Journal of
Sensor and Actuator Networks

IMPACT FACTOR 4.2

CITESCORE 9.4

Article

Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study

Nurtay Albanbay, Yerlan Tursynbek, Kalman Graffi, Raissa Uskenbayeva, Zhuldyz Kalpeyeva, Zhastalap Abilkaiyr and Yerlan Ayapov

Special Issue
Federated Learning: Applications and Future Directions

Edited by
Dr. Giovanni Paragliola, Dr. Laura Verde, Dr. Fiammetta Marulli and Dr. Rosario Catelli

MDPI

https://doi.org/10.3390/jsan14040078

Journal of
Sensor and Actuator Networks

MDPI

Article

Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study

Nurtay Albanbay [1], Yerlan Tursynbek [1,*], Kalman Graffi [2], Raissa Uskenbayeva [1], Zhuldyz Kalpeyeva [1], Zhastalap Abilkaiyr [1] and Yerlan Ayapov [1]

[1] Institute of Automation and Information Technologies, Satbayev University, Satbayev 22, Almaty 050013, Kazakhstan; n.albanbay@satbayev.university (N.A.); r.k.uskenbayeva@satbayev.university (R.U.); z.kalpeyeva@satbayev.university (Z.K.); z.abilkaiyr@satbayev.university (Z.A.); 040610502260@stud.satbayev.university (Y.A.)
[2] Department for Computer Science, Technical University of Applied Sciences Bingen, Berlinstr. 109, 55411 Bingen, Germany; k.graffi@th-bingen.de
* Correspondence: 971031300885-d@stud.satbayev.university

Abstract

This paper presents a large-scale empirical study aimed at identifying the optimal local deep learning model and data volume for deploying intrusion detection systems (IDS) on resource-constrained IoT devices using federated learning (FL). While previous studies on FL-based IDS for IoT have primarily focused on maximizing accuracy, they often overlook the computational limitations of IoT hardware and the feasibility of local model deployment. In this work, three deep learning architectures—a deep neural network (DNN), a convolutional neural network (CNN), and a hybrid CNN+BiLSTM—are trained using the CICIoT2023 dataset within a federated learning environment simulating up to 150 IoT devices. The study evaluates how detection accuracy, convergence speed, and inference costs (latency and model size) vary across different local data scales and model complexities. Results demonstrate that CNN achieves the best trade-off between detection performance and computational efficiency, reaching ~98% accuracy with low latency and a compact model footprint. The more complex CNN+BiLSTM architecture yields slightly higher accuracy (~99%) at a significantly greater computational cost. Deployment tests on Raspberry Pi 5 devices confirm that all three models can be effectively implemented on real-world IoT edge hardware. These findings offer practical guidance for researchers and practitioners in selecting scalable and lightweight IDS models suitable for real-world federated IoT deployments, supporting secure and efficient anomaly detection in urban IoT networks.

Keywords: federated learning; Internet of Things; intrusion detection system; deep learning; data scaling; DNN; CNN; BiLSTM; CICIoT 2023; anomaly detection

check for updates

Academic Editors: Laura Verde, Fiammetta Marulli, Rosario Catelli and Giovanni Paragliola

Received: 10 May 2025
Revised: 26 June 2025
Accepted: 8 July 2025
Published: 23 July 2025

Citation: Albanbay, N.; Tursynbek, Y.; Graffi, K.; Uskenbayeva, R.; Kalpeyeva, Z.; Abilkaiyr, Z.; Ayapov, Y. Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study. *J. Sens. Actuator Netw.* **2025**, *14*, 78. https://doi.org/10.3390/jsan14040078

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Cybersecurity is one of the key research areas today as information systems and infrastructures are increasingly embedded in everyday life. The growing dependence on data and the rising number of devices participating in the digital economy necessitate the reliable protection of servers, computers, and network devices from numerous cyber threats. In particular, the expansion of the Internet of Things (IoT) means that even household appliances become potential targets for attacks, creating additional challenges for traditional security systems [1].

The early era of cybersecurity was characterized by using intrusion detection systems (IDS) based on the analysis of network traffic and the monitoring of system behavior. These systems relied on extensive databases of attack signatures, which enabled the detection of known threats by analyzing data packets and user actions [2]. However, these methods proved ineffective when adapting to new, evolving threats, such as massive Distributed Denial-of-Service (DDoS) attacks, where the system is overloaded through simultaneous impact from many sources [3].

The situation became more complicated with the advent of the Internet of Things. In IoT environments, not only traditional computers and servers are connected, but also smart household devices, such as refrigerators, coffee machines, security systems, and even wearable gadgets [4]. These devices often have limited computational resources and low energy efficiency, making them particularly vulnerable to continuous network threats. Memory and processing power limitations prevent the use of resource-intensive protection methods and large-scale deep learning models that have been designed for server systems.

Recent studies in IoT networks have revealed numerous gaps in existing solutions. The primary issue is that traditional IDS methods, whether signature-based or anomaly-based, do not provide sufficient adaptability and responsiveness in the face of changing threat vectors. On the other hand, deep learning (DL) methods have demonstrated high effectiveness due to their ability to extract complex patterns from large volumes of data. However, the application of centralized DL models is associated with the challenges of transmitting data to a central server, which in turn increases the risk of data leaks, violates confidentiality principles, and creates additional computational and communication overheads [5].

One of the most promising solutions for overcoming these limitations is federated learning (FL). FL is a distributed machine learning method in which models are trained directly on local devices rather than through centralized data collection. This approach significantly increases the level of confidentiality, as raw data never leaves the device, and reduces the load on the network infrastructure by minimizing the amount of transferred information. In IoT environments—where each device may have its own unique memory and computational limitations—the use of FL becomes particularly relevant.

In this work, a client–server FL architecture is employed, where each IoT device (client) performs local training using its private traffic data and periodically transmits model updates to a central server. The server aggregates the received parameters to update the global model and then redistributes it back to the clients. This iterative communication and aggregation process continues over multiple rounds until model convergence is achieved. A schematic overview of the federated learning workflow is presented in Figure 1.

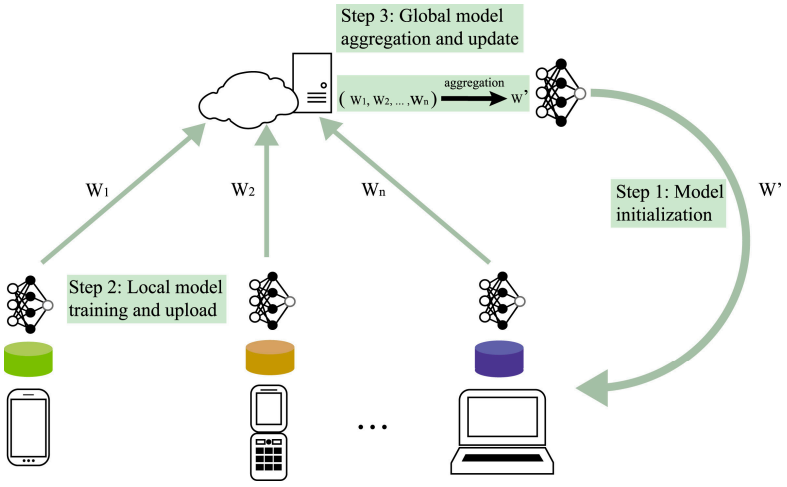


Figure 1. A schematic diagram of federated learning [6].

Despite the significant potential of federated learning to improve IoT security, most current research focuses predominantly on enhancing attack detection accuracy and developing aggregation mechanisms to protect data confidentiality. At the same time, the impact of data volume and the number of participating devices on the performance of DL models in distributed learning conditions remains insufficiently explored. The practical aspect of such scaling is crucial for deploying IDS in IoT, since edge devices have limited resources and may struggle with processing large amounts of information [7].

The objective of this work is to conduct a comprehensive experimental study aimed at determining the optimal data scale for anomaly detection in IoT networks using deep learning models under federated learning conditions. The study examines the influence of both the volume of input data and the number of devices participating in the training on the metrics of detection accuracy and model convergence. Such an approach allows us to identify the optimal conditions for deploying IDS that operate under limited computational resources and high confidentiality requirements.

Within the framework of this study, three deep learning architectures implemented in a federated environment are considered:

- Deep Neural Network (DNN)—a classical architecture capable of detecting complex patterns, albeit requiring significant computational resources.
- Convolutional Neural Network (CNN)—a model specialized in extracting local features from data, which increases training speed and improves detection quality under limited computational power.
- Hybrid Model CNN+BiLSTM—a combination that leverages convolutional layers for spatial feature extraction and Bidirectional Long Short-Term Memory (BiLSTM) for analyzing the temporal dynamics of data, thereby efficiently processing sequences of network traffic.

All three models were tested on the CICIoT 2023 dataset, which is considered one of the most modern and extensive datasets for IoT security research. This dataset covers a wide range of attack types and contains data from numerous different devices, allowing for a comprehensive evaluation of the proposed approach.

The main contributions of this paper are as follows:

- Development and implementation of an FL-oriented IDS adapted for distributed IoT networks with limited computational resources.
- A comparative evaluation of three deep learning models (DNN, CNN, and CNN+BiLSTM) for detecting cyberattacks, including an analysis of their detection accuracy and convergence time.
- The realization of the first experimental study of data scaling under federated learning conditions, which determines the optimal volume of input data for effective anomaly detection.
- Formulation of practical recommendations for data preparation and the organization of distributed learning in real IoT environments.

Thus, this research not only demonstrates the advantages of federated learning for enhancing IoT security but also significantly contributes to the understanding of how data volume and the number of devices affect attack detection quality. The obtained results allow us to formulate recommendations for the optimal use of resources in distributed systems, which is an urgent task for modern IoT platforms.

The remainder of this paper is organized as follows. Section 2 is dedicated to the literature review and analysis of existing solutions in the field of IDS for IoT and the application of federated learning. Section 3 describes the research methodology, including the characteristics of the experimental environment, data preprocessing procedures, and

the CICIoT 2023 dataset. Section 4 details the proposed model architecture and training process, while Section 5 presents the analysis of experimental results. Section 6 includes the conclusions of the study, a discussion of its limitations, and directions for future research.

2. Related Work

Ensuring security in Internet of Things (IoT) networks is a rapidly evolving research area due to the increasing number of connected devices, emerging cyber threats, and the limited computational resources of IoT systems. In this section, we review existing studies related to intrusion detection systems (IDS), anomaly detection methods, and machine learning-based approaches applied in IoT environments. Special attention is given to federated learning (FL), which has become a promising paradigm for privacy-preserving and distributed model training in resource-constrained IoT networks.

2.1. Intrusion Detection Systems and Machine Learning Approaches in IoT Networks

Intrusion Detection Systems (IDS) have become a fundamental component of network security, providing mechanisms to detect malicious activities and protect systems from various cyberattacks. Traditionally, IDS techniques are classified into two main categories: signature-based detection and anomaly-based detection. Signature-based IDS rely on predefined patterns or signatures of known attacks stored in databases [8]. While these systems are effective against previously encountered attacks, they fail to detect novel or unknown attack patterns (zero-day attacks) [9].

Anomaly-based IDS are designed to detect deviations from normal network behavior, making them more suitable for identifying new and sophisticated attacks [10]. However, anomaly-based systems often suffer from high false-positive rates, especially in dynamic and heterogeneous environments like the Internet of Things (IoT).

The integration of IDS into IoT networks presents additional challenges due to the resource-constrained nature of IoT devices, such as limited memory, processing power, and energy capacity [11]. Furthermore, the heterogeneous structure of IoT networks—consisting of diverse devices and communication protocols—complicates the design of unified security systems. Therefore, the development of lightweight and adaptive IDS solutions becomes crucial for protecting IoT environments.

To overcome the limitations of traditional IDS approaches, machine learning (ML) and deep learning (DL) techniques have been widely adopted for anomaly detection in IoT networks. ML models such as Decision Trees, Random Forest, Naïve Bayes, and Support Vector Machines (SVM) have demonstrated the ability to identify abnormal patterns in network traffic [12]. Among them, SVM is recognized for its effectiveness in anomaly detection tasks due to its ability to handle high-dimensional data and its generalization capabilities [13].

In recent years, deep learning models have shown even greater potential for intrusion detection [14]. Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN)—including Long Short-Term Memory (LSTM) networks—have been successfully applied to extract complex features from large-scale network traffic data [15]. CNNs are particularly effective in identifying local patterns, while LSTM-based models are well-suited for capturing sequential dependencies in time-series data.

Despite the promising results of ML- and DL-based IDS in IoT security, their deployment in real-world IoT environments remains a challenging task. The centralized nature of traditional ML/DL models requires collecting data on a central server, which raises privacy concerns, increases communication costs, and creates scalability issues. Moreover, training

large-scale DL models requires considerable computational resources, which may exceed the capabilities of IoT devices deployed at the network edge.

Several recent studies have reinforced the value of both classical and deep learning models for intrusion detection in IoT networks. Djenouri et al. [16] introduced emergent deep learning methods that dynamically adapt to evolving threats in heterogeneous environments. Alsamiri and Alsubhi [17] showed that traditional ML algorithms can still offer competitive performance for lightweight IDS in resource-constrained devices. Jony and Arnob [18], using the CIC-IoT2023 dataset, emphasized the importance of model-dataset alignment for realistic benchmarking. Complementing these centralized approaches, Okey et al. [19] proposed a federated CNN-GRU and LSTM-GRU ensemble that enhanced detection robustness under heterogeneous and adversarial IoT conditions.

These challenges highlight the need for decentralized and privacy-preserving approaches to anomaly detection in IoT networks, capable of operating efficiently under resource constraints. This motivates the adoption of federated learning (FL), which enables collaborative model training without sharing raw data between devices, offering a suitable solution for intrusion detection in IoT environments.

2.2. Federated Learning for IoT Security

In recent years, federated learning (FL) has gained significant attention as a privacy-preserving alternative to centralized intrusion detection methods for Internet of Things (IoT) networks. Unlike traditional architectures, FL enables IoT devices to collaboratively train machine learning models without transferring raw data to a central server, thereby reducing communication costs and mitigating privacy risks.

One of the earliest systems that demonstrated the practical applicability of FL in constrained IoT environments was DÏoT, proposed by Nguyen et al. [20]. The system achieved a detection rate of 95.6% with zero false positives in detecting Mirai-based malware in a real smart home setup. It used device-type-specific behavioral models and a federated architecture to aggregate local updates across gateways. Following this, Ferrag et al. [21] introduced Edge-IIoTset, one of the most comprehensive and realistic datasets designed for both centralized and federated learning. It encompasses 14 types of attacks across a multi-layer testbed representing diverse IoT/IIoT environments.

Despite encouraging results, FL-based approaches face critical challenges. As shown by Mármol Campos et al. [22], the inherent non-IID distribution of data and class imbalance in distributed IoT networks significantly degrade model convergence and overall detection performance. Furthermore, the resource-constrained nature of IoT devices—limited in memory, processing power, and energy—restricts the deployment of complex deep learning architectures.

To address these limitations, multiple studies have integrated lightweight deep learning architectures within FL frameworks. For instance, Ferrag et al. [23] experimentally evaluated the performance of RNN, CNN, and DNN models in federated settings using Bot-IoT, MQTTset, and ToN_IoT datasets. Their findings confirmed that with suitable architecture and data preprocessing, FL can ensure both accuracy and reliability.

However, the decentralized nature of FL introduces unique security risks, particularly model poisoning attacks. Bhagoji et al. [24] demonstrated that even a single adversarial client could compromise global model integrity by manipulating updates during the aggregation phase. This led to the development of secure aggregation techniques and differential privacy mechanisms aimed at defending against adversarial devices [25–27]. Recent studies, including those by Rey et al. [25], have confirmed that FedAvg remains vulnerable in adversarial settings, and that robust aggregation mechanisms are essential for realistic FL deployments in IoT security.

Sun et al. proposed FedMADE, a dynamic aggregation method tailored to tackle both data heterogeneity and class imbalance in federated intrusion detection systems [28]. Unlike FedAvg and its variants, FedMADE clusters local models based on class-wise probability outputs and assigns adaptive aggregation weights to each group. This enables the system to prioritize contributions from clients that demonstrate consistent performance across all attack classes, including minority ones. Evaluated on the CICIoT2023 dataset, FedMADE achieved up to 71.07% improvement in minority attack classification accuracy while maintaining robustness against poisoning attacks and incurring only a 4.7% increase in latency compared to FedAvg. Its design—requiring no changes on client devices—makes it particularly suitable for resource-constrained IoT environments. Alsaleh et al. [29] further addressed heterogeneity and communication constraints in FL-based IoT IDS by introducing a semi-decentralized framework using BiLSTM models. Their method clustered clients via autoencoder-based similarity metrics and delegated aggregation to cluster heads, significantly reducing communication overhead. Evaluated on multiple datasets, including CICIoT2023, Edge-IIoTset, and BoT-IoT, the proposed system achieved a peak F1-score of 0.705 and improved training efficiency by over 1000 s per round compared to centralized FL. These findings underscore the viability of hierarchical aggregation and lightweight recurrent architectures in practical deployments.

In a parallel effort, Devine et al. [30] proposed a federated learning framework leveraging linear Support Vector Machines (SVMs) for intrusion detection in IoT networks. Their study focused on evaluating the trade-offs between detection performance and physical constraints such as memory usage and training delay. Using the CICIoT2023 dataset, the proposed federated SVM model achieved competitive F1-scores (up to 0.981) compared to Random Forest and Artificial Neural Networks. However, it exhibited higher memory consumption per node, limiting its applicability to low-power IoT devices. Despite this, the results suggest that federated SVMs remain promising in scenarios where model interpretability and robustness outweigh memory limitations.

Another major concern is computational efficiency. Danquah et al. [26] proposed a lightweight FL model utilizing feature selection and PCA-based dimensionality reduction to minimize complexity. Their approach, built on differentially private MLPs, achieved 99.9% accuracy while reducing the computation overhead by 87.34%, making it highly suitable for smart homes and edge environments.

An additional trend is the integration of FL with complementary technologies. Javeed et al. [31] introduced a Zero Trust-based FL architecture using CNN+BiLSTM, demonstrating robust performance under adversarial and heterogeneous conditions. Their system incorporated verification and trust mechanisms during model aggregation to address the insider threat problem in FL. Similarly, blockchain integration has been proposed to ensure verifiability and decentralization in FL-based security systems, especially in industrial environments [23].

Taken together, this body of literature reflects both rapid progress and persistent challenges in deploying FL-based intrusion detection systems [32]. While advances in accuracy, privacy, and efficiency are evident, key gaps remain—particularly in terms of scalability, resilience to adversaries, and adaptation to real-world resource-constrained settings. These gaps serve as the central motivation for the present study, which aims to provide an experimental evaluation of FL-based deep learning models under realistic constraints of data volume, device count, and heterogeneity, ultimately offering deployment-ready insights for securing next-generation IoT networks.

2.3. Research Gap and Motivation

Despite significant progress in applying federated learning (FL) to intrusion detection in IoT environments, one critical research question remains insufficiently addressed:

At what scale of local data and number of devices do deep learning (DL) models trained in FL settings perform effectively in IoT networks?

This question is fundamental for the practical deployment of FL-based intrusion detection systems, especially in resource-constrained and privacy-sensitive environments. However, most existing studies focus on accuracy improvements and privacy mechanisms, often assuming ideal data volumes and ignoring the variability introduced by different numbers of participating devices.

This work directly addresses this gap by conducting a comprehensive experimental evaluation of DL models under varying data scales and client counts in a federated setting. The goal is to identify the conditions under which FL-based anomaly detection is both effective and efficient, thereby providing practical deployment guidelines for real-world IoT systems.

3. Materials and Methods

This section describes the experimental setup, including the dataset used, data preprocessing steps, federated learning configuration, evaluation metrics, and computing environment. As part of the experimental design, a brief analysis of existing IoT security datasets was conducted to justify the selection of CICIoT2023 as the primary data source. The data was preprocessed to ensure quality and balance. Federated learning was simulated with varying numbers of devices using three deep learning models. Performance was evaluated using standard classification metrics, and all experiments were conducted on a high-performance computing platform.

3.1. Dataset Overview

To ensure a reliable evaluation of the proposed federated intrusion detection system, a comparative analysis of publicly available IoT security datasets was conducted. Several commonly used datasets were reviewed, including Bot-IoT, ToN-IoT, MQTTset, IoT-23, Edge-IIoTset, and MedBIoT. These datasets differ significantly in terms of testbed realism (virtual, real, or hybrid), number and diversity of IoT devices, variety of attack types, and the availability of structured features. Earlier datasets such as Bot-IoT and MQTTset were mainly generated in virtual environments, with limited device types and outdated attack models. More recent datasets like IoT-23, CIC-IDS2017 [33], and Edge-IIoTset offer improved realism but may lack scale, diversity, or modern protocol coverage.

A comparative summary of these datasets is provided in Table 1, highlighting their individual strengths and limitations. Based on this assessment, the CICIoT2023 dataset was selected for this study due to its scale, diversity, high fidelity, and recentness.

The CICIoT2023 dataset was published by the Canadian Institute for Cybersecurity in 2023 and is among the most comprehensive and realistic intrusion detection datasets available for IoT research. It contains over 46 million labeled records generated from 105 IoT devices, including 67 IP-based and 38 Zigbee/Z-Wave smart devices. The dataset includes 33 different cyberattacks, grouped into seven categories: Distributed Denial-of-Service (DDoS), Denial-of-Service (DoS), Reconnaissance, Web-Based Attacks, Brute-Force Attacks, Spoofing, and Mirai Botnet Activity. All traffic was collected in a real-world testbed, simulating smart home and industrial environments to ensure authentic and varied traffic patterns.

Table 1. Comparison of commonly used IoT security datasets.

Dataset	Year	Testbed Setup	Devices	Attack Types	Features Provided	Notes
Bot-IoT [34]	2018	Virtual	5 simulated smart devices	DoS, DDoS, keylogging, information theft	46 features	Synthetic traffic; limited diversity
N-BaIoT [35]	2018	Real	9 physical devices	Mirai, BASHLITE variants	23 features	Real traffic; malware focus
IoT-23 [36]	2020	Real	3 physical devices	Mirai, Torii, Hajime, Trojan, etc.	Raw + logs	Good diversity; smaller scale
MedBIoT [37]	2020	Hybrid	80 virtual, 3 physical devices	Mirai, BASHLITE, Torii	Raw	Focused on medical/healthcare IoT
MQTTset [38]	2020	Virtual	10 MQTT sensors	DoS, flooding, brute force	33 features	Protocol-specific; narrow scope
ToN-IoT [39]	2020	Hybrid	7 simulated devices	Scanning, ransomware, injection, XSS	Raw + engineered	Industrial + IoT; broad attack types
Edge-IIoTset [21]	2022	Real	12 physical IoT/IIoT devices	DoS/DDoS, MiTM, malware, injection	Yes	Real traffic; limited volume
CICIoT2023 [40]	2023	Real	67 IP + 38 Zigbee/Z-Wave devices	33 attack types: DDoS, spoofing, brute force, etc.	39 engineered	High scale, protocol diversity, modern attack patterns

Each traffic record in CICIoT2023 is described using 39 engineered features that span transport-level behavior, flag distributions, protocol frequency, packet statistics, and timing patterns. These include both relative and absolute flag counts (e.g., SYN, ACK, RST), protocol-specific metrics (HTTP, DNS, SSH, etc.), and statistical measures such as Rate, IAT, Variance, and Tot Size. This comprehensive feature set provides rich contextual information and supports the training of deep learning models for both anomaly detection and fine-grained attack classification. A detailed breakdown of all features is provided in Table 2.

Table 2. CICIoT2023 feature descriptions.

No	Feature	Description
1	Header Length	Mean of transport layer header lengths
2	Time-To-Live	TTL value
3	Rate	Packet rate within aggregation window (packets/sec)
4	FIN flag number	Proportion of packets with FIN flag
5	SYN flag number	Proportion of packets with SYN flag
6	RST flag number	Proportion of packets with RST flag
7	PSH flag number	Proportion of packets with PSH flag
8	ACK flag number	Proportion of packets with ACK flag
9	ECE flag number	Proportion of packets with ECE flag
10	CWR flag number	Proportion of packets with CWR flag
11	SYN count	Count of SYN flags
12	ACK count	Count of ACK flags
13	FIN count	Count of FIN flags
14	RST count	Count of RST flags
15	IGMP	Average IGMP packets

Table 2. *Cont.*

No	Feature	Description
16	HTTPS	Average HTTPS packets
17	HTTP	Average HTTP packets
18	Telnet	Average Telnet packets
19	DNS	Average DNS packets
20	SMTP	Average SMTP packets
21	SSH	Average SSH packets
22	IRC	Average IRC packets
23	TCP	Average TCP packets
24	UDP	Average UDP packets
25	DHCP	Average DHCP packets
26	ARP	Average ARP packets
27	ICMP	Average ICMP packets
28	IPv	Average IPv packets
29	LLC	Average LLC packets
30	Tot Sum	Total packet size within window
31	Min	Minimum packet size
32	Max	Maximum packet size
33	AVG	Average packet size
34	Std	Standard deviation of packet size
35	Tot Size	Mean packet size
36	IAT	Inter-arrival time between packets
37	Number	Number of packets per window
38	Variance	Variance of packet sizes
39	Protocol Type	Most frequent protocol observed (mode)

Further inspection of CICIoT2023 reveals that a substantial portion of the dataset is composed of DDoS attack traffic, which covers a broad set of protocols and flooding techniques. Figure 2 illustrates the distribution of DDoS attack subtypes by number of instances. Prominent examples such as ICMP_Flood, UDP_Flood, TCP_Flood, and PSHACK_Flood are represented by millions of records each. In total, over 50% of malicious traffic in the dataset falls under the DDoS category. This abundance of DDoS-specific data ensures sufficient volume and diversity for training effective models and justifies the decision to focus primarily on DDoS detection in this study.

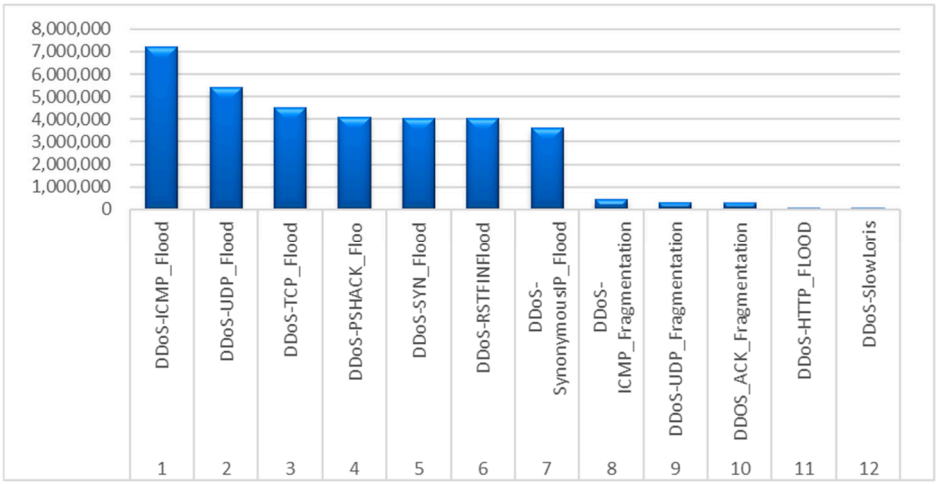


Figure 2. Distribution of DDoS attack types by record count in the CICIoT2023 dataset.

3.2. Data Preprocessing

Preprocessing is a critical stage in ensuring data quality and consistency for training deep learning models, especially in federated environments. The CICIoT2023 dataset required substantial preprocessing due to its scale and heterogeneity. The complete workflow, shown in Figure 3, includes the following six steps:

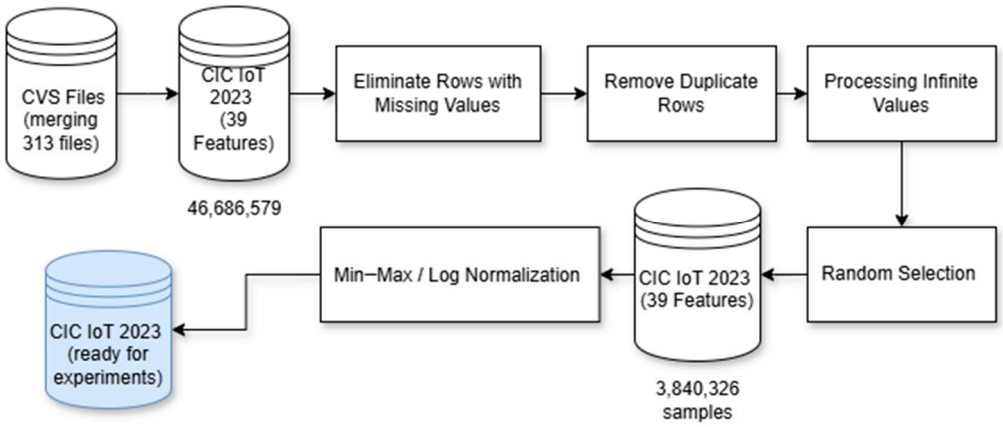


Figure 3. Structured preprocessing stages for the CICIoT2023 dataset: merging, cleaning, reduction, and normalization.

1. Merging CSV Files

The original dataset consists of 313 CSV files, each capturing specific attack types or benign traffic. These were merged into a unified dataset containing 46,686,579 records and 39 structured features, enabling consistent processing and integration into the modeling pipeline.

2. Handling Missing and Infinite Values

All rows with missing values (NaN) were removed to maintain data integrity. Additionally, several features contained infinite values, which were replaced by the maximum finite value within the same feature. This helped prevent distortion of feature distributions and ensured numerical stability during training.

3. Removing Duplicate Records

Duplicate entries, including both exact duplicates and near-duplicates (detected using similarity metrics), were removed to reduce redundancy, mitigate overfitting risks, and ensure a more representative and diverse training set.

4. Dataset Reduction via Random Sampling

To reduce computational overhead, a stratified random sampling method was applied, resulting in a dataset of 3,840,326 records. We selected 250,000 records for benign traffic, approximately 1,100,000 records for non-DDoS attacks due to the large number of subclasses, and 2,500,000 records for DDoS attacks by including 10 distinct DDoS classes with 250,000 records each. This approach ensured balanced class representation and preserved diversity across traffic types for robust model training.

5. Normalization

- Logarithmic Normalization

The Rate feature exhibited an extreme long-tailed distribution, potentially hindering model convergence. To mitigate this, a logarithmic transformation was applied to compress outliers and stabilize variance, as shown in Equation (1):

$$x' = log(x + 1), \tag{1}$$

- Min–Max Scaling

After log normalization, min–max scaling was applied to all features, mapping values to the [0, 1] range to ensure scale uniformity and improve learning stability. The normalization formula is provided in Equation (2):

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{2}$$

These preprocessing steps produced a clean, numerically stable, and computationally tractable dataset, enabling effective training of deep learning models in a federated learning context while maintaining the integrity and diversity of the original traffic data.

3.3. Federated Learning Setup

To evaluate the applicability and scalability of the intrusion detection system in decentralized IoT networks, a federated learning (FL) environment was simulated using the preprocessed CICIoT2023 dataset. This section outlines the key components of the FL setup: (1) client simulation, (2) data distribution, (3) model architectures, (4) training parameters, and (5) model aggregation.

1. Client Simulation and Data Distribution

Federated learning was conducted across four experimental configurations involving 10, 50, 100, and 150 devices. A uniform total of approximately 3.8 million records from the CICIoT2023 dataset was consistently used across all experiments. For clarity and consistency, the dataset was equally partitioned among clients—for example, in the 100-client scenario, each device received exactly 38,000 records.

To realistically simulate the decentralized and heterogeneous nature of IoT networks, data was intentionally partitioned in a non-IID (non-identically independently distributed) manner. Each client received a distinct combination of benign and malicious traffic, deliberately varying in attack types and their relative proportions. Specifically, certain devices predominantly received benign traffic with sparse malicious samples, while others were assigned predominantly malicious traffic, simulating the inherent heterogeneity observed in real-world IoT deployments. This partitioning method was implemented through randomized but controlled sampling to ensure representative diversity across the federated environment.

Although the equal number of records per client facilitated controlled comparative analyses, we recognize that real-world IoT environments typically exhibit unequal data distributions due to differences in device roles, usage patterns, and connectivity constraints. Such realistic scenarios might significantly impact model convergence, accuracy, and overall performance. Therefore, future research will extend the current study by incorporating dynamically imbalanced and evolving data distributions, closely aligning experiments with practical deployment conditions. Each configuration in this study was treated as an independent experiment, initialized from scratch, to objectively evaluate the impact of these variables on model performance.

2. Model Architectures

The following deep learning models were employed and configured with the hyperparameters summarized in Table 3. Three deep learning architectures were evaluated:

- DNN (Deep Neural Network): A fully connected neural network with three hidden layers consisting of 128, 64, and 32 neurons, respectively, using ReLU activation and dropout regularization (0.3) after each hidden layer. This model served as a baseline for comparison. Although standard ReLU was used due to its simplicity and effectiveness, alternative activation functions like Leaky ReLU or Parametric ReLU (PReLU) could potentially improve model performance by mitigating neuron saturation and dying gradient problems, thus enhancing training stability and final accuracy [41]. Future

research will investigate these activation variants within the context of federated intrusion detection in IoT networks.

- CNN (Convolutional Neural Network): This architecture includes two one-dimensional convolutional layers with 64 and 32 filters (kernel size = 3), followed by pooling operations, global average pooling, and a dense classification head. CNNs are well-suited for identifying local correlations among features.
- CNN+BiLSTM: A hybrid model combining the same CNN block as above with a Bidirectional Long Short-Term Memory (BiLSTM) layer with 64 units to capture temporal patterns. This model is particularly suited for analyzing sequential behaviors in network traffic.

Table 3. Model architectures and hyperparameters.

Model	Hyperparameters	Value
DNN	Hidden layers	64, 64, 32
	Activation function	ReLU
	Output activation	Softmax
	Optimizer	Adam
	loss func	Sparse categorical crossentropy
	Learning rate	0.001
	Input dim	variable
CNN	Conv1D blocks	(64, 64), (128)
	Conv1D kernel size	3
	Dropout rates	0.2, 0.3, 0.3
	fc layer	128
	Learning rate	0.001
	Output activation	softmax
	Optimizer	Adam
	Batch norm	True
	loss func	Sparse categorical crossentropy
	Input dim	variable
CNN+BiLSTM	reshape shape	(input_dim, 1)
	Conv1D filters	32, 64
	Conv1D kernel size	32, 64
	LSTM units	64, 16
	Dropout lstm	0.2
	dropout conv	0.2
	Learning rate	0.001
	Output activation	softmax
	loss func	Sparse categorical crossentropy
	Metrics	accuracy
	Input dim	variable

All models accept a 39-dimensional normalized input vector and use a softmax output layer for multiclass classification. These architectures were selected to evaluate both baseline and advanced learning strategies, considering the constraints of resource-limited IoT devices.

3. Training Parameters and Rounds

Each configuration (10, 50, 100, and 150 devices) was trained for 20 communication rounds in the federated learning setting. For local training on each client, the following hyperparameters were used consistently across all configurations and models: Adam optimizer, learning rate of 0.001, batch size of 1024, and 10 local epochs per round.

To ensure a fair comparison between centralized and federated training, all model architectures and training parameters were kept identical in both settings. The only differ-

ence was in the training mode: centralized training was conducted with the full dataset over 10 epochs in a single global round, while federated training used partitioned local datasets with periodic aggregation over 20 rounds.

4. Model Aggregation

The standard Federated Averaging (FedAvg) algorithm was used for global aggregation. After local training, devices sent their updated weights to a central server, which computed a weighted average (based on the size of local datasets) and redistributed the updated global model to devices for the next round.

This staged simulation allowed for a thorough investigation of how the number of devices affects the performance and convergence behavior of federated learning, as well as the resilience of different model architectures in distributed environments.

3.4. Experimental Environment

All experiments were conducted in a consistent and controlled computing environment to ensure fair performance comparison and reproducibility. The hardware configurations for centralized training and edge-based federated inference are provided in Tables 4 and 5, respectively, while the software libraries used in all experiments are summarized in Table 6. Data preprocessing, federated learning simulation, and model training were performed using TensorFlow-GPU on a single high-performance workstation. To emulate a realistic federated setup, multiple clients were simulated in memory without the use of distributed infrastructure.

Hardware acceleration was enabled using TensorFlow 2.10.1 configured with CUDA 12.6, executed on an NVIDIA RTX 4090 GPU. This configuration ensured efficient training of deep learning models across multiple communication rounds.

Table 4. Hardware specifications for centralized model training.

Component	Specification
Operating System	Windows 11 Pro
CPU	Intel(R) Core (TM) i9-13900K
GPU	NVIDIA GeForce RTX 4090
Memory	128 GB RAM
Storage	4 TB SSD
Python Version	3.9.19
CUDA Toolkit	12.6
Deep Learning Framework	TensorFlow-GPU 2.10.1

Table 5. Hardware specifications for federated inference on edge device (Raspberry Pi 5).

Component	Specification
Device	Raspberry Pi 5
Processor	Broadcom BCM2712—Quad-core 64-bit ARM Cortex-A76 @ 2.4 GHz
Cache	L2: 512 KB per core; L3: 2 MB shared
Memory	8 GB LPDDR4 RAM
Storage	128 GB microSD (UHS-I)
OS	Raspberry Pi OS (64-bit)

Table 6. Software libraries used in experiments.

Library	Purpose of Library
tensorflow 2.18.0	Core deep learning framework for model definition and training
numpy 2.0.2	Data handling, transformations, and tensor manipulation
socket	Data handling, transformations, and tensor manipulation
pickle	Serialization of model weights for transfer
threading	Concurrent client handling on the server
os	File operations and environment variable handling
time	Communication timing and logging delays
logging	Structured output and debugging
sklearn 1.6.1	Metrics calculation, preprocessing, data splitting (optional)
websockets 15.0	Reserved for future implementation of async client-server communication

3.5. Evaluation Metrics

To evaluate the performance of intrusion detection models under the federated learning setup, several standard classification metrics were employed [42]. Given the multiclass nature of the CICIoT2023 dataset and the imbalanced distribution of attacks, the following metrics were selected to provide a comprehensive assessment of model performance:

- **Accuracy:** The ratio of correctly predicted instances to the total number of instances. Although commonly used, it may be misleading in imbalanced settings.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

- **Precision:** The ratio of true positives to all predicted positives. It measures the model's ability to avoid false alarms.

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

- **Recall (Detection Rate):** The proportion of actual positives correctly identified. In the context of IDS, this corresponds to the detection rate.

$$R = \frac{TP}{TP + FN} \tag{5}$$

- **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure in the presence of class imbalance.

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{6}$$

- **Confusion Matrix:** A matrix representation of actual vs. predicted classifications across all classes, used for detailed analysis of per-class performance.
- **Macro-Averaged Metrics:** Since the dataset contains multiple attack types with varying frequencies, macro-averaged precision, recall, and F1-score were used to give equal weight to each class, regardless of sample size.

These metrics were computed at the end of each experiment using the global model after 10 communication rounds. For additional interpretability, confusion matrices were plotted for each model to visualize misclassifications between attack categories.

4. Results

This section presents the core experimental outcomes obtained from training three deep learning models—DNN, CNN, and CNN+BiLSTM—using the CICIoT2023 dataset under a federated learning setup. Each model was evaluated independently across three federated configurations involving 10, 50, 100, and 150 devices. All models were trained for 10 communication rounds, with each device receiving a non-identical partition of the preprocessed and balanced dataset containing 3.8 million records.

Model performance was evaluated using the standard classification metrics: accuracy, precision, recall, and F1-score, computed globally on the aggregated test set after the final communication round. Accuracy measures the proportion of correctly predicted samples. Precision reflects the ability to avoid false positives, recall captures the true positive rate (detection rate), and the F1-score represents the harmonic mean of precision and recall. For fairness across multiple classes and imbalanced data, macro-averaged versions of these metrics were used.

As shown in Figure 4, the DNN model achieved its highest accuracy of approximately 94.95% in the 10-device configuration, converging within 2–3 communication rounds. For 50 and 100 devices, the model reached accuracies of 94.57% and 92.15%, respectively, with slightly slower convergence around 5–10 rounds. In the most distributed setting with 150 devices, convergence was noticeably slower, requiring nearly the full 20 rounds to reach a final accuracy of 93.65%. This trend reflects the increased communication and data heterogeneity challenges in large-scale federated environments.

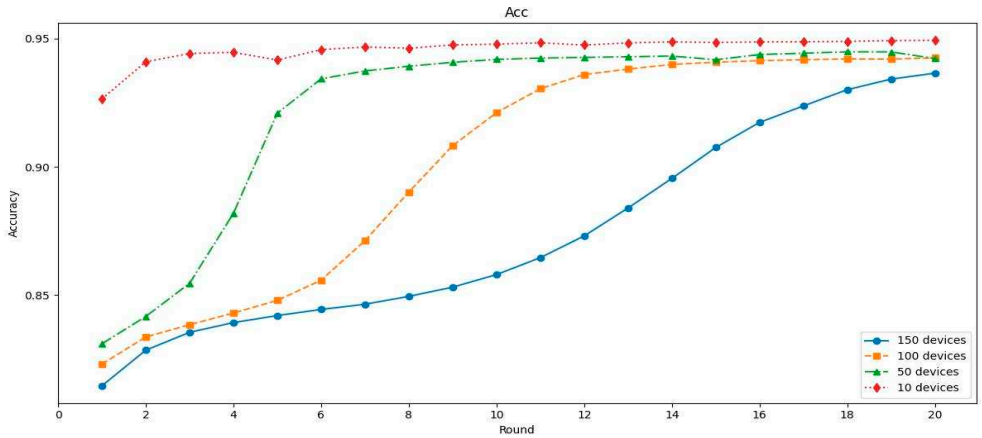


Figure 4. Dependence of DNN model accuracy on the number of IoT nodes.

The CNN model demonstrated rapid convergence and stable high accuracy across all configurations. As shown in Figure 5, the model reached above 94.9% accuracy within the first 2–4 communication rounds in all configurations (10, 50, 100, and 150 devices). Notably, even in the 150-device setup, which is typically more prone to performance degradation due to data heterogeneity and communication delays, the CNN model converged almost as quickly as in smaller setups. This demonstrates the model's strong generalization ability and resilience under federated constraints.

The hybrid CNN+BiLSTM model demonstrated strong overall performance, particularly in lower-device configurations. As shown in Figure 6, it exceeded 94.9% accuracy within the first 2–3 rounds in the 10-device setting and maintained high stability throughout training. For 50 devices, the model reached similar performance by round 8. However, in more distributed settings with 100 and 150 devices, convergence was slower, requiring over 15 rounds to surpass 93% accuracy. Despite this, the model consistently improved across rounds, reflecting its capacity to generalize under varying levels of data distribution and communication constraints.

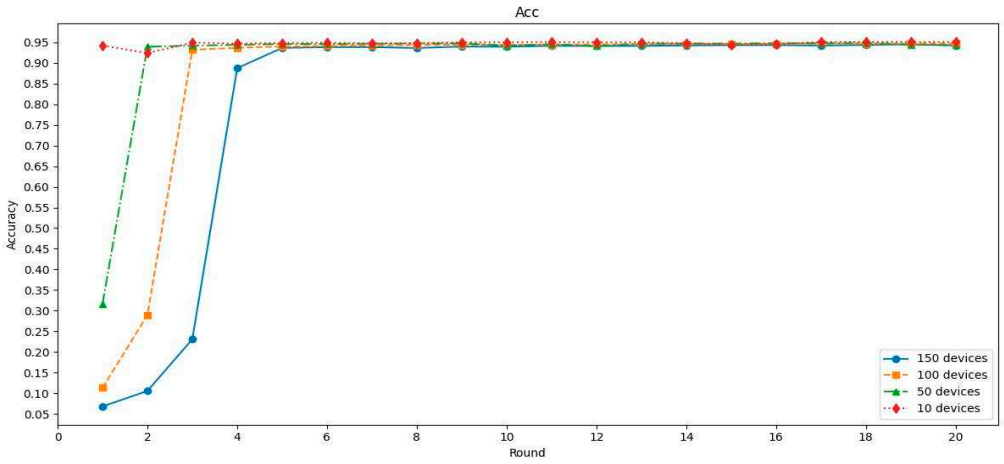


Figure 5. Dependence of CNN model accuracy on the number of IoT nodes.

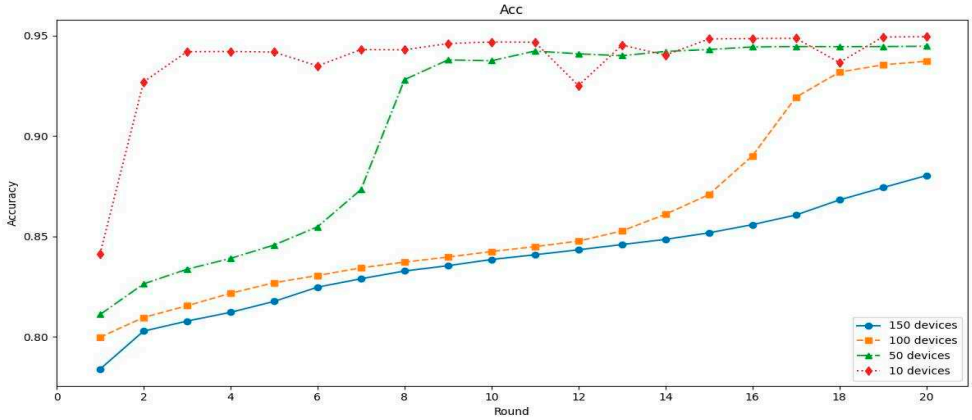


Figure 6. Dependence of CNN+BiLSTM model accuracy on the number of IoT nodes.

To assess the per-class detection performance, a normalized confusion matrix was generated for the CNN+BiLSTM model (Figure 7). The results indicate that the model accurately identified most network traffic categories, achieving over 99% class-wise precision for dominant types such as Benign, DDoS ICMP Flood, UDP Flood, and TCP Flood. However, confusion was observed between structurally similar classes—particularly between DDoS SYN Flood and DDoS SynonymousIP Flood—as well as within fragmentation-based and non-DDoS categories. Despite these overlaps, most of the off-diagonal values remained under 1%, demonstrating the model's strong generalization and fine-grained classification capability, even in complex federated settings.

To enable comparison, each model was additionally trained using a centralized approach, where the entire dataset was made available to a single learner without partitioning. These results, which have been included in Table 7, are intended to represent upper-bound performance baselines and to illustrate the potential impact of decentralization in federated environments. As anticipated, slightly higher accuracy and F1-scores were achieved under centralized training, owing to complete data visibility and the absence of inter-device communication noise. Nevertheless, only minor degradation was observed in the federated CNN model, reinforcing its resilience and effectiveness in distributed intrusion detection scenarios.



Figure 7. Confusion matrix illustrates the performance of the CNN+BiLSTM model in classifying normal traffic and different types of DDoS attacks.

Table 7. Macro-averaged performance metrics across device configurations and centralized baselines.

Model	Devices	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DNN	10	**94.95**	**96.09**	**94.95**	**94.45**
	50	94.57	94.62	94.57	94.51
	100	92.15	93.40	92.15	92.32
	150	92.58	92.90	92.58	92.58
	Centralized	**94.99**	**94.09**	**93.60**	**93.29**
CNN	10	**95.27**	**96.07**	**95.27**	**94.92**
	50	94.98	95.71	94.98	94.59
	100	93.88	94.01	93.88	93.81
	150	93.98	95.26	93.98	93.32
	Centralized	**95.10**	**94.05**	**93.67**	**93.48**
CNN+BiLSTM	10	**94.97**	**95.85**	**94.97**	**94.58**
	50	94.57	94.77	94.57	94.41
	100	92.93	93.79	92.93	92.93
	150	87.82	89.48	87.82	87.83
	Centralized	**94.84**	**94.31**	**93.42**	**92.96**

To assess the practical deployability of our proposed intrusion detection models in real-world conditions, we conducted a hardware-level evaluation using a Raspberry Pi 5 device.

This setup was chosen to emulate a realistic edge environment typical of IoT gateways and embedded systems, where computational and thermal constraints are critical factors.

Each of the trained models—DNN, CNN, and CNN+BiLSTM—was deployed locally on the Raspberry Pi 5 and evaluated across four federated configurations (10, 50, 100, and 150 clients). During each configuration, we measured average CPU and RAM utilization, device temperature, local training time per round, inference latency per sample, model size, number of parameters, and the size of local datasets.

The results, presented in Table 8, clearly demonstrate that the CNN model provides the best balance between detection performance and resource consumption. It maintained a low inference latency (~1.4 ms/sample), stable temperature levels (<72 °C), and moderate memory usage, making it well-suited for on-device deployment. While the CNN+BiLSTM model achieved slightly higher accuracy, it incurred significant computational overhead and thermal stress, exceeding safe operating thresholds on the device. The DNN model, although highly efficient in terms of speed and memory, delivered lower detection accuracy, which may be insufficient in security-critical scenarios.

Table 8. Federated learning model benchmark on Raspberry Pi 5: resource consumption and inference performance across configurations.

Model	Series	CPU (%)	RAM (%)	Temp (deg C)	Time per Round (s)	Time Inference (ms/Sample)	Size (KiB)	Local Samples	Local Data Size (mb)	Param Number
DNN	150	80.9	20.3	52.9	28.34	88.709	144.1	25577	8.16	9196
	100	67.5	25	57.1	40.4	114.368		38364	12.2	
	50	67.5	21	57.6	54.1	120.111		76730	24.4	
	10	68	24.8	58.9	395.1	105.238		383649	122	
CNN	150	91.7	23.2	71.6	250.9	104.03	732.6	25577	8.16	56396
	100	92.5	26.2	70.5	305.9	107.179		38364	12.2	
	50	92	29.2	76	539.3	107.166		76730	24.4	
	10	91.9	39.2	70.5	2781.2	109.375		383649	122	
CNN+BiLSTM	150	91	35.8	70	558	114.604	1144	25577	8.16	91340
	100	91.8	37.5	76.5	669.5	131.125		38364	12.2	
	50	91.5	40.4	76	1515.7	91.135		76730	24.4	
	10	92.5	49.3	85.9	6666.8	107.396		383649	122	

These findings confirm that our federated CNN-based intrusion detection architecture is not only effective in simulation but also robust and feasible for implementation on real-world IoT hardware platforms.

5. Discussion

Figures 5 and 6 illustrate the accuracy progression over communication rounds for the CNN and CNN+BiLSTM models under different network scales, revealing notable differences in convergence speed across both model complexity and number of IoT devices. The CNN architecture consistently achieved near-peak detection accuracy (~95%) within 5–7 communication rounds for federations of up to 150 devices. In contrast, the more complex CNN+BiLSTM model required around 10–15 rounds to attain comparable performance, exceeding 94% accuracy by round 15 for the 150-device configuration. This delay can be attributed to the hybrid model's increased complexity (due to sequential BiLSTM layers), which demands more iterative updates to fully optimize. From a practical perspective, the faster convergence of the CNN-based IDS is advantageous: fewer communication rounds translate to reduced inter-device communication overhead, lower cumulative energy consumption for battery-powered IoT nodes, and a shorter training

duration before the model becomes effective. These efficiencies are critical in large-scale IoT deployments where communication bandwidth and energy resources are constrained. Notably, this analysis of convergence behavior across varying network scales contributes valuable insights into the scalability and training efficiency of FL-based intrusion detection in IoT networks.

To explain the mechanisms behind performance variation with respect to the number of clients and local data sizes, we observe that increasing the number of clients reduces the volume and diversity of local data per device. This reduction hinders the local model's ability to generalize, especially when clients do not encounter all traffic classes uniformly. Consequently, federated aggregation may underrepresent rare or complex attack patterns, leading to slower convergence and higher misclassification. This behavior was evident under the 150-device scenario, where local datasets were smaller and non-IID.

These results, while specific to intrusion detection in IoT networks, may extend to other classification tasks involving sequential and heterogeneous data—such as anomaly detection in industrial control systems or smart grid environments. However, their generalizability depends on the similarity of traffic patterns and attack behaviors. Tasks involving structured or protocol-specific behaviors not well represented in CICIoT2023 may require model adaptation.

As shown in Figure 7, the CNN+BiLSTM model achieved strong classification performance on major traffic categories such as Benign, DDoS ICMP Flood, UDP Flood, and TCP Flood, with class-wise precision exceeding 98%. This confirms the model's ability to generalize well across a range of volumetric DDoS attacks and normal network traffic. The model's architecture, which combines convolutional layers for local pattern recognition with BiLSTM for capturing temporal dependencies, enables it to extract both spatial and sequential characteristics that are critical for distinguishing between malicious and legitimate behaviors.

However, the confusion matrix also highlights certain areas where classification accuracy deteriorates. The most prominent confusion occurred within the non-DDoS class, which includes a variety of attacks with heterogeneous behaviors and feature distributions. Since this category groups together multiple types of low-volume or protocol-based attacks (e.g., scanning, brute force), the model may struggle to develop a consistent decision boundary across its subcomponents. As a result, some non-DDoS samples were misclassified as either benign or DDoS traffic, particularly under the 150-device configuration, where the available data per client was relatively limited and imbalanced.

Another source of misclassification was observed between the DDoS SYN Flood and DDoS SynonymousIP Flood classes. These attacks exhibit highly similar traffic characteristics, including packet size distributions, flag patterns (e.g., repeated SYN), and transmission intervals. The overlapping temporal and structural features of these attack types likely contributed to the confusion observed in the matrix. It is important to note that even for a human analyst, differentiating such closely related DDoS variants based solely on flow-level features would be challenging. This limitation underscores the need for future work on integrating more protocol-specific or behavioral context into federated intrusion detection models.

Additionally, the confusion matrix illustrates the effect of label imbalance and class density on classification performance. Classes with fewer examples or higher intra-class variability tend to experience more misclassification. This is especially relevant in federated learning scenarios with non-IID data distributions, where not all devices encounter all traffic types uniformly. Some devices may be entirely unaware of specific attack classes, limiting the effectiveness of local training and resulting in underrepresentation of certain patterns in the global model.

Despite these challenges, the confusion matrix confirms that the CNN+BiLSTM model is robust across most attack categories and resilient to noise introduced by federated learning at scale. The extremely low off-diagonal values, particularly for critical classes like Benign and high-volume DDoS attacks, suggest strong potential for practical deployment in IoT networks. However, the identified weaknesses, especially for complex or composite attack classes, highlight opportunities for further architectural refinements, such as attention mechanisms or meta-learning strategies that can enhance per-class discrimination without compromising scalability.

The experimental results demonstrate that a federated learning-based IDS can maintain high detection performance even as the number of participating IoT devices scales up to 150. Across all tested models—DNN, CNN, and CNN+BiLSTM—classification accuracy remained above 92% even at the largest scale. The CNN+BiLSTM consistently achieved the highest accuracy (up to ~95%), followed by the CNN (~94%) and then the DNN (~93%). This ranking reflects each model's capacity: the hybrid CNN+BiLSTM excelled at capturing complex attack patterns (e.g., subtle sequential features), whereas the simpler DNN, while faster to converge, missed some nuances of multi-class traffic. Notably, all models successfully converged within a reasonable number of communication rounds (10–20) even at larger scales, indicating robust scalability of the federated training process. These findings suggest that, given a sufficient total data volume, FL-based IDS can achieve reliable anomaly detection in moderately distributed IoT networks.

A key novelty of this work is the systematic evaluation of model scalability under realistic IoT data distributions. We explicitly simulated moderately large-scale IoT deployments by distributing a substantial dataset (approximately 3.8 million CICIoT2023 flows) across up to 150 devices. This approach allowed us to observe how detection performance holds up as local data availability diminishes at larger scales. Importantly, even with only tens of thousands of samples per device, the CNN model maintained macro-averaged accuracy and F1-scores above 93%, approaching the centralized baseline (F1 ~93.5–95%) and underscoring the viability of federated IDS on data-constrained edge nodes. By contrast, the more complex CNN+BiLSTM achieved slightly higher metrics (~95% F1) but incurred greater computational cost, while the lightweight DNN exhibited slightly lower accuracy with the benefit of faster convergence and reduced overhead. As summarized in Table 7, performance differences became more evident under increased scale and were contrasted with centralized results to quantify the cost of decentralization.

Beyond classification accuracy, we evaluated each model's resource demands and runtime characteristics to assess their deployability in real-world scenarios. Table 8 provides empirical results collected from Raspberry Pi 5 edge devices, detailing inference speed, memory usage, CPU load, temperature, and local training time. The DNN model incurred the lowest computational overhead (~0.9 ms/sample, minimal RAM and CPU load), but with reduced accuracy. The CNN+BiLSTM model was the most computationally intensive (~3.9 ms/sample, peak temperature ~86 °C), which may limit its use on constrained hardware. The CNN architecture demonstrated the best balance overall, maintaining ~1.4 ms/sample inference time, stable thermal characteristics, and strong classification performance—making it the most practical choice for edge-based FL-IDS deployments.

The class-wise confusion matrix provides further insight into model behavior. The CNN+BiLSTM accurately classified the major traffic categories—benign traffic and high-volume DDoS attacks—with high precision and recall. However, certain infrequent or composite attack classes proved more challenging. Notably, the broad "non-DDoS" category (encompassing diverse low-volume attacks) saw higher misclassification rates, with some samples erroneously labeled as benign or DDoS. Likewise, very similar attack types—e.g., DDoS SYN Flood vs. DDoS Synonymous IP Flood—were occasionally confused, reflecting

the difficulty of distinguishing closely related behaviors using only flow-level features. These rare-class errors highlight opportunities for model improvement through more fine-grained features or attention mechanisms.

Finally, we acknowledge several limitations of our study and outline directions for future work. First, our experiments were conducted in a controlled offline setting using a static IoT dataset partitioned across devices. This approach, while useful for evaluation, does not fully capture the challenges of a live deployment where network conditions, device availability, and traffic patterns may change over time. Deploying the federated IDS on a real IoT testbed would be a valuable next step to assess performance under dynamic conditions, including potential concept drift in attack behaviors. Second, our federated learning setup assumed honest and reliable devices—adversarial scenarios such as model poisoning were not considered. Future research should investigate robust aggregation techniques and defenses against such adversarial behaviors to ensure IDS resilience. Additionally, our current implementation uses synchronous communication rounds; extending the approach to asynchronous or personalized federated learning could improve adaptability in heterogeneous IoT environments. Despite these limitations, our findings provide strong evidence that a federated deep learning approach to IoT intrusion detection is practically viable. The insights obtained through large-scale experimentation offer practical guidelines for deploying IDS models on real devices, advancing the security of IoT ecosystems.

6. Conclusions

This study demonstrates that federated learning enables accurate and scalable intrusion detection in IoT environments, even on resource-constrained devices. Through systematic evaluation using the CICIoT2023 dataset, we found that the CNN model consistently achieved a strong balance between accuracy (~94–95%) and computational efficiency, making it well-suited for edge deployment. While the hybrid CNN+BiLSTM model offered slightly higher accuracy (~95–96%), it incurred significantly higher inference latency and resource usage. The DNN model, while lightweight, underperformed in detecting complex multi-class traffic.

Our scaling experiments further revealed that as the number of clients increases, performance is preserved provided each client receives a sufficient volume of local data. Beyond a certain point, however, additional data yields diminishing returns, suggesting an optimal range of training volume per device. The federated approach scaled effectively up to 150 clients, maintaining model convergence and accuracy, thereby confirming its applicability in moderately large IoT networks.

Future work will extend this baseline by exploring more robust and adaptive training strategies, including personalized and asynchronous federated learning to address client heterogeneity and availability. In addition, real-world deployment on live IoT infrastructures with dynamic traffic and concept drift will help assess the system's resilience under practical constraints. Investigating adversarial robustness and defense mechanisms (e.g., against model poisoning) will also be essential for securing FL-based IDS in untrusted environments.

Author Contributions: Conceptualization, N.A. and Y.T.; methodology, N.A., Z.K. and Z.A.; software, Y.A. and Y.T.; validation, R.U., K.G. and Y.A.; formal analysis, Y.T.; investigation, N.A., Y.A. and Y.T.; resources, Y.T.; data curation, N.A., Z.K. and Z.A.; writing—original draft preparation, N.A. and Y.T.; writing—review and editing, N.A. and Y.T.; visualization, Y.T. and Y.A.; supervision, K.G., Z.K. and Z.A.; project administration, R.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CiCIoT2023 dataset used in this study is publicly available at https://www.unb.ca/cic/datasets/iotdataset-2023.html (accessed on 22 April 2025). Additional data supporting the findings of this study are available from the corresponding authors upon reasonable request.

Acknowledgments: The authors would like to thank Djamel Djenouri for contributing to this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DNN	Deep Neural Network
CNN	Convolutional Neural Network
BiLSTM	Bidirectional Long Short-Term Memory
SVM	Support Vector Machines
RNN	Recurrent Neural Network
FedMADE	Federated Masked Autoencoder for Distribution Estimation
IDS	Intrusion Detection System
FL	Federated Learning
DL	Deep Learning
ML	Machine Learning
FedAvg	Federated Averaging
IoT	Internet of Things
CICIoT2023	Canadian Institute for Cybersecurity IoT Dataset 2023
DDoS	Distributed Denial-of-Service
DoS	Denial-of-Service
GRU	Gated Recurrent Unit
PCA	Principal Component Analysis
MLP	Multilayer Perceptron
NaN	Not a Number

References

1. Dubey, K.; Dubey, R.; Panedy, S.; Kumar, S. A review of IoT security: Machine learning and deep learning perspective. *Procedia Comput. Sci.* **2024**, *235*, 335–346. [CrossRef]
2. Bakhsh, S.A.; Khan, M.A.; Ahmed, F.; Alshehri, M.S.; Ali, H.; Ahmad, J. Enhancing IoT network security through deep learning-powered intrusion detection system. *Internet Things* **2023**, *24*, 100936. [CrossRef]
3. Hizal, S.; Cavusoglu, U.; Akgun, D. A novel deep learning-based intrusion detection system for IoT DDoS security. *Internet Things* **2024**, *28*, 101336. [CrossRef]
4. Delwar, T.S.; Aras, U.; Mukhopadhyay, S.; Kumar, A.; Kshirsagar, U.; Lee, Y.; Singh, M.; Ryu, J.-Y. The intersection of machine learning and wireless sensor network security for cyber-attack detection: A detailed analysis. *Sensors* **2024**, *24*, 6377. [CrossRef]
5. Thakkar, A.; Ritika, L. A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, 783 security issues, and challenges. *Arch. Comput. Methods Eng.* **2021**, *28*, 3211–3243. [CrossRef]
6. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775. [CrossRef]
7. Sharma, S.; Kumar, V.; Dutta, K. Multi-objective optimization algorithms for intrusion detection in IoT networks: A systematic review. *Internet Things Cyber-Phys. Syst.* **2024**, *4*, 258–267. [CrossRef]
8. Ahmad, R.; Wazirali, R.; Abu-Ain, T. Machine learning for wireless sensor networks security: An overview of challenges and issues. *Sensors* **2022**, *22*, 4730. [CrossRef]
9. Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. [CrossRef]
10. Jony, A.I.; Arnob, A.K.B. A long short-term memory based approach for detecting cyber attacks in IoT using CIC-822 IoT2023 dataset. *J. Edge Comput.* **2024**, *3*, 28–42. [CrossRef]

11. Sharma, S.B.; Bairwa, A.K. Leveraging AI for Intrusion Detection in IoT Ecosystems: A Comprehensive Study. *IEEE Access* **2025**, *13*, 66290–66317. [CrossRef]
12. Aljabri, M.; Alahmadi, A.A.; Mohammad, R.M.A.; Alhaidari, F.; Aboulnour, M.; Alomari, D.M.; Mirza, S. Machine learning-based detection for unauthorized access to IoT devices. *J. Sens. Actuator Netw.* **2023**, *12*, 27. [CrossRef]
13. Haque, S.; El-Moussa, F.; Komninos, N.; Muttukrishnan, R. A systematic review of data-driven attack detection trends in IoT. *Sensors* **2023**, *23*, 7191. [CrossRef]
14. Liao, H.; Murah, M.Z.; Hasan, M.K.; Aman, A.H.M.; Fang, J.; Hu, X.; Khan, A.U.R. A survey of deep learning technologies for intrusion detection in Internet of Things. *IEEE Access* **2024**, *12*, 4745–4761. [CrossRef]
15. Shukla, K.A.; Ahamad, S.; Rao, G.N.; Al-Asadi, A.J.; Gupta, A.; Kumbhkar, M. Artificial intelligence assisted IoT data intrusion detection. In Proceedings of the 2021 4th International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 16–17 December 2021; pp. 330–335.
16. Djenouri, Y.; Laouid, A.; Derdour, D.; Badache, N. Emergent deep learning for anomaly detection in Internet of Everything. *IEEE Internet Things J.* **2023**, *10*, 3206–3219. [CrossRef]
17. Alsamiri, J.; Alsubhi, K. Internet of Things cyber attacks detection using machine learning. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 627–635. [CrossRef]
18. Jony, A.I.; Arnob, A.K.B. Securing the Internet of Things: Evaluating machine learning algorithms for detecting IoT cyberattacks using CIC-IoT2023 dataset. *Int. J. Inf. Technol. Comput. Sci.* **2024**, *16*, 56–65. [CrossRef]
19. Okey, O.D.; Rodriguez, D.Z.; Kleinschmidt, J.H. Enhancing IoT Intrusion Detection with Federated Learning-Based CNN-GRU and LSTM-GRU Ensembles. In Proceedings of the 2024 19th International Symposium on Wireless Communication Systems (ISWCS), Rio de Janeiro, Brazil, 14–17 July 2024; pp. 1–6.
20. Nguyen, T.D.; Marchal, S.; Miettinen, M.; Fereidooni, H.; Asokan, N.; Sadeghi, A.-R. DÏoT: A Federated Self-learning Anomaly Detection System for IoT. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; pp. 756–767.
21. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access* **2022**, *10*, 40281–40306. [CrossRef]
22. Campos, E.M.; Saura, P.F.; González-Vidal, A.; Hernández-Ramos, J.L.; Bernabé, J.B.; Baldini, G.; Skarmeta, A. Evaluating federated learning for intrusion detection in internet of things: Review and challenges. *Comput. Netw.* **2022**, *203*, 108661. [CrossRef]
23. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Federated deep learning for cyber security in the Internet of Things: Concepts, applications, and experimental analysis. *IEEE Access* **2021**, *9*, 138537–138561. [CrossRef]
24. Bhagoji, A.N.; Chakraborty, S.; Mittal, P.; Calo, S. Analyzing federated learning through an adversarial lens. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 634–643.
25. Rey, C.; Sánchez, P.M.S.; Celdrán, A.H.; Bovet, G.; Jaggi, M. Federated learning for malware detection in IoT devices. *Comput. Netw.* **2022**, *209*, 108917. [CrossRef]
26. Danquah, L.K.G.; Appiah, S.Y.; Mantey, V.A.; Danlard, I.; Akowuah, E.K. Computationally efficient deep federated learning with optimized feature selection for IoT botnet attack detection. *Intell. Syst. Appl.* **2024**, *25*, 200462. [CrossRef]
27. Savelyeva, D.D.; Tatarnikova, T.M. Hybrid System for Monitoring the Traffic Consumption of IoT Devices. In Proceedings of the 2024 Wave Electronics and Its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, Russia, 3–7 June 2024; pp. 1–4.
28. Sun, S.; Sharma, P.; Nwodo, K.; Stavrou, A.; Wang, H. FedMADE: Robust Federated Learning for Intrusion Detection in IoT Networks Using a Dynamic Aggregation Method. In *International Conference on Information Security*; Springer International Publishing: Cham, Switzerland, 2024; pp. 286–306.
29. Alsaleh, S.; Menai, M.E.B.; Al-Ahmadi, S. A Heterogeneity-Aware Semi-Decentralized Model for a Lightweight Intrusion Detection System for IoT Networks Based on Federated Learning and BiLSTM. *Sensors* **2025**, *25*, 1039. [CrossRef] [PubMed]
30. Devine, M.; Ardakani, S.P.; Al-Khafajiy, M.; James, Y. Federated machine learning to enable intrusion detection systems in IoT networks. *Electronics* **2025**, *14*, 1176. [CrossRef]
31. Javeed, D.; Saeed, M.S.; Adil, M.; Kumar, P.; Jolfaei, A. A federated learning-based zero trust intrusion detection system for Internet of Things. *Ad Hoc Netw.* **2024**, *162*, 103540. [CrossRef]
32. Belenguer, A.; Navaridas, J.; Pascual, J.A. A review of federated learning in intrusion detection systems for IoT. *arXiv* **2022**, arXiv:2204.12443.
33. Rosay, A.; Cheval, E.; Carlier, F.; Leroux, P. Network intrusion detection: A comprehensive analysis of CIC-IDS2017. In Proceedings of the 8th International Conference on Information Systems Security and Privacy, Online, 9–11 February 2022; pp. 25–36.
34. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]

35. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-baiot—Network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Comput.* **2018**, *17*, 12–22. [CrossRef]
36. Parmisano, A.; Garcia, S.; Erquiaga, M.J. *A Labeled Dataset with Malicious and Benign IoT Network Traffic*; Stratosphere Laboratory: Praha, Czech Republic, 2020.
37. Guerra-Manzanares, A.; Medina-Galindo, J.; Bahsi, H.; Nõmm, S. MedBIoT: Generation of an IoT botnet dataset in a medium-sized IoT network. In Proceedings of the 6th International Conference on Information Systems Security and Privacy ICISSP, Valletta, Malta, 25–27 February 2020; pp. 207–218.
38. Vaccari, I.; Chiola, G.; Aiello, M.; Mongelli, M.; Cambiaso, E. MQTTset, a new dataset for machine learning techniques on MQTT. *Sensors* **2020**, *20*, 6578. [CrossRef]
39. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]
40. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A real-time dataset and 787 benchmark for large-scale attacks in IoT environment. *Sensors* **2023**, *23*, 5941. [CrossRef] [PubMed]
41. Biswas, K.; Reza, A.; Karri, M.; Jha, D.; Pan, H.; Tomar, N.; Bagci, U. Optimizing Neural Network Effectiveness via Non-monotonicity Refinement. In Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Tucson, AZ, USA, 28 February–4 March 2025; pp. 4300–4309.
42. Powers David, M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.