LETTERS IN BIOMATHEMATICS
An International Journal

RESEARCH ARTICLE

OPEN ACCESS

Distributed Network Traffic Categorizing using Federated Learning in Industrial Internet of Things

Basmah Hayder Mahmod[1], Jamshid Bagherzadeh Mohasefi[1]

[1]Department of Computer Engineering, Urmia University, Urmia, Iran

ABSTRACT

Industrial Internet of Things (IIoT) is a major technological breakthrough that makes it possible for new kinds of automation, optimization, and cooperation between physical and virtual entities. However, complex interplay between digital and physical industrial components of these systems has increased their susceptibility to advanced cyberattacks. Although Intrusion Detection Systems (IDSs) can reliably identify attacks from normal behavior, it is necessary to have a system to categorize network traffic into various classes due to the increasing intricacy and subtlety of these traffics. The idea of federated learning is to solve data privacy issues by leveraging collaborative learning across a wide range of devices. In this paper, we have implemented homogenous and heterogenous FL-based methods to analyze and compare Edge-IIoT dataset to categorize network traffic into different categories. First, we implemented centralized deep learning methods on the specified dataset. Then, we applied homogeneous FL based methods and compared it with the centralized methods using various assessment methods. Then, we considered heterogenous FL models using combined data heterogeneity and structural heterogeneity. The experiments were conducted using 5 categories and 9 categories of traffic in the context of the IIoT dataset. The results show competitive results for both homogenous and heterogenous methods.

ARTICLE HISTORY
Received July 3, 2026
Accepted September 4, 2026

KEYWORDS
Federated Learning, Industrial Internet of Things, Network Traffic Classification, Distributed Machine Learning, Cybersecurity

1. Introduction

Deep Learning (DL) methods have demonstrated significant promise in classification capabilities, including the possibility of automatic threat detection. New developments in deep learning present viable options for enhancing IoT's cyber security. The flexibility and large-scale data handling that these approaches enable can be used to identify abnormalities in the system, forecast future trends, and learn from data. For attack detection in IIoT systems, a variety of techniques have been investigated. Recent improvements in deep learning have contributed to a promising future for improving the security of these kinds of systems. These methods have the potential to completely transform attack detection because of their ability to recognize patterns of attacks. The most widely used deep learning methods are DNNs, CNNs, Autoencoders, Dynamic Bayesian networks, and Deep Belief Networks (DBNs). However, the use of deep learning to intrusion detection in IoT systems is still in its infancy and comes with special challenges.

In practice, datasets with limited samples of cyberattacks are available to users. On the other hand, because of security considerations and privacy policies, IIoT users tend not to share their private samples of multiple attacks. Furthermore, the unpredictability and rapid evolution of unknown cyber-attacks increase the difficulty of acquiring samples and retraining the models. Federated learning (FL) is a secure method for industrial IoTs that only transmit model parameters by encrypting the transmitted data. The idea of federated learning is proposed by Google to solve data privacy issues by leveraging collaborative learning across a wide range of devices (i.e., IoT/edge/end-node devices).

FL methods are either homogenous or heterogenous. Homogeneous federated learning methods heavily rely on the assumption that all the participants share the same network structure and possess similar data distributions with similar features. In heterogeneous federated learning (HFL) there is heterogeneity in the system in some way. Heterogeneity can exist in data distribution, model structure, network environment, and devices. It poses challenges in realizing collaboration. The heterogeneous federated learning approaches can be classified into data heterogeneity, device heterogeneity, communication heterogeneity, and model heterogeneity (Lin et al., 2023). In this research we will focus mainly on data heterogeneity and structure heterogeneity.

CONTACT Basmah Hayder Mahmod: basma.haider1993@gmail.com

Heterogeneous data distribution refers to the situation where the data collected by different distributed devices does not follow the same distribution. The data collected in different participants may be Non-Independent Identically Distributed (Non-IID) or unbalanced, resulting in inconsistent update optimization directions of participants. Original methods would fail in biased collaboration. Therefore, many recent approaches attempt to tackle this challenge from various aspects.

Non-IID data occurs due to skewed data collection by different clients. This can be in the form of label skewness, feature skewness, quality skewness, or quantity skewness. In label skewness, the label distributions across participating clients are different. This phenomenon is commonly encountered in practical applications in which the data collection or annotation is inconsistent (Zhang et al., 2022). Feature skewness refers to the scenario that the feature distributions across participant clients are different. This phenomenon often occurs in complex real-world environments, implying that the feature distribution of local data on individual clients may be significantly different.

There is no work in the literature which analyzes the heterogeneous data distribution in the IIoT systems. In this paper, we will design and implement various FL-based methods to analyze and compare IID data and Non-IID data over Edge-IIoT dataset (Ferrag et al., 2022). First, we implement centralized deep learning methods on the specified dataset. Then, we consider applying FL based methods and compare it with the centralized methods from accuracy perspective. Then, we consider heterogenous FL models, either label skewness (horizontal non-IID) or structural heterogeneity and compare the various heterogenous models with homogenous and centralized methods. The experiments would be done in the context of the Edge-IIoT dataset.

As a result, there are two problems investigated in this paper. Firstly, to investigate and analyze the research that has examined federated learning techniques for intrusion detection in the context of IIoT. Secondly, to improve the application and validity of these models by efficiently using a large, real-world dataset—in this case, the EdgeIIoTset dataset—for training and assessment. In the end, useful implementation of DL and FL-DL for intrusion detection in IIoTs, will improve the security of IIoT systems enabling them to be used reliably.

Therefore, the main objective of this paper is to improve the detection of IIoT attacks by categorizing the capabilities of FL based DL techniques both homogenous and heterogenous. The following are contributions of this research:

- FL based methods have been applied and performed very well as popular DL approaches like DNN, and CNN, when it comes to categorizing various cybersecurity threats, simultaneously keeping data privacy.
- We applied FL based algorithms which performed as well as or better than classical deep learning in terms of data privacy while keeping accuracy and F1-score comparable to other methods, categorizing intrusions in the Edge-IIoT dataset.
- We show that the efficacy of heterogenous FL models trained and evaluated on the Edge-IIoTset dataset is as strong as homogenous methods.

This paper is organized into following sections. Section 2 covers literature review and the various IDS methods applied on previous works. Then section 3 proposes the materials and methods. Including dataset analysis and proposed models. Section 4 illustrates the results, and section 5 discusses the outcomes and concludes the paper.

2. Literature Review

The goal of this section is to thoroughly examine the corpus of existing research on the selection and use of deep learning (DL) and Federated Learning (FL) approaches, with a focus on federated learning tactics designed to strengthen Intrusion Detection Systems (IDS) in Industrial Internet of Things (IIoT) cyber resilience. The Industrial Internet of Things (IIoT) integrates sensors, actuators, communications, edge/cloud computing, and analytics into industrial systems to enable real-time monitoring, automation, and data-driven decision making.

Industrial Internet of Things are enablers to advance the transportation, health system, industrial control, and other industries because of the growing interconnection of computational and physical processes. These systems' importance and sophistication make them a desirable target for cyberattacks, which can have negative consequences that go beyond the digital world and affect the actual world as well (Umer et al., 2022).

This emphasizes the need for intrusion detection technologies to protect IIoTs. This serves as the main driving force for this thesis. Traditional intrusion detection techniques, which are frequently based on signatures and rules, are successful against known threats but ineffective against emerging and novel attacks, especially Zero-Day attacks (Oluwadare & ElSayed, 2023). This is especially important in the case of IIoT, where a detection system that can anticipate and protect against new threats is necessary. The necessity to overcome this obstacle and enhance IIoT's cyber security is what drives our research. IIoT is a central pillar of Industry 4.0 and the digitization of industrial operations: it couples cyber systems and physical processes to enable predictive analytics, autonomous control loops, and visibility across value chains.

Due to their versatility and capacity to handle and learn from enormous volumes of data, deep learning approaches have been increasingly popular and useful in the field of intrusion detection in recent years (Buczak & Guven, 2015;

Goodfellow et al., 2016). In IIoT context, deep learning techniques represent a potential means of improving the accuracy and efficiency of intrusion detection systems (Khraisat et al., 2019). However, there hasn't been much research done on their actual use in IIoT IDS. The need to understand more about these learning approaches and their effectiveness in the practical setting of IIoT is what drives this paper. The utilization of extensive, real-world datasets is crucial for training and assessing DL models in order to guarantee a realistic and successful investigation.

Heterogeneous devices and proprietary protocols hinder integration. Standards like OPC UA and efforts combining OPC UA + TSN are highlighted as promising routes to deterministic, interoperable industrial communication, but adoption and resource constraints on endpoint devices are practical obstacles (Ladegourdie & Kua, 2022).

Industrial control requires bounded latency and determinism for safety-critical loops. TSN and 5G URLLC are being studied to satisfy these constraints, but end-to-end performance across mixed networks and clouds requires careful architecture and validation (Zhang, Wang, et al., 2024).

Large sensor fleets produce massive data volumes; systems must support distributed processing, model retraining, and explainability for operational acceptance. Research into efficient edge-cloud partitioning, federated learning, and lifecycle management of ML models is active (Zhang, Xue, et al., 2024).

Technical solutions succeed only when organizations adapt processes and skills. Many deployments fail to realize expected value because of poor change management, lack of integration with maintenance/planning functions, or unrealistic expectations about ML performance. Consulting reviews stress aligning IIoT initiatives with business KPIs and phased pilots (McKinsey, 2021).

Early adopters of DL for IIoT IDS commonly employed sequence-aware models to exploit the temporal nature of network traffic and process telemetry. LSTM-based autoencoders and sequence models are frequently used to learn normal behavior and flag anomalous deviations, capitalizing on their ability to model time dependencies inherent to industrial control loops. Studies show that LSTM and bi-LSTM models produce strong recall on time-series attacks, such as replay and DoS variations, especially when trained on datasets that capture realistic IIoT device behavior (Waqas et al., 2025). Nevertheless, sequence models can be data-hungry and sensitive to concept drift when production processes or device firmware change, requiring retraining or continual learning strategies to remain effective (Li et al., 2024; Waqas et al., 2025).

Convolutional neural networks (CNNs) and hybrid CNN–RNN pipelines have been explored to extract local temporal patterns and hierarchical features from transformed network-flow matrices or spectrogram-like representations of traffic. Hybrid CNN–LSTM and CNN–GRU architectures frequently report competitive accuracy and lower false positive rates compared with classical ML baselines on contemporary IIoT datasets (Edge-IIoTset, TON-IoT, X-IIoTID), particularly when models are carefully tuned and combined with feature selection or dimensionality reduction pre-processing. However, researchers warn that cross-dataset generalization remains an open problem: architectures optimized on a single dataset often lose performance when transferred to different device mixes or network topologies (Ferrag et al., 2022; Salehiyan et al., 2025).

Autoencoders—both shallow and deep—have been a dominant choice for unsupervised anomaly detection in IIoT because they can be trained on largely unlabeled normal traffic and then used to (Zhong et al., 2024) detect deviations without extensive attack labeling. Variational autoencoders (VAEs), denoising autoencoders, and stacked architectures have demonstrated good detection of novel attacks (e.g., new botnet strains) on device-specific traffic traces (N-BaIoT, TON-IoT). Yet autoencoder-based detectors must deal with thresholds, reconstruction bias, and the possibility of learning to reconstruct some attack patterns if those are present in the training data; therefore, careful dataset curation and thresholding strategies are required (N-BaIoT studies; Edge-IIoT evaluations).

A notable recent trend is the adoption of graph neural networks (GNNs) and topology-aware methods to capture relational and lateral movement patterns among IIoT components. Because IIoT environments have structured connectivity (devices, gateways, PLCs, HMIs) and attack progression often involves multi-host correlation, GNNs have been shown to detect stealthy lateral propagation and multi-stage intrusions more robustly than purely flow-based models in several 2023–2025 studies. Early GNN-IDS prototypes and evaluations indicate superior ability to flag coordinated attacks and to incorporate process-level relationships, although these approaches are computationally heavier and require careful graph construction in dynamic networks (GNN-IDS literature; (Zhong et al., 2024)).

Mainstream FL approaches assume that data are sampled IID across participants, and all the users share the same network architecture. This kind of FL is called homogeneous FL. However, there are situations where data gathered by different participants is not uniform or non-IID (does not follow the same distribution), or the devices and network architecture of the participants are different. This general idea of FL is called heterogeneous FL which might have many forms of heterogeneity (Lin et al., 2023).

Li et al. (2021) produce a method called FedH2L which works contrary to existing processes that share parameters

or gradients. FedH2L relies on mutual distillation, exchanging only posteriors on a shared seed set between participants in a decentralized manner. This makes it extremely bandwidth efficient, model agnostic, and crucially produces models capable of performing well on the whole data distribution when learning from heterogeneous silos.

Zhang et al. (2022) investigate the label distribution skew in FL, where the distribution of labels varies across clients. They demonstrate both theoretically and empirically that previous methods based on softmax cross-entropy are not suitable, which can result in local models heavily overfitting to minority classes and missing classes. They propose FedLC (Federated learning via Logits Calibration), which calibrates the logits before softmax cross-entropy according to the probability of occurrence of each class.

3. Materials and Methods

We will use a cross-validation technique to guarantee the validity and reliability of our findings. The training data must be divided into m folds. The model must then be trained on m-1 of those folds and validated on the remaining folds. The fact that this process is repeated m times ensures that every section is used for validation exactly once. A more robust indicator of the model's efficacy is provided by averaging its performance throughout the m iterations.

3.1 Dataset

The dataset Edge-IIoTset was introduced in Ferrag et al. (2022) specifically for IoT and IIoT applications, with the objective of supporting cyber-security research and it is relevant for applying ML and DL models for IDS purposes. This dataset is a comprehensive source embodying many features that are indicative of a wide range of IIoT and IoT applications. It contains tuples from both typical network traffic and a variety of simulated cyberattacks, which is exactly what this study is trying to identify and categorize. The dataset's embedded cyberattacks provide a reliable environment for training and testing the ML and DL models by imitating the patterns and behaviors of different threat kinds. These threats, which range from cross-site scripting to DDoS attacks and scanning, encompass a broad range of CPS's cybersecurity issues. To serve as the target variable for the intrusion detection models, each record is also annotated to indicate if it reflects typical traffic or a particular kind of attack.

The initial step in our data preprocessing procedure involves eliminating the part of data which is not of broad applicability. In this research we are interested in creating and training models to identify threats and attacks using network traffic. The models trained on a specific dataset should have the property of generalizability to unseen and unfamiliar settings, i.e., it should be applied accurately to environments with different data. To ensure generalization of the built models, we should ignore the features of data which are exclusively available in the training environment, and they are not available in any form in other (test) networks.

The dataset includes one normal and 14 attack classes. Every attack is labeled as a number between 0 to 14. The normal class is labeled with the value of 7. After initial analysis of the data, a sort of preprocessing operation is done on the dataset to ensure the data cleaning and appropriateness. This process involves encoding categorical variables, addressing missing values, data normalization, and noisy data removal. Attributes, such as ip.src_host and ip.dst_host, will be removed because they are inherent to the testbed's network architecture. Also, features like frame. time and icmp.transmit. timestamp which represents time-related values may not be much more accurate for other network traffic attacks, so they are removed too. In order to standardize features, z-score normalization is applied on all numeric variables. In this paper, class imbalance problem is taken into consideration using SMOTE, and Random under sampling.

Since our dataset's target variable is also categorical, it's critical that we convert the target variable into a numerical format so that our machine learning models can process it efficiently. We accomplish this by using label encoding, which offers a practical means of numerically representing class labels. Each different class label is given a unique integer as part of the label encoding process. For dependent variables in classification tasks, this kind of encoding is especially useful since it provides a streamlined, numerical representation of different classes that can be quickly understood and adjusted by a variety of machine learning algorithms.

3.2 Model Construction

The baseline models implemented in this paper include Convolutional Neural Networks (CNN) and Deep Neural Network (DNN). Then we apply homogenous DNN and heterogenous DNN and CNN federated learning models in industrial internet of things, particularly their capacity to categorize various cybersecurity attacks. These models constitute DL and FL based methods well-known for their highly private and efficient methods in environments with high-dimensional and imbalanced data.

In model construction and training, we must define the architecture of the deep learning models. It includes describing the number of layers, the type of each layer (flat, convolution, pooling, etc.), the number of nodes in each layer and the

type of each node, and other related parameters such as loss function, learning rate and so on. For federated learning setup, we must define the number of distributed nodes, the number of epochs, and the type of heterogeneity.

Two deep learning-based models including CNN, and DNN are developed for attack prediction of network traffic. To compare with federated learning situation, homogeneous and heterogenous methods including FL_Hom, and FL_Het are implemented. In this paper we considered two different scenarios to study network attacks and intrusions. First Scenario considers 5 attack categories with one normal case. In the second scenario we consider 8 attacks and one normal traffic. FL-based methods are considered in two modes with homogeneous and with heterogonous data. The results are compared to show the differences and advantages of both.

The dataset contains 157,800 samples along with attack types and some other features about the network traffic. In the first scenario, we categorize the samples into 5 classes. Class 1 representing DDoS attacks includes DDoS_HTTP (1), DDoS_ICMP (2), DDoS_TCP (3), DDoS_UDP (4). Second class represents web attacks including SQL_injection (11), Uploading (12), Vulnerability_scanner (13), XSS (14). The third category includes spyware attacks containing Backdoor (0), Password (8), Port_Scanning (9), Ransomware (10). In the fourth category we consider Fingerprinting (5), and Man in the Middle (MITM: 6) attacks. The fifth category includes Normal (7) samples.

In the second scenario, the scores were divided into 9 classes. The cases in which attack_type == 0, attack_type == 2, attack_type == 3, attack_type == 4, attack_type == 6, or attack_type == 10 which represent Backdoor, DDoS_ICMP, DDoS_TCP, DDoS_UDP, MITM, and Ransomware attacks respectively represent six classes of attacks. The attack_type == 7 which represents Normal vector constitutes 7th class. The cases in which attack_type == 1, attack_type == 8, attack_type == 11, attack_type == 12, attack_type == 13, and attack_type == 14 which represent DDoS_HTTP, Password, SQL_Injection, Uploading, Vulnerability_scanner, and XSS attacks are considered altogether as a single class. The cases in which attack_type == 5, and attack_type == 9 which represent Fingerprinting and Port_scanning are grouped altogether. These constitute 9 classes of the third scenario. Table 1 summarizes the categories of two scenarios.

Table 1: Distribution of Samples in Each Scenario

Scenarios	Class	Number of Samples
First Scenario: Five Classes	DDoS	49396
	Web based	40708
	Spyware	41180
	Man in the Middle	2215
	Normal	24301
Second Scenario: Nine Classes	Backdoor	10195
	DDoS_ICMP	14090
	DDoS_TCP	10247
	DDoS_UDP	14498
	MITM	1214
	Ransomware	10925
	Normal	24301
	DDoS_HTTP, Password, SQL_Injection, Uploading, Vulnerability_scanner, XSS	10561+ 9989+ 10311+ 10269+ 10076+ 10052
	Fingerprinting and Port_scanning	1001+ 10071

4. Implementation and Results

4.1 The First Scenario: Five Classes

In the first scenario we divide the samples into five classes, various attack vectors are considered as 4 attack classes and one normal class. The performance results of the Random Forest (RF) classifier for the five-class classification task are summarized in Table 2.

Table 2(a): Classification Report of Random Forest for 5 Classes

Class	Accuracy	Precision	Recall	F1-Score	Support
1	0.94	0.95	0.95	0.95	12297
5		0.93	0.78	0.84	535
7		1.00	1.00	1.00	6052

Table 2(b): Classification Report of Random Forest for 5 Classes

Class	Accuracy	Precision	Recall	F1-Score	Support
8	0.94	0.95	0.94	0.94	10307
11		0.90	0.92	0.91	10259
Macro Avg		0.95	0.92	0.93	39450
Weighted Avg		0.94	0.94	0.94	39450

Table 3 provides an overview of the DNN model's results for the five-class detection problem.

Table 3: Classification Report of DNN for 5 Classes

Class	Accuracy	Precision	Recall	F1-Score	Support
0	0.84	0.73	1.00	0.84	14175
1		1.00	1.00	1.00	4810
2		1.00	1.00	1.00	4784
3		0.77	0.81	0.79	18654
4		0.94	0.75	0.83	30480
Macro avg		0.89	0.91	0.89	72903
Weighted Avg		0.86	0.84	0.84	72903

Table 4 summarizes the CNN model's detection performance across five categories.

Table 4: Classification Report of CCNN for 5 Classes

Class	Accuracy	Precision	Recall	F1-Score	Support
0		0.73	1.00	0.84	14153
1		0.89	0.89	0.89	9712
2		1.00	1.00	1.00	4868
3	0.84	0.74	0.81	0.77	17585
4		0.99	0.73	0.84	26585
Macro Avg		0.87	0.88	0.87	72903
Weighted Avg		0.86	0.84	0.84	72903

To simulate an IID dataset using Edge-IIoT dataset, first we balance the dataset. Then dataset is split into train and test set with 80% of data for training and 20% for testing. The train set includes 291612 samples which are divided into 5 shares (5 separate datasets) using a stratified partitioning method. The way we will have 5 separate datasets with 29161 samples in each share. These shards are stored in local nodes for federated learning algorithm. Each local node runs a DNN on this local dataset and the parameters of the trained model is sent to a central node. The central node gets parameters from 5 local nodes and aggregates the results using various aggregation methods. The easiest way is to use average aggregation which calculates mean value of each parameter. After k rounds of applying this cycle the final trained model in the central node, which is called global model, is tested on test dataset with 72903 samples. The target feature includes five classes in this section. The structure of the network used in local and central nodes is represented in Table 5.

Table 5: Structure Of DNN Used for Federated Learning in 5 Classes Scenario

Layer (Type)	Output Shape	Param #
Dense (Dense)	(None, 32)	1344
Dense_1 (Dense)	(None, 24)	792
Dense_2 (Dense)	(None, 20)	500
Dense_3 (Dense)	(None, 16)	336
Dense_4 (Dense)	(None, 5)	85

The accuracy of applying the method on the test set is represented in Table 6 and the confusion matrix is represented in the Figure 1.

Table 6: Homogenous FL DNN in Five Classes Scenario

Method	Accuracy	Loss	No. Rounds	No Clients
Federated Learning DNN	0.87217	0.11484	10	10

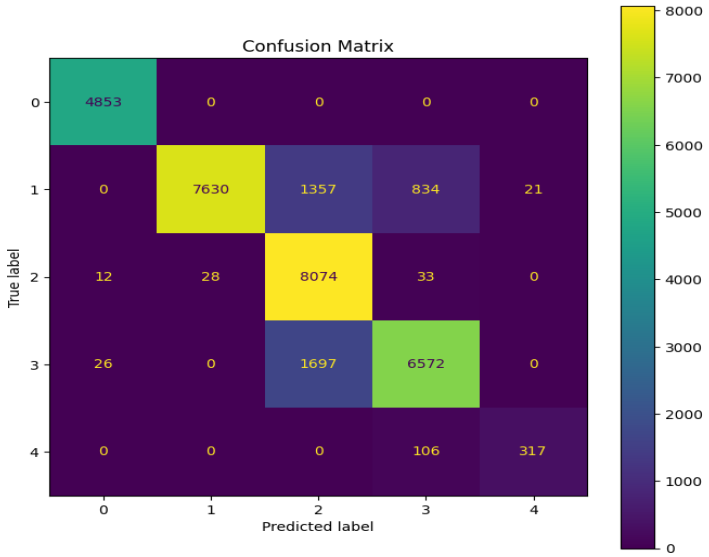


Figure 1: Confusion Matrix for 5 Classes FL-DNN

Following represents 10 rounds of federated learning performance:
comm_round: 0 | global_acc: 0.8033586818757922 | global_loss: 0.24236424267292023
comm_round: 1 | global_acc: 0.8629594423320659 | global_loss: 0.12223909050226212
comm_round: 2 | global_acc: 0.8677122940430925 | global_loss: 0.11860012263059616
comm_round: 3 | global_acc: 0.8701204055766794 | global_loss: 0.12001931667327881
comm_round: 4 | global_acc: 0.8709125475285171 | global_loss: 0.12141401320695877
comm_round: 5 | global_acc: 0.8706907477820025 | global_loss: 0.11826775223016739
comm_round: 6 | global_acc: 0.8708808618504436 | global_loss: 0.1188369020819664
comm_round: 7 | global_acc: 0.8696134347275032 | global_loss: 0.11694640666246414
comm_round: 8 | global_acc: 0.8710709759188847 | global_loss: 0.11709468811750412
comm_round: 9 | global_acc: 0.8651140684410646 | global_loss: 0.11894769221544266

In this section also we apply two types of heterogeneity, structure heterogeneity and data heterogeneity. For data heterenity we use a Dirichlet distribution to distribute data to different nodes. The data distribution proportion, accuracy, and loss of the first round of execution is as follows:

Samples: 6929 - accuracy: 0.8056 - loss: 0.1749
Samples: 60 - accuracy: 0.3566 - loss: 0.6086
Samples: 74 - accuracy: 0.3312 - loss: 0.6164
Samples: 11749 - accuracy: 0.8169 - loss: 0.1584
Samples: 915 - accuracy: 0.7014 - loss: 0.3116

For structure hetereginity we define two different types of structures for the models used in various local nodes. For local nodes with odd numbers, we use a Deep Neural Network (DNN) structure of Table 7. however, for the local nodes with even numbers, we use a Convolution Neural Network (CNN) with the structure represented in the Table 8.

Table 7: Structure of DNN Network used in Odd Clients for Five Classes and Five Clients

Layer (Type)	Output Shape	Param #
Flatten (Flatten)	(None, 41)	0
Dense_1 (Dense)	(None, 128)	5376
Dense_2 (Dense)	(None, 10)	1290
Dense_3 (Dense)	(None, 5)	55

The structure of global network is like the Table 7. The accuracy of the method using 5 clients and at the first 3 rounds and 10th round are represented in Table 9; the confusion matrix is represented in Figure 2.

Table 1 (a): Structure of CNN Network used in Even Clients for Five Classes and Five Clients

Layer (Type)	Output Shape	Param #
conv1d (Conv1D)	(None, 36, 32)	224
max_pooling1d (MaxPooling1D)	(None, 18, 32)	0

Table 2 (b): Structure of CNN Network used in Even Clients for Five Classes and Five Clients

Layer (Type)	Output Shape	Param #
conv1d (Conv1D)	(None, 16, 64)	6208
max_pooling1d (MaxPooling1D)	(None, 8, 64)	0
flatten_1 (Flatten)	(None, 512)	0
dense_1 (Dense)	(None, 10)	5130
dense_2 (Dense)	(None, 5)	55

Table 3: Accuracy of Local Clients at Different Rounds of Training Process

Round	Accuracy (Round1)	Accuracy (Round2)	Accuracy (Round3)	Round 4…9	Round 10
Local Node 1	0.8056	0.8582	0.8630	....	0.8671
Local Node 2	0.3566	0.5554	0.7663	…	0.8188
Local Node 3	0.3312	0.8536	0.8906	…	0.8871
Local Node 4	0.8169	0.8635	0.8637	…	0.8688
Local Node 5	0.7014	0.8515	0.8645	…	0.8723

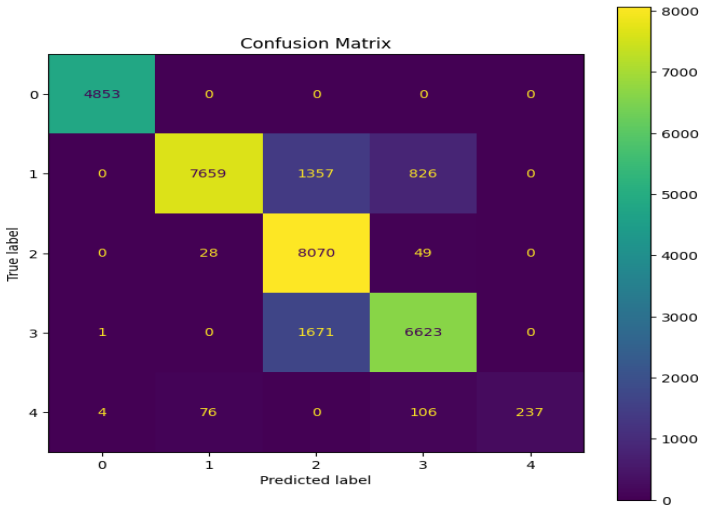


Figure 1: Confusion Matrix for Heterogenous FL (DNN+CNN) for Five Classes

4.2 Comparison of Models

After applying all the models on the 5 classes dataset, here we summarize and compare the results of the methods. For sake of conciseness, we just write the average of all classes in Table 10 but not every single class separately.

Table 4: Comparison of Methods for First Scenario

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
RF	0.94	0.94	0.94	0.94
Deep Neural Networks	0.84	0.86	0.84	0.84
Convolutional Neural Networks	0.84	0.86	0.84	0.84
Homogenous Federated Learning	0.87	0.89	0.87	0.87
Heterogenous Federated Learning	0.87	0.89	0.87	0.87

4.3 The Second Scenario: Nine Classes

The predictive performance of the RF algorithm in the nine-class scenario is presented in Table 11.

Table 5 (a): Classification Report of Random Forest for 9 Classes

Class	Accuracy	Precision	Recall	F1-Score	Support
0	0.96	0.97	0.92	0.94	2544
1		0.95	0.97	0.96	12843
2		1.00	1.00	1.00	3581
3		1.00	1.00	1.00	2551
4		1.00	1.00	1.00	3581
5		0.88	0.86	0.87	5299
6		1.00	1.00	1.00	271
7		1.00	1.00	1.00	6052

Table 6 (b): Classification Report of Random Forest for 9 Classes

Class	Accuracy	Precision	Recall	F1-Score	Support
8		0.90	0.90	0.90	2728
Macro Avg		0.97	0.96	0.96	39450
Weighted Avg		0.96	0.96	0.96	39450

The assessment of the DNN method in the nine-class configuration is reported in Table 12.

Table 7: Classification Report of DNN for 9 Classes and 9 Clients

Class	Accuracy	Precision	Recall	F1-Score	Support
0	0.87	0.80	0.99	0.89	3895
1		0.99	0.85	0.92	28219
2		0.99	1.00	0.99	4853
3		1.00	0.62	0.77	7901
4		0.99	1.00	1.00	4803
5		0.53	0.85	0.65	9137
6		1.00	1.00	1.00	4813
7		1.00	1.00	1.00	4784
8		0.80	0.87	0.83	4498
Macro Avg		0.90	0.91	0.89	72903
Weighted Avg		0.91	0.87	0.88	72903

Analysis of the CNN classifier's results within the nine-class framework is presented in Table 13.

Table 8: Classification Report of CNN for 9 Classes

Class	Accuracy	Precision	Recall	F1-Score	Support
0	0.89	0.89	1.00	0.94	4430
1		1.00	0.85	0.92	28477
2		0.99	1.00	1.00	4773
3		0.86	0.67	0.75	6224
4		1.00	1.00	1.00	4816
5		0.58	0.81	0.68	10437
6		1.00	1.00	1.00	4893
7		1.00	1.00	1.00	4868
8		0.81	0.98	0.88	3985
Macro Avg		0.90	0.92	0.91	72903
Weighted Avg		0.91	0.89	0.89	72903

Next, we implement FL based DNN for the second scenario. The structure of the networks used in homogenous federated learning for local nodes and the central node is represented in Table 14.

Table 9: Structure of the DNN used in 9 Classes Scenario for Homogenous FL

Layer (Type)	Output Shape	Param #
Dense (Dense)	(None, 32)	1344
Dense_1 (Dense)	(None, 24)	792
Dense_2 (Dense)	(None, 20)	500
Dense_3 (Dense)	(None, 16)	336
Dense_4 (Dense)	(None, 9)	153

The performance of homogenous FL is represented in Table 15, and the confusion matrix is shown in the Figure 3, respectively.

Table 10: Performance of the FL DNN in 9 Classes Scenario

Method	Accuracy	Loss	No. Rounds	No Clients
Federated Learning DNN	0.897782	0.0536113	15	9

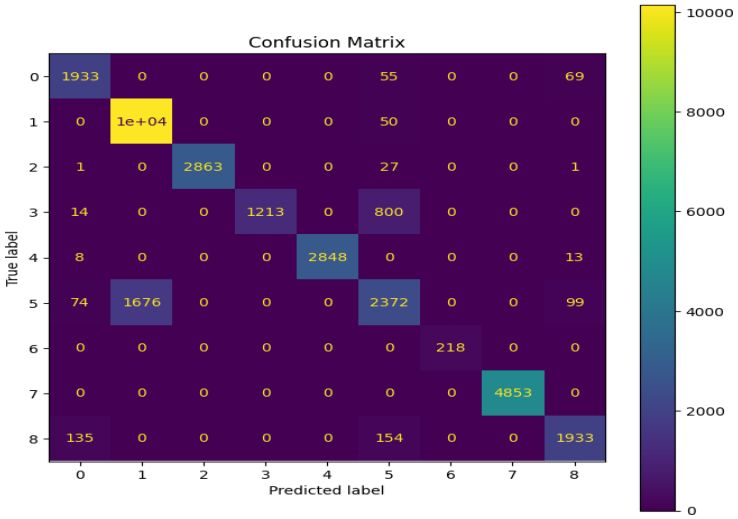


Figure 2: Confusion Matrix of Homogenous FL DNN in 9 Classes Scenario

The class wise precision, recall and f1-score is represented in the Table 16.

Table 11: Class-Wise Performance for FL DNN In 9 Classes

Class	Precision	Recall	F1-Score	Support
0	0.89	0.94	0.92	2057
1	0.86	1.00	0.92	10201
2	1.00	0.99	0.99	2892
3	1.00	0.60	0.75	2027
4	1.00	0.99	1.00	2869
5	0.69	0.56	0.62	4221
6	1.00	1.00	1.00	218
7	1.00	1.00	1.00	4853
8	0.91	0.87	0.89	2222

The heterogenous federated learning for nine classes scenario is implemented in two ways of heterogeneity: data and structure. As described in the previous sections data heterogeneity is done using a Dirichlet Distribution with alpha=0.3, and structural heterogeneity is implemented using two types of networks including DNN and CNN for odd and even nodes. The structure of the networks is like those used in the 5 classes scenario with output layer changed to 9 units. The results of the execution in 10 rounds of 9 clients are represented in the Table 17. The final accuracy is represented in the Table 18 and the confusion matrix is shown in the Figure 4.

Table 12: Accuracy on Rounds of Training of Heterogenous FL of 9 Classes

Round	Accuracy (Round1)	Accuracy (Round2)	Accuracy (Round3)	Round 4…9	Round 10
Local Node 1	0.8068	0.8703	0.8929	....	0.8992
Local Node 2	0.1208	0.5464	0.7565	…	0.8558
Local Node 3	0.6316	0.8320	0.8737	…	0.8885
Local Node 4	0.7030	0.8264	0.8564	…	0.8977
Local Node 5	0.5001	0.8601	0.8992	…	0.9101
Local Node 6	0.4774	0.8073	0.8175	…	0.8893
Local Node 7	0.6500	0.8346	0.8852	…	0.9086
Local Node 8	0.7931	0.8980	0.8976	…	0.9007
Local Node 9	0.7780	0.8604	0.8899	…	0.9098

Table 13 (a): Precision, Recall, F-Score, and Accuracy of Het-FL of Nine Classes

Class	Precision	Recall	F1-Score	Support
0	0.91	0.91	0.91	2057
1	0.86	0.99	0.92	10201
2	1.00	0.99	0.99	2892
3	0.94	0.63	0.75	2027
4	1.00	0.99	1.00	2869
5	0.70	0.53	0.60	4221
6	1.00	1.00	1.00	218

Table 14 (b): Precision, Recall, F-Score, and Accuracy of Het-FL of Nine Classes

Class	Precision	Recall	F1-Score	Support
7	1.00	1.00	1.00	4853
8	0.87	0.93	0.90	2222
Macro Avg.	0.92	0.89	0.90	31560
Weighted Avg	0.90	0.90	0.89	31560

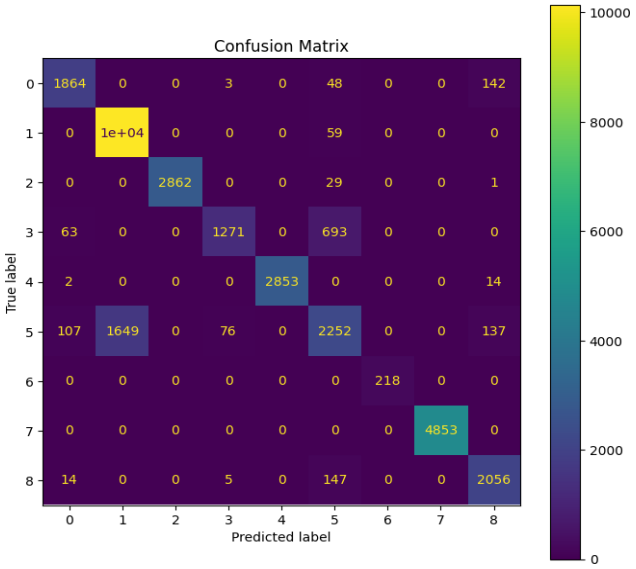


Figure 3: Confusion Matrix for Heterogenous FL with Nine Classes

4.4 Comparison of all Methods for Nine Classes

The methods implemented for nine classes scenario are compared in a unified way in the Table 19.

Table 15: Comparison of the Models for 9 Classes Scenario

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
RF	0.96	0.96	0.96	0.96
Deep Neural Networks	0.87	0.91	0.87	0.88
Convolutional Neural Networks	0.89	0.91	0.89	0.89
Homogenous Federated Learning	0.90	0.90	0.90	0.89
Heterogenous Federated Learning	0.90	0.90	0.90	0.89

5. Conclusion

In this paper, we investigated the comparative study of applying traditional Random Forest (RF) method, deep learning methods, and federated learning (FL) methods to predict attack from normal traffic in intrusion detection systems using IIoTSet 2022 dataset. We used a dataset including 41 features representing various features of network traffic. Two scenarios were implemented for traffic classification, including five classes, and nine classes. Various methods were used for classification. One Machine Learning (ML) algorithm, two deep learning models, and two FL based methods (homogeneous and heterogenous) were developed for various scenarios. We found the best hyper-parameters for all models using GridCV search method. The results revealed that the heterogenous FL works as better as Homogenous and both work as well as centralized DNN and CNN methods. Due to the high accuracy and transparency in decision-making, the proposed models can be further improved using better feature engineering techniques.

6. References

Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, *18*(2), 1153-1176.

Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access*, *10*, 40281-

40306.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1-22.

Ladegourdie, M., & Kua, J. (2022). Performance analysis of opc ua for industrial interoperability towards industry 4.0. *IoT*, *3*(4), 507-525.

Li, Y., Zhou, W., Wang, H., Mi, H., & Hospedales, T. M. (2021). FedH2L: Federated learning with model and statistical heterogeneity. *arXiv preprint arXiv:2101.11296*.

Li, Z., Fang, W., Zhu, C., Song, G., & Zhang, W. (2024). Toward deep learning based intrusion detection system: A survey. Proceedings of the 2024 6th International Conference on Big Data Engineering,

Lin, Y., Long, Y., Zhou, Z., Pang, Y., & Yang, C. (2023). Towards Heterogeneous Federated Learning: Analysis, Solutions, and Future Directions. International Conference on Artificial Intelligence Security and Privacy,

McKinsey, C. (2021). The Internet of Things: Catching up to an accelerating opportunity. *Internet of Things*.

Oluwadare, S., & ElSayed, Z. (2023). A Survey of Unsupervised Learning Algorithms for Zero-Day Attacks in Intrusion Detection Systems. The International FLAIRS Conference Proceedings,

Salehiyan, A., Moghaddam, P. S., & Kaveh, M. (2025). An optimized Transformer–GAN–AE for intrusion detection in edge and IIoT systems: experimental insights from WUSTL-IIoT-2021, edgeIIoTset, and TON_IoT datasets. *Future Internet*, *17*(7), 279.

Umer, M. A., Junejo, K. N., Jilani, M. T., & Mathur, A. P. (2022). Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection*, *38*, 100516.

Waqas, A., Khan, S. D., Ullah, Z., Ullah, M., & Ullah, H. (2025). Comparative Analysis of Deep Learning Models for Intrusion Detection in IoT Networks. *Computers*, *14*(7), 283.

Zhang, J., Li, Z., Li, B., Xu, J., Wu, S., Ding, S., & Wu, C. (2022). Federated learning with label distribution skew via logits calibration. International Conference on Machine Learning,

Zhang, T., Wang, G., Xue, C., Wang, J., Nixon, M. S., & Han, S. (2024). Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions. *ACM Computing Surveys*, *57*(2).

Zhang, T., Xue, C., Wang, J., Yun, Z., Lin, N., & Han, S. (2024). A Survey on Industrial Internet of Things (IIoT) Testbeds for Connectivity Research. *arXiv e-prints*, arXiv: 2404.17485.

Zhong, M., Lin, M., Zhang, C., & Xu, Z. (2024). A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges. *Computers & Security*, *141*, 103821.