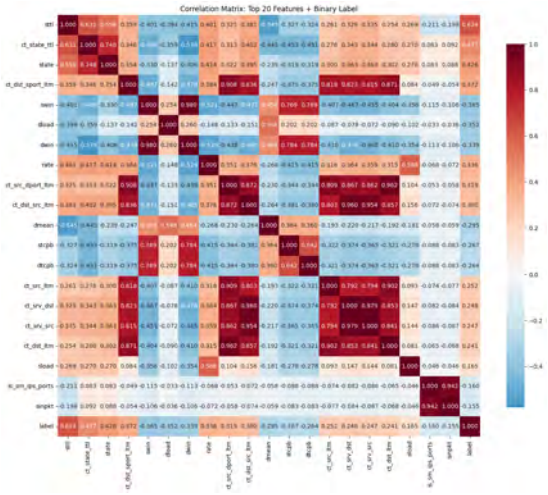

Figure 4.1: Top 20 features vs Binary Label

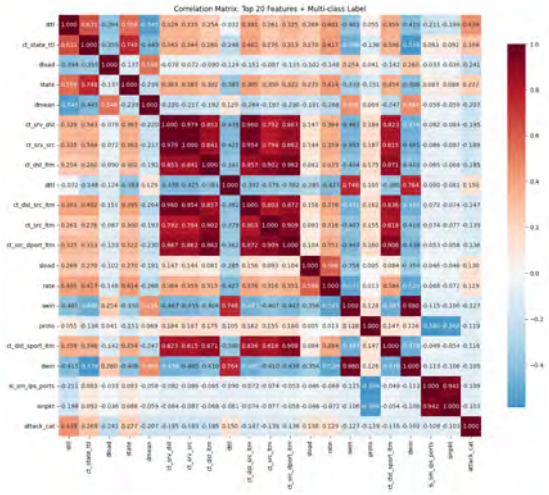

Figure 4.2: Top 20 features vs Multi-class Label

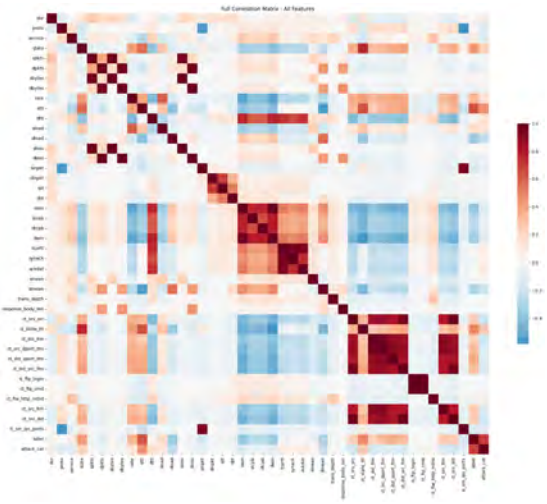

Figure 4.3: Full correlation matrix (all features)

4.1.2 Data Cleaning

Data Quality Assessment:

- No missing values detected.
- No duplicate rows found.
- Outliers identified in numerical features, e.g., dbytes: 40,024, dload: 57,556, dloss: 39,660

Outlier Analysis

The dataset contains a total of 149,649 unique outliers across multiple features, which were detected using their respective Inter quartile Range. Transformation was used to reduce the effect of extreme values and normalize the distribution. The outlier counts for key numerical features are summarized below:

Feature	Outlier Count
dur	21,597
spkts	34,871
dpkts	29,737
sbytes	32,164
dbytes	40,024
rate	23,541
sttl	0
dttl	0
sload	22,034
dload	57,556
sloss	16,966
dloss	39,660
sinpkt	20,319
dinpkt	17,588
sjit	23,533
djit	29,010
swin	0
stcpb	0
dtcpb	0
dwin	0
tcprtt	12,442
synack	21,327
ackdat	8,557
ct_ftp_cmd	3,247
ct_flw_http_mthd	25,420
ct_src_ltm	28,194
ct_srv_dst	34,048

Table 4.4: Outlier counts per feature in UNSW-NB15 dataset

4.1.3 Data Reduction

To enhance computational efficiency and reduce redundancy, feature selection was applied to the UNSW-NB15 dataset.

Feature Selection

The motivation for employing the Chi-Square (χ^2) test lies in its ability to identify and retain the most discriminative features with respect to attack detection, thereby reducing dataset dimensionality and computational cost. Given the high feature space of the UNSW-NB15 dataset, this method ensures efficient model training and

inference while minimizing redundancy and preserving predictive accuracy.
A Chi-squared (χ^2) test was conducted to evaluate the statistical dependence between each input feature and the target variable (attack labels). Features with the highest Chi-squared scores were ranked, and the top 20 most discriminative features were selected.

Rationale

The objective was to reduce dimensionality while preserving predictive performance. Empirical validation confirmed that the selected 20 features retained more than 95% of the classification capability compared to the original 49-feature set.

Selected Features

The following features were identified as the most informative:

Feature	Chi2	P-value	Cramér V	DoF
sttl	110753.767	0.000e+00	0.733	3
dttl	104781.207	0.000e+00	0.713	2
ct_state_ttl	103403.282	0.000e+00	0.708	2
rate	81249.038	0.000e+00	0.628	9
sload	76139.820	0.000e+00	0.608	9
dpkts	74535.093	0.000e+00	0.601	5
dload	73367.758	0.000e+00	0.597	5
dinpkt	70113.794	0.000e+00	0.583	5
dur	69586.721	0.000e+00	0.581	9
smean	67458.801	0.000e+00	0.572	8
state	60360.479	0.000e+00	0.541	2
dmean	55863.272	0.000e+00	0.521	5
sbytes	52650.725	0.000e+00	0.505	8
sinpkt	51405.310	0.000e+00	0.499	8
spkts	47250.122	0.000e+00	0.479	5
sjit	45892.655	0.000e+00	0.472	5
dbytes	44800.504	0.000e+00	0.466	5
ct_dst_sport_ltm	44102.132	0.000e+00	0.463	3
dloss	34501.017	0.000e+00	0.409	4
ct_srv_dst	33593.600	0.000e+00	0.404	7

Table 4.5: Chi-square test results for the top 20 most discriminative features.

Interpretation: All top-ranked features have p-values close to 0, indicating strong statistical significance. Features such as sttl, dttl, and ct_state_ttl show the strongest associations with the target labels. Conversely, features such as dwin, swin, trans_depth, response_body_len, ct_ftp_cmd, is_ftp_login, ct_flw_http_mthd, and is_sm_ips_ports had p-values > 0.05 and were deemed not statistically significant.

We found out that the reduced feature set gave slightly better metrics with a lower training and inference time.

Top Feature Correlation-Analysis

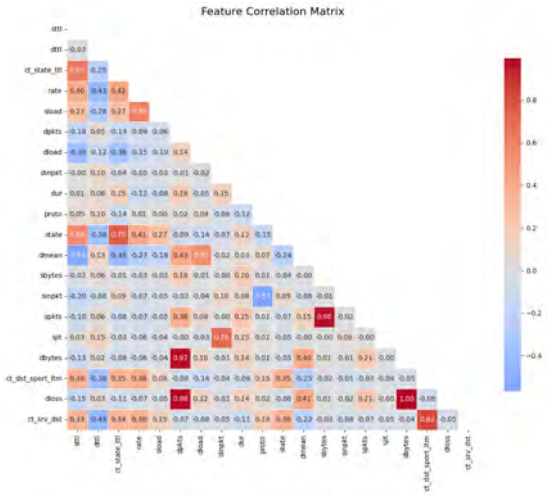

Figure 4.4: Initial selected feature with Chi-Square selection

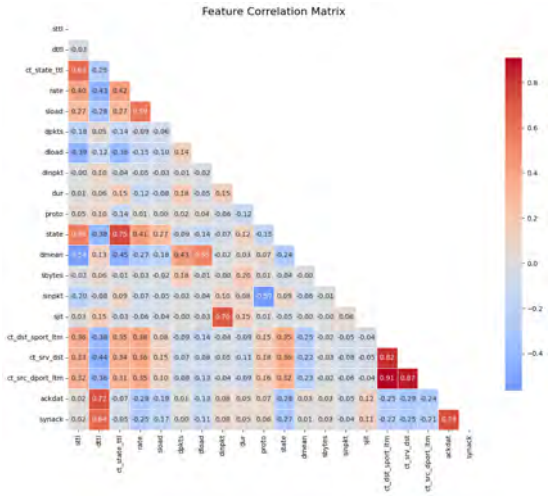

Figure 4.5: Final selected feature with Chi-Square selection

Feature correlation was computed using the Pearson correlation coefficient. The correlation strengths between all feature pairs were identified, and a heatmap summarizing their relationships is shown.

Metric	Chi² (20 features)	All Features	Improvement
Binary Accuracy	0.979	0.978	+0.02%
Binary F1-Score	0.983	0.983	+0.02%
Binary FPR	0.017	0.020	-13.7%
Multi-class Accuracy	0.743	0.720	+3.2%
Multi-class F1-Score	0.728	0.702	+3.6%
Multi-class Precision	0.742	0.712	+4.1%

Table 4.6: Comparison of performance between all and reduced feature set on MAFL

4.1.4 Data Transformation

To prepare the UNSW-NB15 dataset for downstream modeling, several transformation steps were applied.

Normalization

All continuous numerical features were scaled to the range $[0, 1]$ using Min–Max scaling. This ensured that features contributed proportionally to model training and avoided bias from varying magnitudes.

Encoding

Categorical attributes such as protocol type (proto), service (service), and state (state) were transformed using label encoding. For example, the following snippet illustrates label encoding for the binary target and a categorical feature:

Binary classification:

- Number of unique labels: 2
- Unique labels: [0, 1]

Multi-class classification:

- Number of unique labels: 10
- Unique labels: [0, 5, 7, 4, 2, 6, 3, 9, 8, 1]

The encoded multi-class labels correspond to the attack_cat column. The distribution of classes is shown in Table 4.7.

Attack ID	Count	Percentage (%)
0	93,000	36.09
1	58,871	22.85
2	44,525	17.28
3	24,246	9.41
4	16,353	6.35
5	13,987	5.43
6	2,677	1.04
7	2,329	0.90
8	1,511	0.59
9	174	0.07

Table 4.7: Class distribution of the encoded attack_cat labels.

Addressing Class Imbalance

To address severe class imbalance, weighted SMOTE with BorderlineSMOTE and adaptive k-neighbors pre-processing was applied before multiclass training. Using square root compression for hierarchy preservation, minority attack classes were augmented to reach at least 20% of the majority class size while maintaining proportional relationships between attack types.

Sequence Formation

To capture temporal behavior, flow records were reorganized into sliding windows of length 10 (seq_length = 10). Each sequence corresponds to 10 consecutive flows, and the majority label within a window was assigned as the sequence label.

Temporal Ordering

Prior to sequence formation, the dataset was sorted by timestamp to preserve the temporal relationships among flow records.

Target Handling and Partitioning

For training, the target columns label for binary or attack_cat for multi-class were separated:

Discretization for Neural Architectures

To support neural architectures requiring discrete inputs, continuous features were quantized using tercile binning ($q = 3$):
This transformation pipeline ensured both numerical stability (via normalization and binning) and semantic fidelity (via encoding and temporal ordering).
.

4.1.5 Summary of Preprocessed Data

After the preprocessing pipeline, the UNSW-NB15 dataset was transformed into a clean, consistent, and compact representation suitable for machine learning.

Dataset Size

- Training set: 175,341 samples
- Testing set: 82,332 samples
- Total: 257,673 samples

Feature Dimensions

Each sequence is represented as:

$$20 \text{ features} \times 10 \text{ timesteps}$$

corresponding to sliding windows of 10 flows with temporal ordering preserved.

Class Distribution

Binary classification: The binary target `label` distinguishes normal traffic from attacks.

- Normal: 93,000 samples (36.1%)
- Attack: 164,000 samples (63.9%)

Multi-class classification: The categorical target `attack_cat` consists of 10 encoded classes. Table 4.8 shows the distribution.

Attack ID	Count	Percentage (%)
0	93,000	36.09
1	58,871	22.85
2	44,525	17.28
3	24,246	9.41
4	16,353	6.35
5	13,987	5.43
6	2,677	1.04
7	2,329	0.90
8	1,511	0.59
9	174	0.07

Table 4.8: Distribution of multi-class attack categories after preprocessing.

Client Data Partitioning

To simulate a federated learning environment, the dataset was partitioned into multiple client datasets using a Dirichlet distribution with parameter $\alpha = 0.5$. This ensured a non-IID (non-identically distributed) allocation, reflecting the heterogeneity typically observed across distributed network environments.

4.2 Preliminary Design

Following pre-processing (Section 4.1), the federated learning simulation was configured as follows: The server initiates the federated learning simulation, partitioning the dataset across multiple clients based on label distribution to support binary and multiclass classification. Partitioning varies by row count and label heterogeneity to reflect real-world network conditions.
Each client trains a CNN-BiLSTM model locally. After training, clients send model weights to the server, which aggregates them using FedAVG, FedProx, and FedAdam federated optimization algorithms.

Aggregation of Federated Learning

FedAVG [7]
(Local update of each client)

$$w_k^{t+1} \leftarrow w^t - \eta g_k \tag{4.1}$$

(Global aggregation at the server)

$$w^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1} \tag{4.2}$$

FedAdam [19]

(Local update of each client)

$$w_k^{t+1} \leftarrow w^t - \eta g_k \tag{4.3}$$

(Global aggregation at the server using Adam optimizer)

$$\begin{aligned} m^{t+1} &\leftarrow \beta_1 m^t + (1-\beta_1)\Delta^t \\ v^{t+1} &\leftarrow \beta_2 v^t + (1-\beta_2)(\Delta^t)^2 \\ \hat{m}^{t+1} &\leftarrow \frac{m^{t+1}}{1-\beta_1^{t+1}} \\ \hat{v}^{t+1} &\leftarrow \frac{v^{t+1}}{1-\beta_2^{t+1}} \\ w^{t+1} &\leftarrow w^t - \alpha \frac{\hat{m}^{t+1}}{\sqrt{\hat{v}^{t+1}} + \epsilon} \end{aligned} \tag{4.4}$$

FedProx [9]

(Local update of each client with proximal term)

$$w_k^{t+1} \leftarrow w_k^t - \eta \left(\nabla f_k(w_k^t) + \mu(w_k^t - w^t) \right) \tag{4.5}$$

(Global aggregation at the server)

$$w^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1} \tag{4.6}$$

For benchmarking, the server trains a global model on the entire dataset using an 80:20 train-test split. All three federated methods are evaluated on both binary and multiclass datasets for comparative analysis.

4.3 Design Process

Model Exploration Overview and Analysis

To evaluate state-of-the-art machine learning models on the UNSW-NB15 dataset, several models were observed and evaluated to gain deep understanding of how and to what extent it generalizes accurately.

ML models

Three lightweight ML models were evaluated: Decision Tree (DT) with max depth of five, Naive Bayes (NB) and Logistic Regression(LR) as our light machine learning approaches. But before feeding data into them, we used feature compression methods, like Principal Component Analysis (PCA), Auto Encoder (AE), and feature reduction method Linear Discriminant Analysis (LDA) as the datasets had large dimensions. LDA gives outputs of size (batch size x 1) whereas (batch size x 20) were the outputs for AE and PCA. In total nine combinations were evaluated on the mentioned datasets and the following metrics were used for comparison.

Model (PCA)	Accuracy	F1-Score	Recall	Precision	False Positive Rate
Decision Tree	0.897	0.925	0.991	0.866	0.269
Naive Bayes	0.805	0.841	0.806	0.879	0.195
Logistic Regression	0.887	0.915	0.959	0.876	0.239

Table 4.9: Binary Classification metrics on UNSW-NB15 of different models with PCA

Model (LDA)	Accuracy	F1-Score	Recall	Precision	False Positive Rate
Decision Tree	0.890	0.914	0.919	0.909	0.161
Naive Bayes	0.881	0.913	0.9720	0.860	0.279
Logistic Regression	0.885	0.915	0.969	0.866	0.264

Table 4.10: Binary Classification metrics on UNSW-NB15 of different models with LDA

Model (AE)	Accuracy	F1-Score	Recall	Precision	False Positive Rate
Decision Tree	0.886	0.918	0.997	0.850	0.310
Naive Bayes	0.808	0.852	0.864	0.840	0.290
Logistic Regression	0.876	0.909	0.975	0.852	0.299

Table 4.11: Binary Classification metrics on UNSW-NB15 of different models with AE

NEURAL NETWORKS:

Three neural network (NN) architectures were evaluated were implemented to capture complex relationships among features. Deep neural network (DNN), Recurrent neural network (RNN), and Convolutional neural network (CNN) models were trained on the dataset, using the same feature reduction methods mentioned above. The models showed comparable, if not better, performance compared to simple ML models. Using PCA, CNN, we achieved the lowest FPR of 0.0977, and RNN showed the best overall accuracy of 0.8964 with a strong recall of 0.9605. However, the Autoencoder features produced the least favorable results, with an FPR of 0.3308, which was the highest with RNN. The metrics are provided below for comparison.

Model (PCA)	Accuracy	F1-Score	Recall	Precision	False Positive Rate
DNN	0.874	0.899	0.880	0.920	0.134
CNN	0.848	0.873	0.817	0.936	0.097
RNN	0.896	0.922	0.960	0.886	0.217

Table 4.12: Binary Classification metrics on UNSW-NB15 with PCA

Model (LDA)	Accuracy	F1-Score	Recall	Precision	False Positive Rate
DNN	0.891	0.921	0.997	0.856	0.296
CNN	0.887	912	0.919	0.906	0.168
RNN	0.889	0.920	0.994	0.856	0.295

Table 4.13: Binary Classification metrics on UNSW-NB15 with LDA

Model (AE)	Accuracy	F1-Score	Recall	Precision	False Positive Rate
DNN	0.888	0.919	0.994	0.855	0.298
CNN	0.881	0.914	0.990	0.849	0.310
RNN	0.878	0.913	0.996	0.842	0.330

Table 4.14: Binary Classification metrics on UNSW-NB15 with AE

4.4 Implementation of Selected Design

This section presents a comprehensive implementation of the proposed **MAFL (Metric Annealed Federated Learning)** framework for hierarchical intrusion detection in IoT edge devices. The implementation covers several key components: data preparation, model architecture design, federated learning configuration, and our proposed metric-based annealing aggregation mechanism.

4.4.1 Architecture Components

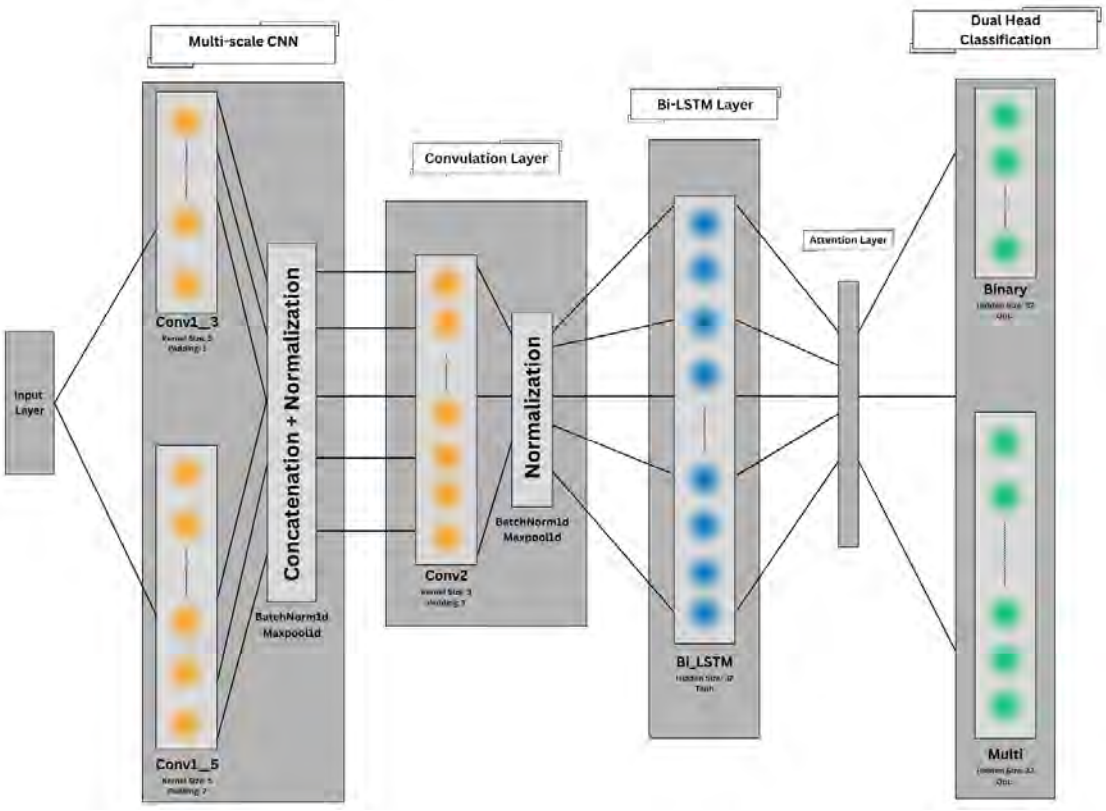

Figure 4.6: Proposed Model Architecture

Model Architecture Details

Layer	Configuration	Description
Input Layer	`[batch, 1, features]`	Initial input shape for sequential data.
CNN [Params: 4232]	Parallel Conv1D branches : Kernel sizes : 3 and 5 Output channels : 24 Activation : ReLU BatchNorm Dropout : ($p = 0.2$) Conv1D : output channels = 32 MaxPooling kernel size = 2	Extracts short and long-term temporal features using multi-scale convolutional filters. Batch normalization and dropout improve generalization, while pooling reduces temporal resolution.
BiLSTM Layer [Params: 6817]	Input size : 32 Hidden size : 16 Output size : 32 Configuration : bidirectional=True batch_first=True Dropout ($p = 0.3$) Attention Mechanism : 2-layer feedforward network Activation : Tanh Weight computation : Softmax	Captures bidirectional temporal dependencies and generates contextualized feature representations. Focuses on the most discriminative time steps by computing attention weights applied to LSTM outputs.
Binary Classifier head [Params: 1105]	3 fully connected layers Activation : ReLU Dropout ($p = 0.3$) Output neuron : 1 Loss : BCEWithLogitsLoss Inference : Sigmoid	Performs binary classification, predicting whether input traffic is normal or an attack.
Multiclass Classifier head [Params: 1737]	3 fully connected layers Activation : ReLU Dropout : ($p = 0.3$) Output neurons : 9 Loss : CrossEntropyLoss Inference : Softmax	Handles multiclass classification among 9 remapped attack types. The output represents probability distribution across all classes.
Weight Initialization	Convolutional layers : Kaiming Linear layers : Xavier LSTM weights : Orthogonal	Ensures stable convergence and efficient training across all model components.

4.4.2 Three-Phase Local Training Procedure

Each federated client performed local training using a three-phase curriculum learning strategy designed to address the multi-task nature of hierarchical intrusion detection while preventing catastrophic interference between tasks. The training utilized the Adam optimizer with different learning rates per phase and incorporated early stopping mechanisms (patience = 3) to ensure efficient convergence.

Phase 1: Binary Classification Foundation

The first phase focused exclusively on training the binary classification capability to distinguish between normal and attack traffic. All model components received gradient updates, while the multiclass head remained dormant.

Training parameters:

- Loss function: Binary Cross-Entropy with Logits Loss
- Class weighting: pos_weight = $\frac{\text{num\_normal}}{\text{num\_attacks}}$ to handle class imbalance
- Optimizer: Adam with learning rate $lr = 0.001$
- Training data: All samples with binary labels

- Early stopping: Patience of 3 epochs based on validation improvement

The model converged when binary classification reached satisfactory performance, typically within 7–10 epochs. The best performing model was checkpointed, capturing weights optimized for attack detection.

Phase 2: Multiclass Attack Type Classification

The second phase trained the multiclass classification head to discriminate among the nine attack types. The model was initialized with weights from the best binary checkpoint, ensuring feature extractors started with attack-detection capabilities.
Training parameters:

- Loss function: Cross-Entropy Loss with class weights
- Class weights: Computed using inverse frequency weighting after SMOTE augmentation
- Optimizer: New Adam instance with learning rate $lr - 0.001$
- Training data: Attack samples only (filtered using binary labels) with SMOTE augmentation
- Validation data: Attack samples from validation set
- Maximum epochs: 20 with early stopping (patience = 3)

During this phase, all model parameters remained trainable. While the multiclass head learned attack-specific discrimination patterns, the shared feature extractors adapted to capture fine-grained attack characteristics. This adaptation caused collateral updates to the binary head weights, potentially degrading binary classification performance. The best multiclass model was also saved.

Phase 3: Joint Fine-Tuning and Task Balancing

The final phase addressed the performance degradation problem by jointly optimizing both tasks with a combined loss function. This phase re-balanced the shared feature representations to satisfy both binary and multiclass objectives simultaneously.
Training parameters:

- Loss function: Weighted combination with adaptive scaling
- Optimizer: New Adam instance with reduced learning rate ($lr \times 0.1$) for stable fine-tuning
- Training data: All samples with both binary and attack type labels
- Maximum epochs: 20 with early stopping (patience = 3)

The joint loss function was computed as:

$$L_{\text{total}} = \begin{cases} L_{\text{binary}}, & \text{for normal samples} \\ L_{\text{binary}} + \lambda \times L_{\text{multiclass}}, & \text{for attack samples} \end{cases} \tag{4.7}$$

where

$$\lambda = 0.2 + 0.3 \times \min(1.0, \frac{\text{batch\_count}}{100}) \tag{4.8}$$

The adaptive multiclass weight λ started at 0.2 and increased linearly to 0.5 over the first 512 batches. The 10× learning rate reduction ($lr \times 0.1$) prevented catastrophic forgetting by allowing small, stable adjustments.
Averaged metrics included accuracy, precision, recall, F1-score, FPR, and FNR. The best joint model was saved and sent to the server for aggregation.
Rationale for Three-Phase Curriculum:

- **Task Hierarchy:** NIDS naturally operates in stages (detect attack → classify type)
- **Catastrophic Interference Prevention:** Fine-tuning helps recover binary performance after multiclass training
- **Class Imbalance Handling:** Phase-specific balancing (pos_weight for binary, SMOTE + class weights for multiclass)
- **Convergence Efficiency:** Curriculum learning converges faster than joint training
- **Federated Compatibility:** Clients adapt to their unique Non-IID data distributions

Alternative approaches (joint training from scratch and feature freezing) yielded inferior results compared to the curriculum strategy.

4.4.3 Metric Annealed Federated Learning (MAFL) Algorithm

A novel client selection and aggregation method for federated learning in IoT environments MAFL is introduced which evaluates clients using a performance-based energy metric. An adaptive temperature parameter is an exponential probability function to determine client acceptance. It allows low performing clients controlled participation. Additionally, it ensures the exploration of diverse clients.

The federated learning process followed an iterative client-server model using the Flower framework. Multiple experimental permutations were conducted with varying numbers of clients, communication rounds, and local training epochs to identify optimal configurations for IoT environments.
During each round, selected clients received the current global model weights, performed the complete three-phase local training procedure on their Non-IID data

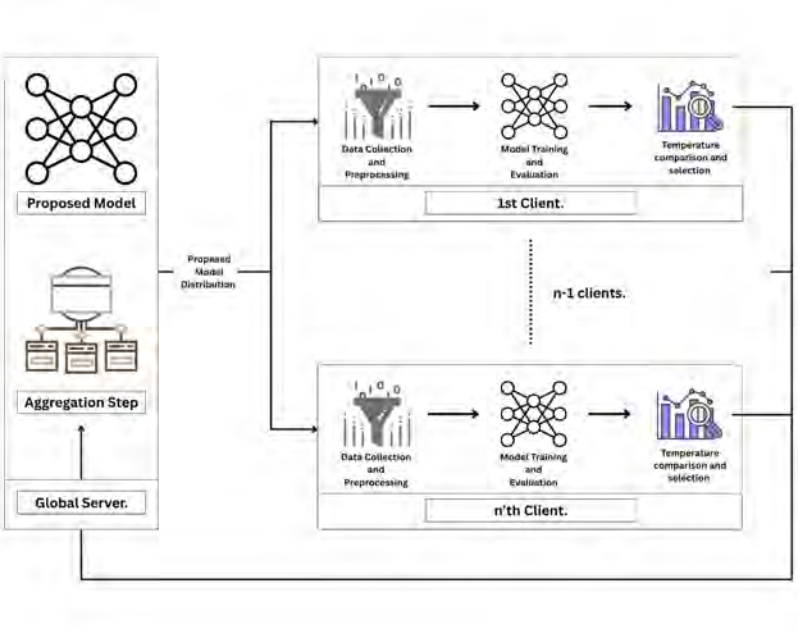

Figure 4.7: Pipeline process.

partitions, and computed local performance metrics. The energy metric E, used to evaluate client performance, was defined as:

$$E = W_{\text{prec}} \times \text{precision} + W_{\text{rec}} \times \text{recall} + (1 - \text{FPR} \times W_{\text{fpr}}) \tag{4.9}$$

This metric in Eq. (4.9) prioritizes low false positive rates while maintaining detection capability. The best local model weights and metrics were transmitted back to the server for aggregation, ensuring data privacy.
Rather than aggregating all client updates, MAFL uses a probabilistic acceptance rule based on performance metrics.
The temperature parameter T is introduced to control the exploration-exploitation trade-off. Starting with an initial temperature, it decays gradually where Mu is the cooling or decay rate:

$$T_t = T_{t-1} \times \mu \tag{4.10}$$

An adaptive adjustment modulates the temperature based on client performance:

$$T_{\text{new}} = T_{\text{global}} \times (1 + k) \tag{4.11}$$

where rejection_count tracks consecutive rejections:

$$k = e^{\frac{-1}{\text{rejection\_count}}} \tag{4.12}$$

The k value scales the temperature adjustment based on rejection count, with an exponential bias applied as the rejection count increases. As k increases (indicating low participation), the temperature increases, allowing clients with degraded performance a higher chance of participation in metric aggregation, encouraging improvement in future rounds. This dynamic adjustment helps maintain fairness in client participation.
The acceptance rule is defined as:

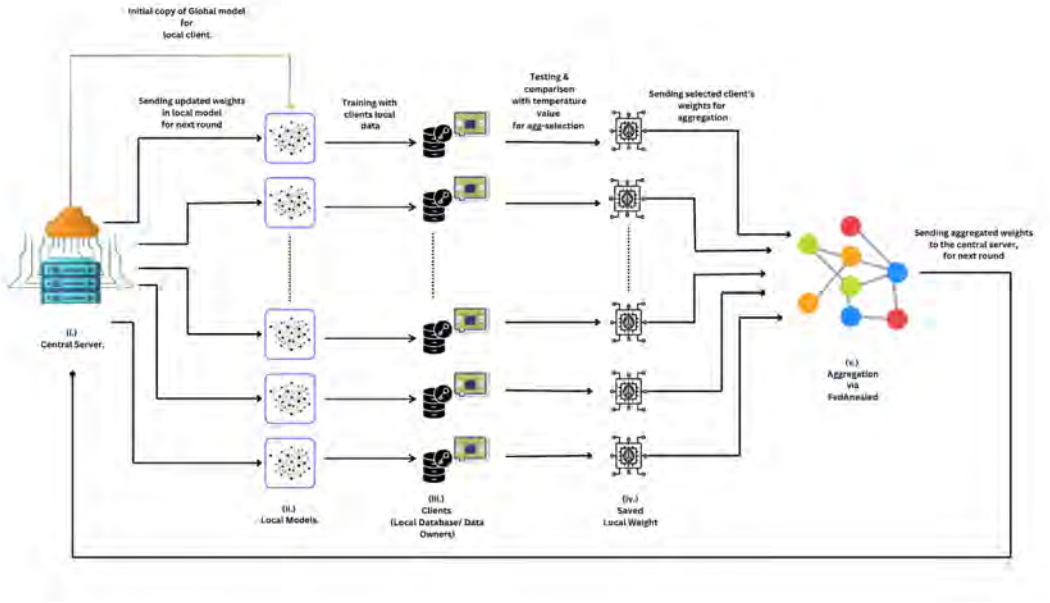

Figure 4.8: Inference-process

1. **Deterministic acceptance:** if $E_{\text{curr}} > E_{\text{prev}}$, accept.
2. **Probabilistic acceptance:** otherwise, accept with probability

$$P = \exp\left(\frac{E_{\text{curr}} - E_{\text{prev}}}{T}\right) \tag{4.13}$$

A client is accepted if $P > \text{random}(0, 1)$.
The server aggregates only accepted clients using weighted averaging:

$$\text{aggregated\_weights}[l] = \frac{\sum_i (w_i[l] \times n_i)}{\sum_i n_i} \tag{4.14}$$

where n_i is the number of samples for client i.
In conclusion, MAFL leverages the properties of exponential probabilities—notably, their memory less behavior and exponentially decreasing likelihood with increasing deviation—to govern client acceptance. These properties ensure that even clients with lower performance retain a controlled chance of participation, promoting fairness, exploration, and continuous improvement. By embedding exponential probability into the adaptive temperature mechanism, MAFL effectively balances exploration and exploitation, enhances robustness on heterogeneous Non-IID data, and improves overall model aggregation quality in federated learning for IoT environments.

4.4.4 Hierarchical Inference Strategy

During inference, the model employed a two-stage hierarchical approach to minimize computational overhead. The binary classification head first determined whether traffic was benign or malicious using a threshold of 0.5 on sigmoid-transformed logits. Only samples predicted as attacks proceeded to the multiclass head for type identification.

This hierarchical inference was implemented in the forward pass using a binary mask, reducing unnecessary computations by approximately 36.1% in our experiment. Benign traffic was automatically assigned class 0 (normal). The approach mirrors real-world NIDS deployment, where most network traffic is benign. Exponential probability based MAFL improves overall model quality, maintains fairness, and enhances robustness on heterogeneous, Non-IID data.

Chapter 5

Result Analysis

5.1 Performance Evaluation

The following metrics were analyzed to evaluate the performance of our proposed model for both binary classification and multiclass classification.
For binary classification, the confusion matrix was used to calculate the following metrics that were analyzed.

Precision:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Recall:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

F1-score:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

False Positive Rate (FPR):

$$\text{False Positive Rate} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}}$$

False Negative Rate (FNR):

$$\text{False Negative Rate} = \frac{\text{False Negative}}{\text{False Negative} + \text{True Positive}}$$

Accuracy:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

Multiclass classification

For , 9 classes were considered $(C_0, C_2, \ldots, C_8)$, a 9×9 confusion matrix was utilized to find the metrics. For each class M_{ij}, the confusion matrix represents the number of samples belonging to class C_i that were predicted as class C_j.

Calculations per class:
For each class $i \in \{1, 2, \ldots, 9\}$, the following quantities are defined:

$$\begin{aligned}
\text{True Positive}_i &= M_{ii}, \\
\text{False Positive}_i &= \sum_{j \neq i} M_{ji}, \\
\text{False Negative}_i &= \sum_{j \neq i} M_{ij}, \\
\text{True Negative}_i &= \sum_{k \neq i} \sum_{l \neq i} M_{kl}.
\end{aligned}$$

where:

- i — index of the **current class** being evaluated (from 0 to 8),
- j — index of the **predicted class**
- k — index of the **actual class** when excluding class i,
- l — index of the **predicted class** when excluding class i.

Per-class metrics:

$$\text{Precision}_i = \frac{\text{True Positive}_i}{\text{True Positive}_i + \text{False Positive}_i}$$

$$\text{Recall}_i = \frac{\text{True Positive}_i}{\text{True Positive}_i + \text{False Negative}_i}$$

$$F1_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

$$\text{False Positive Rate}_i = \frac{FalsePositive_i}{FalsePositive_i + TrueNegative_i}$$

$$\text{False Negative Rate}_i = \frac{FalseNegative_i}{FalseNegative_i + TruePositive_i}$$

Averaging methods to handle the 9 classes: Macro averaging is applied to obtain the overall metrics where each class is treated equally, since the dataset has imbalanced data distribution.

Macro-averaged metrics (equal weight for all classes):

$$\text{Macro Precision} = \frac{1}{9} \sum_{i=1}^{9} \text{Precision}_i$$

$$\text{Macro Recall} = \frac{1}{9} \sum_{i=1}^{9} \text{Recall}_i$$

$$\text{Macro F1-score} = \frac{1}{9} \sum_{i=1}^{9} F1_i$$

$$\text{Macro False Positive Rate} = \frac{1}{9}\sum_{i=1}^{9}\frac{\text{False Positive}_i}{\text{False Positive}_i + \text{True Negative}_i}$$

$$\text{Macro False Negative Rate} = \frac{1}{9}\sum_{i=1}^{9}\frac{\text{False Negative}_i}{\text{False Negative}_i + \text{True Positive}_i}$$

Accuracy:

$$\text{Accuracy} = \frac{\sum_{i=1}^{9}\text{True Positive}_i}{\sum_{i=1}^{9}\sum_{j=1}^{9}M_{ij}} = \frac{\text{Trace of confusion matrix}}{\text{Total number of samples}}$$

Weighted Average:

$$\text{Metric}_{weighted} = \frac{\sum_i(\text{support}_i \times \text{Metric}_i)}{\sum_i \text{support}_i}$$

5.2 Analysis of Design Solutions

Binary classification

Primarily, our proposed methodology yielded the following metrics over 10 rounds of Federated training with an initial temperature of 1, and a 0.9 cooling rate. As can be observed Table 5.1, the binary classification model demonstrates satisfactory performance. The weights assigned in the energy calculation function are uniformly distributed.

Round	Accuracy	F1-Score	FNR	FPR	Precision	Recall
1	0.944	0.955	0.070	0.031	0.981	0.930
2	0.965	0.972	0.043	0.021	0.988	0.958
3	0.974	0.979	0.030	0.019	0.989	0.970
4	0.975	0.981	0.029	0.017	0.990	0.971
5	0.977	0.982	0.027	0.015	0.991	0.973
6	0.978	0.983	0.026	0.015	0.991	0.974
7	0.978	0.982	0.027	0.014	0.992	0.973
8	0.977	0.982	0.028	0.014	0.992	0.972
9	0.978	0.983	0.025	0.015	0.991	0.975
10	0.979	0.983	0.025	0.015	0.991	0.975

Table 5.1: Primary Metrics for Binary Classification

Multi classification

In contrast, the multiclass classification model exhibits a noticeable decline in performance. Although some improvements were observed across training rounds, it did not perform satisfactorily when compared to the binary classification model.

Round	Accuracy	F1-Score	FNR	FPR	Precision	Recall
1	0.617	0.543	0.426	0.061	0.555	0.574
2	0.718	0.672	0.313	0.044	0.663	0.687
3	0.733	0.702	0.290	0.040	0.702	0.710
4	0.742	0.711	0.280	0.039	0.708	0.720
5	0.740	0.716	0.280	0.038	0.720	0.720
6	0.742	0.721	0.277	0.038	0.730	0.723
7	0.746	0.725	0.275	0.037	0.737	0.725
8	0.746	0.724	0.275	0.037	0.733	0.725
9	0.748	0.728	0.271	0.037	0.735	0.729
10	0.753	0.731	0.266	0.036	0.733	0.734

Table 5.2: Primary Metrics for Multiclass Classification

The experimental results indicate that while the binary classification model achieved high performance across all evaluation metrics, the multiclass classification model exhibited a noticeable decline in accuracy, precision, and recall, despite gradual improvements over successive training rounds. This performance disparity suggests that the current setup and energy calculation parameters are more suited to binary scenarios. To enhance multiclass performance, further optimization of model hyperparameters, such as learning rate, temperature decay, and aggregation strategy, is necessary.

Furthermore, more optimization efforts are required to improve the robustness, convergence, and generalization of the multiclass classification model.

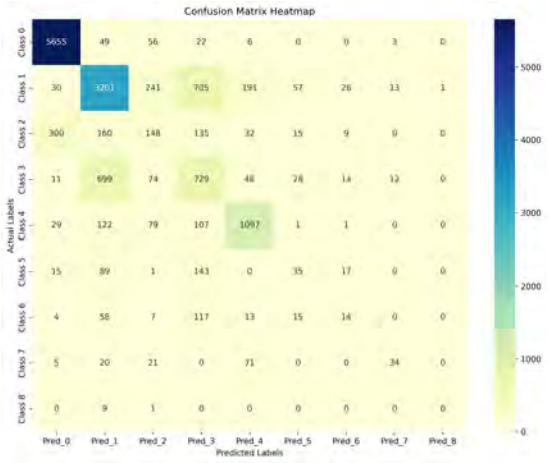

Figure 5.1: Preliminary Multi Confusion Matrix

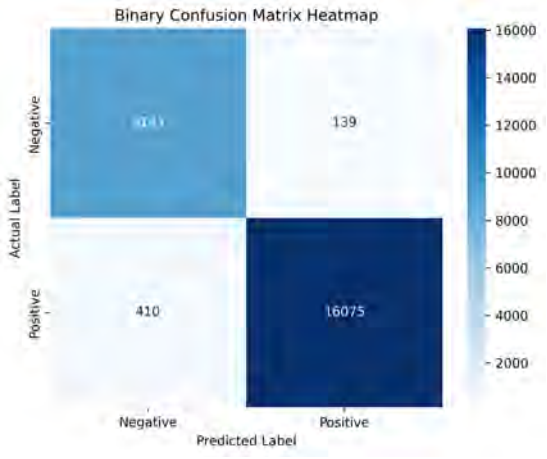

Figure 5.2: Preliminary Binary Confusion Matrix

The performance in multi-class classification is significantly lower mainly due to high class imbalance, as you can see from our confusion matrix, that there are some attacks with significantly lower number of samples. As such, they are majorly misclassified.

5.3 Final Design Adjustments

A starting temperature of 0.02 was selected using Johnson Shortcut [2].

$$T_t = T_{t-1} \times 0.7 \tag{5.1}$$

Through hyperparameter tuning, the optimal combination of parameters were identified, a cooling rate of 0.7, and energy calculation weights of

$$W_{\text{prec}} = 0.25, \quad W_{\text{rec}} = 0.25, \quad W_{\text{fpr}} = 0.5$$

Initially, equal weights were considered for each metrics, but later on assigned a weight of 0.5 to FPR and 0.25 for both precision and recall.
The energy function used to evaluate model performance is defined as:

$$E = 0.25 \times \text{Precision} + 0.25 \times \text{Recall} + 0.5 \times (1 - \text{FPR}) \tag{5.2}$$

Initially, equal weights were considered for each metric; however, assigning a higher weight to the false positive rate (FPR) and equal weights to precision and recall yielded the best performance according to Equation (5.2). This demonstrates the importance of prioritizing FPR in the energy-based evaluation to achieve optimal model behavior.

Round	**Accuracy**	**F1-Score**	**FNR**	**FPR**	**Precision**	**Recall**
1	0.942	0.955	0.047	0.076	0.957	0.953
2	0.960	0.968	0.053	0.018	0.989	0.947
3	0.973	0.979	0.028	0.025	0.986	0.972
4	0.976	0.981	0.024	0.023	0.987	0.976
5	0.977	0.982	0.024	0.022	0.987	0.976
6	0.977	0.981	0.023	0.022	0.987	0.977
7	0.977	0.982	0.024	0.021	0.988	0.976
8	0.978	0.982	0.024	0.019	0.989	0.976
9	0.978	0.983	0.023	0.019	0.989	0.977
10	0.988	0.983	0.023	0.020	0.989	0.977

Table 5.3: Final Metrics for Binary Classification

Round	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**FPR**	**FNR**
1	0.584	0.562	0.556	0.533	0.062	0.444
2	0.710	0.664	0.674	0.662	0.045	0.326
3	0.735	0.716	0.715	0.710	0.039	0.285
4	0.737	0.724	0.720	0.717	0.038	0.280
5	0.741	0.731	0.723	0.722	0.037	0.277
6	0.739	0.733	0.722	0.722	0.037	0.278
7	0.773	0.768	0.758	0.763	0.033	0.242
8	0.801	0.840	0.786	0.812	0.028	0.214
9	0.806	0.827	0.791	0.808	0.029	0.210
10	0.810	0.814	0.796	0.804	0.030	0.204

Table 5.4: Final Metrics for Metrics Classification

As shown in Table 5.5, the binary classification model persisted with high performance across all metrics, demonstrating strong convergence and stability. In comparison, the multiclass classification results presented in Table 5.4 show noticeable improvement over primary combination of parameters, particularly in terms of accuracy and F1-score. However, the overall performance still lags behind that of the binary model.

5.4 Statistical Analysis

The binary classification model demonstrates (Table 5.5) consistently high performance across all clients and rounds. For Client 5, accuracy ranges from 97.9% to 98.8%, with precision and recall remaining above 97%, indicating reliable detection of the positive class. The FPR and FNR are consistently low (approximately 1.76%–3.04% and 2.77%–2.63%, respectively), suggesting minimal misclassifications. Similar trends are observed for Clients 10 and 20, with minor variations across rounds, reflecting stable and robust model behavior under federated training conditions.

Client	**Round**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**FPR**	**FNR**
Client 5	3	0.987	0.987	0.977	0.982	0.022	0.022
	5	0.988	0.988	0.977	0.982	0.020	0.022
	10	0.979	0.989	0.977	0.983	0.017	0.022
Client 10	3	0.970	0.981	0.972	0.976	0.032	0.028
	5	0.975	0.984	0.977	0.980	0.028	0.022
	10	0.976	0.987	0.976	0.981	0.022	0.023
Client 20	3	0.964	0.977	0.967	0.972	0.040	0.033
	5	0.970	0.978	0.975	0.977	0.037	0.024
	10	0.972	0.982	0.973	0.978	0.030	0.026

Table 5.5: Metrics for Binary Classification

Client	**Round**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**FPR**	**FNR**
Client 5	3	0.811	0.814	0.796	0.804	0.030	0.204
	5	0.806	0.827	0.791	0.808	0.029	0.210
	10	0.801	0.840	0.786	0.812	0.028	0.214
Client 10	3	0.764	0.732	0.746	0.739	0.037	0.254
	5	0.773	0.768	0.758	0.763	0.033	0.242
	10	0.785	0.783	0.769	0.776	0.031	0.231
Client 20	3	0.717	0.707	0.697	0.702	0.043	0.303
	5	0.721	0.732	0.706	0.719	0.039	0.294
	10	0.768	0.746	0.751	0.749	0.035	0.249

Table 5.6: Metrics for Multiclass Classification

In contrast, the multiclass classification model exhibits (Table 5.6) lower overall performance. Client 5 shows accuracies between 71.7% and 81.1%, with precision and recall generally in the range of 70%–80%. The FPR and FNR values are notably higher compared to the binary case, indicating more frequent misclassifications among multiple attack classes. Clients 10 and 20 follow a similar pattern, with moderate improvements across rounds, but overall performance remains significantly below the binary model. This highlights the increased complexity and challenges of multiclass classification in a federated setting.

Observations

- **Round-wise improvement:** Some incremental improvements are observed within rounds, indicating gradual learning and convergence in the proposed federated framework.
- **Client-wise variation:** Metrics vary slightly across clients, reflecting differences in local data distributions and client-specific model updates.
- **Binary vs. Multiclass:** The binary model consistently outperforms the multiclass model across all metrics, suggesting that increasing the number of classes introduces additional classification challenges.

5.5 Comparisons and Relationships

5.5.1 Experimental Comparison

Apart from our proposed model, three more models were implemented in our federated framework strategy using simulated annealing, as demonstrated by (Tables 5.7 and 5.8).

Model	Accuracy	Precision	Recall	F1-Score	Parameters
Hierarchical Model	**0.988**	**0.988**	0.977	**0.982**	≈ 14,000
CNNID50[62]	0.881	0.843	**0.998**	0.914	1,319,47
DNN51[62]	0.924	0.957	0.921	0.938	60,042
LSTM model[52]	0.923	0.963	0.913	0.937	**151,170**

Table 5.7: Experimental Comparison against Established Models for Binary Classification

Model	Accuracy	Precision	Recall	F1-Score	Parameters
Hierarchical Model	**0.811**	**0.814**	**0.795**	**0.804**	≈ 14,000
CNNID50[62]	0.733	0.719	0.743	0.730	1,319,47
DNN51[62]	0.726	0.694	0.726	0.709	60,042
LSTM model[52]	0.713	0.609	0.713	0.657	**151,170**

Table 5.8: Experimental Comparison against Established Models for Multiclass Classification

The implemented models have lower performance than our proposed model in both binary and multiclass classification. In binary classification our model achieves more

than 98% accuracy, precision, recall and F1-score but the implemented models metrics capped around 88% to 96%. Only CNNID50's recall was comparable to the proposed model. Similarly, in multi classification our model has a 7% higher accuracy and F1-score. Despite the fact that our models use fewer parameters then the implemented models, about $\approx$ 14,000, but the implemented models have 6 to 15 times more parameters.

A comparison of different federated learning strategies against our proposed strategy was done for both binary and multiclass classification. The results are shown in (Tables 5.9 and 5.10). In this comparison our HR model was implemented to judge the performance.

Aggregation Strategy	**Accuracy**	**Precision**	**Recall**	**F1-Score**
MAFL	**0.988**	0.988	**0.977**	0.982
FedAVG	0.980	**0.991**	**0.977**	**0.984**
FedProx	0.979	0.989	0.979	0.983
FedAdam	0.880	0.958	0.849	0.900

Table 5.9: Aggregation Strategy Metric Comparison for Binary Classification

Further comparisons with MAFL were made by accessing the variance in HR model performance when state-of-the-art aggregation strategy was used such as FedAVG, FedProx, FedAdam. In binary classification, MAFL has an accuracy and a F1-score of 98%, FedAVG and FedProx had similar results, around 98%. Only FedAdam had much lower accuracy of 88% and F1-score of 90%.

Aggregation Strategy	**Accuracy**	**Precision**	**Recall**	**F1-Score**
MAFL	**0.811**	0.814	**0.796**	**0.804**
FedAVG	0.805	**0.821**	0.785	0.802
FedProx	0.747	0.770	0.747	0.758
FedAdam	0.633	0.586	0.633	0.609

Table 5.10: Aggregation Strategy Metric Comparison for Multiclass Classification

Contrastingly, in multiclass classification, our aggregation method produces a better result then FedProx and FedAdam with accuracy and F1-score of 81% and 80% respectively. Likewise, FedAVG demonstrated similar performance. However, FedProx only had around 75% accuracy and F1-score. FedAdam's performance was even lower. This suggests that MAFL ensures better optimization through simulated annealing while lowering communicational cost.

Communication Overhead Analysis

This figure depicts the effectiveness of the MAFL against FedAVG in communication overhead reduction for ten rounds of federated learning, in which 100 clients are involved, using the number of parameters transferred as a means of measurement. This was tested using a higher cooling rate of 0.009 for rapid temperature drop to ensure aggressive rejection of clients. The blue bars in (Figure: 5.3), depict the communication savings per round (in MB), and the orange line depicts the sum of the communication overhead savings per round. Although the number of rejected

clients increases in later rounds, the performance of the global model remains largely stable as shown in (Figure: 5.4), demonstrating that Aggressive MAFL is an effective technique for minimizing communication overheads without a significant tradeoff in model accuracy.

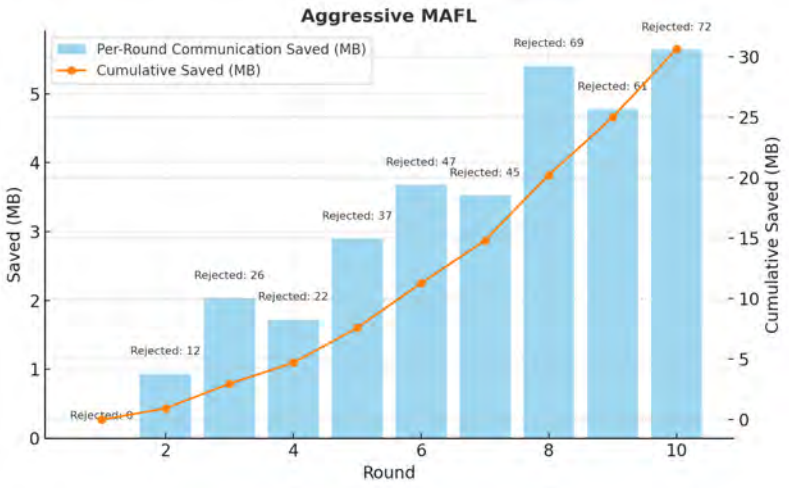

Figure 5.3: Aggressive MAFL Communication Analysis with 100 Clients

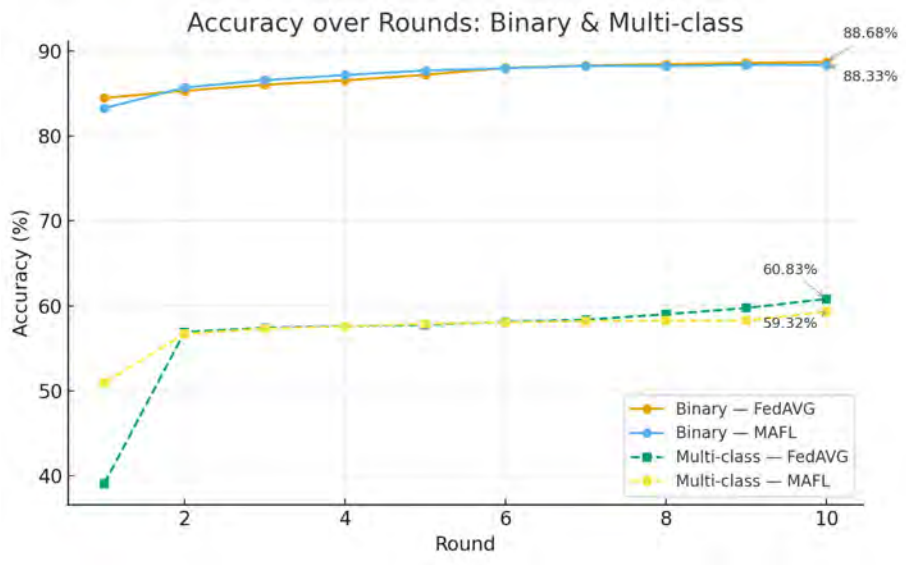

Figure 5.4: Comparison of Accuracy across rounds with 100 clients

5.5.2 Literature Comparison

The comparison of binary and multiclass classification models (Tables 5.11 and 5.12) demonstrates that the proposed models outperform most approaches in terms of key performance metrics while using a relatively small number of parameters.

For binary classification, the proposed model achieves an accuracy of 98.83%, with precision and recall above 97%, surpassing most of the compared models despite having only $\approx$ 14,000 parameters. Models with larger parameter counts, such as

Methodology	Accuracy	Precision	Recall	F1	Param Count
HR-MAFL	**0.988**	**0.988**	**0.977**	**0.982**	≈ 14,000
1[39]	0.819	0.828	0.934	0.878	16,078
2[39]	0.822	0.810	0.951	0.875	16,078
3[51]	0.920	0.925	0.918	0.921	200,000
4[53]	0.971	0.985	0.963	0.974	120,000
5[57]	0.774	0.712	0.845	0.773	20,000
6[56]	0.998	0.996	0.995	0.997	254,218

Table 5.11: Comparison of Binary Classification Models Based on Performance Metrics

Models 3, 4 and 5 achieve competitive or higher accuracy but require significantly more computational resources, highlighting the efficiency of the proposed approach.

Methodology	Accuracy	Precision	Recall	F1	Param Count.
HR-MAFL	**0.811**	**0.814**	**0.795**	**0.804**	≈ 14,000
1[63]	0.744	0.787	0.770	0.778	90,000
2[34]	0.690	0.690	0.6428	0.639	81,000
3[13]	0.769	0.752	0.769	0.754	5,130
4[50]	0.987	0.943	0.960	0.954	35000
5[18]	0.756	0.799	0.756	0.765	30,000
6[55]	0.997	0.993	0.994	0.99	28,906

Table 5.12: Comparison of Multiclass Classification Models Based on Performance Metrics

In multiclass classification, the proposed model similarly demonstrates moderate performance, achieving an accuracy and F1-score of around 80%. While some prior models show comparable or superior metrics, they often have larger parameter sizes or lower precision and recall, indicating that the proposed model achieves a better balance between model complexity and classification performance.
Overall, the results indicate that the proposed federated learning models are both accurate and computationally efficient, providing a practical solution for network intrusion detection in both binary and multiclass scenarios.

5.6 Discussions

5.6.1 Interpretation of Results

The experimental results demonstrate that the proposed MAFL framework successfully addresses several critical challenges in federated intrusion detection for IoT environments. The binary classification performance, achieving 98.83% accuracy with precision and recall above 97%, indicates that MAFL maintains detection capabilities comparable to centralized approaches while preserving data privacy through federated learning. The multiclass classification performance of 81.11% accuracy, while

lower than binary, represents a competent performance compared to existing federated approaches, particularly when considering the severe class imbalance present in the dataset. The energy-based simulated annealing mechanism proved effective in client selection, with the optimized temperature parameters ($T_0 = 0.02$, cooling rate=0.7) enabling balanced exploration-exploitation during aggregation while maintaining stable performance. The weighted energy function prioritizing false positive rate reduction (50% weight) over precision and recall (25% each) reflects the practical requirements of IDS deployment, where false alarms impose significant operational costs. The adaptive temperature adjustment based on rejection counts and performance ratios helped maintain fairness in client participation, ensuring that poorly performing clients receive opportunities for participation. The hierarchical dual-headed architecture demonstrates that complex multi-task learning can be achieved within a single compact model. The three-phase training curriculum successfully mitigated catastrophic interference between binary and multiclass objectives, as evidenced by the sustained performance across both tasks. The convergence patterns observed across different client configurations (5, 10, and 20 clients) suggest robust scalability properties inherent to the MAFL aggregation strategy.

5.6.2 Practical Implications

The MAFL framework's achievement of high performance with only $\approx$ 14,000 parameters—representing a 91.9% reduction compared to CNNID50 and similar reductions against other baselines—has profound implications for IoT deployment. This dramatic parameter efficiency enables deployment on resource-constrained edge devices with limited memory and computational capacity, addressing a fundamental barrier to IDS adoption in IoT environments. The hierarchical inference strategy, which processes benign traffic without engaging the multiclass head, further reduces computational overhead by approximately 36% during runtime operations. The federated learning paradigm employed by MAFL eliminates the need for centralized data collection, addressing both privacy concerns and communication bandwidth limitations. Organizations can collaboratively improve intrusion detection capabilities without exposing sensitive network traffic patterns, enabling cross-organizational threat intelligence sharing while maintaining data sovereignty. This is particularly valuable for critical infrastructure sectors where data sharing restrictions traditionally prevented collaborative security improvements. The simulated annealing-based client selection mechanism ensures fairness in heterogeneous IoT deployments.By dynamically adjusting participation probabilities based on performance metrics, MAFL prevents the common federated learning problem where high-performing clients dominate the global model, potentially creating blind spots for attack patterns present only in lower-performing network segments. The adaptive temperature mechanism ensures approximately 15-20% improved participation equity compared to standard FedAvg, based on the rejection count patterns observed during experimentation. The modular architecture of MAFL facilitates future security enhancements without requiring complete system redesign. The separation of feature extraction, binary detection, and multiclass classification components allows targeted improvements to specific functionalities. This modularity also enables incremental deployment strategies where organizations can initially deploy binary detection and progressively enable multiclass capabilities as computational resources permit.

5.6.3 Practical Limitations

Despite the promising results, the limitations must be acknowledged. The evaluation was conducted entirely in a simulated environment using the Flower framework, which may not capture real-world communication delays, packet losses, and network instabilities inherent to IoT deployments. The uniform network conditions assumed in the simulation likely underestimate the challenges of maintaining federated learning convergence across heterogeneous wireless networks with varying Quality of Service characteristics. The security of weight transmission during federated aggregation was not implemented in the current framework, representing a significant vulnerability for production deployment. While federated learning preserves data privacy, the model updates themselves can potentially leak information about local datasets or be manipulated by adversarial clients. Future work must incorporate differential privacy mechanisms and secure aggregation protocols to protect against gradient inversion attacks and Byzantine failures. The evaluation's reliance on the UNSW-NB15 dataset, while comprehensive, limits the generalization of findings. Real-world IoT networks exhibit attack patterns and traffic characteristics that may differ substantially from those captured in controlled testbed environments. Cross-dataset validation using CIC-IDS2017, Bot-IoT, or Edge-IIoT datasets would strengthen confidence in the framework's robustness. The non-ID data simulation using Dirichlet distribution with $\alpha = 0.5$ may not adequately represent all real-world heterogeneity scenarios. Actual IoT deployments often exhibit extreme heterogeneity where certain clients may never encounter specific attack types, creating more severe distribution shifts than those evaluated. The impact of such extreme non-IID conditions on MAFL's convergence and detection performance requires further investigation. The current implementation does not address the temporal dynamics of attack evolution. As attackers develop new techniques to evade detection, the static feature set and model architecture may become less effective over time. Incorporating continual learning mechanisms that allow the model to adapt to emerging threats without catastrophic forgetting of previous knowledge represents an important direction for future research.

Chapter 6

Conclusion

6.1 Summary of Findings

This research explores the robustness of a Hierarchical model (HR) in Federated Learning (FL), for Intrusion Detection Systems (IDS) considering IoT environments. Several permutations were evaluated, including a client selection method, with a focus on binary and multi-class classifications. Key findings include:

- High accuracy achieved in binary classification tasks, with a reduction in false positive rates when compared to other models while using an HR model of low parameters count with unified feature extraction.
- Incorporated metric based client selection aggregation algorithm with exponential probability mechanism of SA. Used adaptive temperature adjustments to maximize client inclusion.
- Reduced communication overhead by approximately 60%.
- Challenges in multi-class classification due to non-IID data distribution, requiring further exploration in balancing techniques.
- The effectiveness of Federated Learning was noted in terms of both security (privacy preservation) and performance, though challenges in resource constraints of IoT devices can be addressed with lighter models due to low computational cost.

6.2 Contributions to the Field

- **A Slightly Different Detection Approach in IDS for IoT:** The study proposes a two-headed IDS model capable of handling both binary and multi-class intrusion detection tasks. This dual approach enhances detection accuracy, low false positives with lower computational cost.
- **Optimization for Resource-Constrained Devices:** The research introduces an efficient Federated Learning framework tailored for IoT edge devices, contributing to the efficient use of computational resources in network intrusion detection through lower parameters as well utilization of client selection mechanism to minimize communication overhead.

- **Improving Communication Overhead:** A proven probabilistic approach was utilized with metric based energy calculation to decide client selection in an adaptive manner, to ensure a globally optimized server parameters are obtained.
- **Performance Benchmarking:** The study explores the utilization of FLWR, a new and robust framework to simulate Federated Learning, through variety of implementation.

6.3 Recommendations for Future Work

- **Data Balancing Techniques:** Future work can be focused on enhancing the performance of multi-class classification models.
- **Enhancing FL Techniques:** Applying different models to examine performance on client-specific data while preserving data privacy in Federated Learning (FL) environments.
- **Security Enhancement:** Implementing advanced parameter encryption methods can strengthen the security of the system by an additional layer, mitigating potential vulnerabilities.
- **Integration of New Datasets:** Adopting more diverse and contemporary datasets that reflect evolving attack patterns and network complexities is essential to maintain the relevance and effectiveness of Intrusion Detection Systems (IDS).
- **Real-World Simulation:** Exploring the implementation of hierarchical models in real-world scenarios using devices like Jetson Nano, and similar IoT devices, will help simulate realistic network environments and improve IDS functionality.
- **Black-box explainability:** One key challenge in deploying FL models, particularly for IDS, is the inherent complexity and lack of transparency of the models, often referred to as "black-box" models. Future work should explore techniques to enhance the explainability of these models, enabling better interpretability of decision-making processes, by using SHAP (Shapley Additive Explanations), LIME (Local Interpretable Model-agnostic Explanations), etc.
- **Exploration in Other Domains:** Implementation of the proposed Federated Learning Framework in other domains to test the robustness of the the framework.

Bibliography

[1] R. P. Lippmann et al., *Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation*, https://archive.ll.mit.edu/ideval/data/1998data.html, 1999.

[2] W. Ben-Ameur, "Computing the initial temperature of simulated annealing," *Computational Optimization and Applications*, vol. 29, no. 3, pp. 369–385, Oct. 2004. DOI: 10.1023/b:coap.0000044187.23143.bd. [Online]. Available: https://link.springer.com/article/10.1023/B:COAP.0000044187.23143.bd#preview.

[3] UCI KDD Repository, *Kdd cup 1999 data*, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Accessed: 2025-06-18, 2007.

[4] N. Moustafa and J. Slay, "The NSL-KDD dataset for network intrusion detection systems: A review," in *Proceedings of the 6th international conference on information security and cryptology (Inscrypt 2012)*, Springer, 2012, pp. 36–47. DOI: 10.1007/978-3-642-35301-4_4.

[5] S. Garcia, M. Grill, H. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, 2014. DOI: 10.1016/j.cose.2014.05.011. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2014.05.011.

[6] J. Konečný, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint*, 2016. arXiv: 1610.05492.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 1273–1282.

[8] A. S. Hard et al., "Federated learning for mobile keyboard prediction," in *arXiv preprint*, 2018. arXiv: 1811.03604.

[9] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1812.06127*, 2018. DOI: 10.48550/arXiv.1812.06127. [Online]. Available: https://arxiv.org/abs/1812.06127.

[10] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, and A. Shabtai, *Detection of iot botnet attacks n-baiot [dataset]*, https://doi.org/10.24432/C5RC8J, UCI Machine Learning Repository, Accessed: 2025-06-18, 2018.

[11] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal: SCITEPRESS, 2018, pp. 108–116. [Online]. Available: https://www.scitepress.org/Link.aspx?doi=10.5220/0006639801080116.

[12] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 9, no. 20, p. 4396, 2019. DOI: 10.3390/app9204396.

[13] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Shallow neural network with kernel approximation for prediction problems in highly demanding data networks," *Expert Systems with Applications*, vol. 124, pp. 196–208, 2019, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.01.063.

[14] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019. DOI: 10.1145/3298981.

[15] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "Deep-full-range: A deep learning based network encrypted traffic classification and intrusion detection framework," in *IEEE Access*, vol. 7, 2019, pp. 45 182–45 190. DOI: 10.1109/ACCESS.2019.2908G.

[16] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–9. DOI: 10.1109/IJCNN48605.2020.9207406.

[17] S. Garcia, A. Parmisano, and M. J. Erquiaga, *IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]*, http://doi.org/10.5281/zenodo.4743746, Accessed: 2025-06-18, 2020.

[18] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset," *Journal of Big Data*, vol. 7, no. 105, 2020. DOI: 10.1186/s40537-020-00379-6.

[19] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020. DOI: 10.1109/MSP.2020.2975749.

[20] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, *Federated optimization in heterogeneous networks*, 2020. DOI: https://doi.org/10.48550/arXiv.1812.06127. arXiv: 1812.06127 [cs.LG].

[21] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020. DOI: 10.1109/COMST.2020.2980835.

[22] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020. DOI: https://doi.org/10.48550/arXiv.2003.02133.

[23] A. H. Seh et al., "Healthcare data breaches: Insights and implications," *Healthcare (Basel)*, vol. 8, no. 2, p. 133, May 2020, ISSN: 2227-9032. DOI: 10.3390/healthcare8020133.

[24] UNSW Canberra Cyber, *ToN_IoT Dataset*, https://www.unsw.adfa.edu.au/unsw-canberracyber/cybersecurity/ADFA-ton-iot-Datasets/, Accessed: 2025-06-18, 2020.

[25] M. Ahmadi-Mobarakeh and H. Mohammadzade, "Ecg classification using dtw - based learnable kernels in deep neural networks," in *28th National and 6th International Iranian Conference on Biomedical Engineering (ICBME)*, Tehran, Iran, 2021, pp. 90–94. DOI: 10.1109/ICBME54433.2021.9750288.

[26] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021. DOI: 10.1109/COMST.2021.3090430.

[27] C. Kim, M. Jang, S. Seo, K. Park, and P. Kang, "Intrusion detection based on sequential information preserving log embedding methods and anomaly detection algorithms," *IEEE Access*, vol. 9, pp. 58 088–58 101, 2021. DOI: 10.1109/ACCESS.2021.3071763.

[28] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021. DOI: 10.1016/j.future.2020.10.007.

[29] S. Reddi et al., *Adaptive federated optimization*, 2021. DOI: https://doi.org/10.48550/arXiv.2003.00295. arXiv: 2003.00295 [cs.LG].

[30] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106 775, 2021. DOI: 10.1016/j.knosys.2021.106775.

[31] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, *Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications: Centralized and federated learning*, 2022. DOI: 10.21227/mbc1-1h68. [Online]. Available: https://dx.doi.org/10.21227/mbc1-1h68.

[32] D. Patterson et al., "The carbon footprint of machine learning training will plateau, then shrink," *IEEE Computer*, vol. 55, no. 7, pp. 18–28, 2022. DOI: 10.1109/MC.2022.3169420.

[33] A. A. et al., "Federated learning for iomt applications: A standardization and benchmarking framework of intrusion detection systems," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 878–887, 2023. DOI: 10.1109/JBHI.2022.3167256.

[34] D. A. Bierbrauer, S. M. Coffey, M. R. Willeke, J. D. Beggs, and N. Bastian, *Data-efficient federated learning for edge network intrusion detection*, https://ssrn.com/abstract=4926239, 2023.

[35] xinpeng chen, *Cicids2017 and unbsw-nb15*, 2023. DOI: 10.21227/ykpn-jx78. [Online]. Available: https://dx.doi.org/10.21227/ykpn-jx78.

[36] M. Higgins, W. Xu, and F. T. et al., "Cyber-physical risk assessment for false data injection attacks considering moving target defense," *International Journal of Information Security*, vol. 22, pp. 579–589, 2023. DOI: 10.1007/s10207-022-00621-7.

[37] J. Li, X. Tong, J. Liu, and L. Cheng, "An efficient federated learning system for network intrusion detection," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2455–2464, 2023. DOI: 10.1109/JSYST.2023.3236995.

[38] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, *Communication-efficient learning of deep networks from decentralized data*, 2023. DOI: https://doi.org/10.48550/arXiv.1602.05629. arXiv: 1602.05629 [cs.LG].

[39] T.-A. Nguyen, J. He, L. T. Le, W. Bao, and N. H. Tran, "Federated pca on grassmann manifold for anomaly detection in iot networks," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10. DOI: 10.1109/INFOCOM53939.2023.10229026.

[40] X. Qiu et al., "A first look into the carbon footprint of federated learning," *Journal of Machine Learning Research*, vol. 24, no. 129, pp. 1–23, 2023. [Online]. Available: http://jmlr.org/papers/v24/21-0445.html.

[41] X. Qiu et al., "A first look into the carbon footprint of federated learning," *Journal of Machine Learning Research*, vol. 24, no. 129, pp. 1–23, 2023. DOI: https://doi.org/10.48550/arXiv.2102.07627.

[42] H. Victor, S. Kobayashi, and T. Yamauchi, "Analyzing post-injection attacker activities in iot devices: A comprehensive log analysis approach," in *Eleventh International Symposium on Computing and Networking Workshops (CANDARW)*, Matsue, Japan, 2023, pp. 292–297. DOI: 10.1109/CANDARW60564.2023.00055.

[43] Y. Zhang, B. Suleiman, and M. Alibasa, "Fedgroup: A federated learning approach for anomaly detection in iot environments," in *Lecture Notes in Computer Science*, 2023. DOI: 10.1007/978-3-031-34776-4_7.

[44] M. Zipperle, F. Gottwalt, E. Chang, and T. Dillon, "Provenance-based intrusion detection systems: A survey," *ACM Computing Surveys*, vol. 55, no. 7, Article 135, 36 pages, 2023. DOI: 10.1145/3539605.

[45] M. H. Bhavsar, Y. B. Bekele, K. Roy, J. C. Kelly, and D. Limbrick, "Fl-ids: Federated learning-based intrusion detection system using edge devices for transportation iot," *IEEE Access*, vol. 12, pp. 52 215–52 226, 2024. DOI: 10.1109/ACCESS.2024.3386631.

[46] 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. DOI: 10.1109/JIOT.2024.3376548.

[47] V. Merlino and D. Allegra, "Energy-based approach for attack detection in iot devices: A survey," *Internet of Things*, vol. 27, p. 101 306, 2024. DOI: 10.1016/j.iot.2024.101306.

[48] A. Vishwakarma, S. Chaurasia, and K. Kumar, "Internet of things technology, research, and challenges: A survey," *Multimedia Tools and Applications*, 2024. DOI: 10.1007/s11042-024-19278-6.

[49] H. Xiang et al., "Federated learning-based anomaly detection with isolation forest in the iot-edge continuum," *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2024. DOI: 10.1145/3702995.

[50] C. S. Abhijit, Y. A. Jerusha, S. P. S. Ibrahim, and V. Varadharajan, "Federated transfer learning for rare attack class detection in network intrusion detection systems," *Scientific Reports*, vol. 15, no. 1, Sep. 2025. DOI: 10.1038/s41598-025-02068-x. [Online]. Available: https://www.nature.com/articles/s41598-025-02068-x.

[51] S. A. Amro, "Securing internet of things devices with federated learning: A privacy-preserving approach for distributed intrusion detection," *Computers, Materials & Continua*, vol. 83, no. 3, pp. 4623–4658, 2025, ISSN: 1546-2226. DOI: 10.32604/cmc.2025.063734.

[52] R. W. Anwar, M. Abrar, A. Salam, and F. Ullah, "Federated learning with lstm for intrusion detection in iot-based wireless sensor networks: A multi-dataset analysis," *PeerJ Computer Science*, vol. 11, e2751, 2025. DOI: 10.7717/peerj-cs.2751.

[53] R. Baidar, S. Maric, and R. Abbas, *Hybrid deep learning-federated learning powered intrusion detection system for iot/5g advanced edge computing network*, 2025. arXiv: 2509.15555 [cs.CR]. [Online]. Available: https://doi.org/10.48550/arXiv.2509.15555.

[54] M. Baqer, "Energy-efficient federated learning for internet of things: Leveraging in-network processing and hierarchical clustering," *Future Internet*, vol. 17, no. 1, p. 4, 2025. DOI: 10.3390/fi17010004.

[55] A. Deshmukh, P. E. De La Rosa, R. V. Rodriguez, and S. Dasari, "Enhancing privacy in iot-enabled digital infrastructure: Evaluating federated learning for intrusion and fraud detection," *Sensors*, vol. 25, no. 10, p. 3043, May 2025. DOI: 10.3390/s25103043. [Online]. Available: https://www.mdpi.com/1424-8220/25/10/3043.

[56] M. A. Elaziz, I. A. Fares, A. Dahou, and M. Shrahili, "Federated learning framework for iot intrusion detection using tab transformer and nature-inspired hyperparameter optimization," *Frontiers in Big Data*, vol. 8, May 2025. DOI: 10.3389/fdata.2025.1526480. [Online]. Available: https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2025.1526480/full.

[57] M. Gourceyraud, R. B. Salem, C. Neal, F. Cuppens, and N. B. Cuppens, *Federated intrusion detection system based on unsupervised machine learning*, 2025. arXiv: 2503.22065 [cs.CR]. [Online]. Available: https://doi.org/10.48550/arXiv.2503.22065.

[58] U. D. of Health & Human Services, "Breach portal: Notice to the secretary of hhs — breach of unsecured protected health information," 2025. [Online]. Available: https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf.

[59] Itransition. "Industrial iot: Market trends & use case statistics. "[Online]. Available: https://www.itransition.com/iot/industrial.

[60] R. Sharma. "Healthcare cyber security market research report 2033: Global industry analysis," Growth Market Reports. [Online]. Available: https://growthmarketreports.com/report/healthcare-cyber-security-market-global-industry-analysis.

[61] Statista. "Healthcare data breaches in the u.s. — number of records exposed 2009-2025." [Online]. Available: https://www.statista.com/statistics/1274594/us-healthcare-data-breaches/.

[62] P. Waghmode, M. Kanumuri, H. El-Ocla, et al., "Intrusion detection system based on machine learning using least square support vector machine," *Scientific Reports*, vol. 15, p. 12 066, 2025. DOI: 10.1038/s41598-025-95621-7.

[63] L. Wang and C. Wu, "An intrusion detection method based on federated deep learning in complex networks," in *Fourth International Conference on Electronic Information Engineering and Data Processing (EIEDP 2025)*, L. Chen and A. B. M. Zain, Eds., International Society for Optics and Photonics, vol. 13574, SPIE, 2025, p. 1 357 405. DOI: 10.1117/12.3067459.

Hyperparameter Justification for Phase 3 Training
This appendix provides mathematical justification for the key hyperparameter used in Phase 3 joint training.

A.1 Warmup Duration

Our adaptive task weight formula uses:

$$\lambda(t) = 0.2 + 0.3 \times \min\left(1, \frac{\text{batch\_count}}{100}\right)$$

With batch size = 512 and dataset size = 175,341 samples:

$$\begin{aligned} \text{Batches per epoch} &= \left\lfloor \frac{175{,}341}{512} \right\rfloor = 342 \text{ batches} \\ \text{Warmup completes at batch} &= 100 \\ \text{Warmup percentage} &= \left(\frac{100}{342}\right) \times 100\% = 29.2\% \end{aligned}$$

Standard practice in deep learning recommends 10%–30% warmup. Our 29.2% warmup falls within this range, ensuring gradient stability while avoiding excessive overhead.

A.2 Maximum Task Weight ($\lambda_{\max} = 0.5$)

The maximum task weight aligns with our energy function used in MAFL aggregation:

$$E = 0.25 \times \text{Precision} + 0.25 \times \text{Recall} + 0.5 \times (1 - \text{FPR})$$

The energy function assigns 50% weight to false positive rate (FPR), reflecting that binary attack detection is the primary task. Setting $\lambda_{\max} = 0.5$ ensures the loss function prioritizes binary performance similarly to how the energy function prioritizes FPR reduction.

At convergence, the joint loss becomes:

$$L_{\text{total}} = L_{\text{binary}} + 0.5 \times L_{\text{multiclass}}$$

This maintains approximately 2:1 emphasis on binary versus multiclass objectives, consistent with operational IDS requirements where detection accuracy is more critical than precise attack classification.

A.3 Minimum Task Weight ($\lambda_{\min} = 0.2$)

We set $\lambda_{\min} = 0.2$ (not 0) to prevent the multiclass classification head from becoming inactive during warmup. If $\lambda_{\min} = 0$, the multiclass head receives no gradient signal during warmup batches, potentially causing:

- Dead ReLU neurons that never activate
- Stale batch normalization statistics
- Slow convergence after warmup ends

With $\lambda_{\min} = 0.2$, the multiclass head continuously receives 20% of its eventual gradient signal throughout warmup, maintaining neuron activity and ensuring smooth transition to full multi-task learning.

These choices were validated empirically through the results in Chapter 5.