Applying Federated Learning to ICMP Flood Attack Detection in Industrial Control System

Iuon-Chang Lin [1][0009-0002-3367-6317];

[2] Mao-Hsiu Hsu [2][0009-0002-3188-4374] and [1] Yung-Hsing Chen

[1] National Chung Hsing University, Taichung Taiwan
[2] Department of Electro-Optical Engineering, National Formosa University, Yunlin Taiwan

Correspondence
Mao-Hsiu Hsu, Department of Electro-Optical Engineering, National Formosa University, Yunlin Taiwan
Email: mh.hsu@nfu.edu.tw

Abstract

Industry 4.0 has revolutionized the manufacturing sector, driving the integration of Information Technology (IT) into Operational Technology (OT). This convergence has led to the widespread use of IoT and IIoT technologies in industrial control systems (ICSs) to enhance efficiency and quality. However, it also exposes ICSs to cyber threats like Distributed Denial-of-Service (DDoS) attacks, which can compromise system availability. This study aims to simulate and mitigate the impact of common DDoS attacks on ICSs using a public dataset and three physical hosts. To address transmission costs and data security concerns, we employ a Multilayer Perceptron (MLP) model with Federated Learning for training. Additionally, we investigate data augmentation techniques to improve the model's performance in detecting ICMP flood attacks, addressing challenges associated with data silos.

KEYWORDS

DDoS,ICS,IoT ,IIoT ,Deep Learning,Federated Learning

1. INTRODUCTION

1.1. Background and Motivation

The advent of IoT technology has catalyzed the proliferation of IoT devices within Industrial Control Systems (ICS), thereby ushering in the era of Industrial IoT (IIoT). IIoT empowers industrial domains to seamlessly collect real-time data, bolster control mechanisms, and leverage cloud or edge computing infrastructures in an agile and efficient manner, thereby blurring the demarcation between Operation Technology (OT) and Information Technology (IT) within ICS [1, 2]. However, the integration of IT into OT poses significant security challenges, particularly when ICS is exposed to the Internet. OT systems, traditionally confined to closed networks, become vulnerable to service disruptions stemming from unanticipated network traffic, as the remote access control risks are inadequately mitigated. Moreover, the protracted lifecycle of ICS renders it inherently less secure than IT systems, fostering numerous vulnerabilities due to the absence of regular update mechanisms. Consequently, when confronted with analogous threats as IT, vulnerabilities inherent in OT systems can precipitate substantial disruptions and impairments to critical infrastructure within industrial domains [3, 4]. Notably, system availability emerges as the principal vulnerability of ICS, with Denial of Service (DoS) attacks constituting the most prevalent threat vector [5]. DoS attacks strategically target entities with constrained computing and storage capacities, inundating system resources with an influx of Request Packets from a single source, thereby inducing temporary or permanent paralysis and impeding the delivery of normal services. Furthermore, malevolent actors may orchestrate the aggregation of infected computers into a botnet, enabling remote manipulation to execute Distributed Denial of Service (DDoS) assaults from diverse sources [6]. Among common DDoS methodologies, the ICMP flood attack stands out, characterized by the inundation of Internet Control Message Protocol (ICMP) request packets targeting the designated victim, thereby inundating the target with processing errors and network diagnostic data, consequently impeding its ability to respond to legitimate service requests [7]. Compounding the challenge, ICMP serves as a pivotal network diagnostic mechanism in routine network operations, rendering the delineation between legitimate and malicious ICMP packets increasingly arduous.

In summary, the security landscape surrounding Industrial Control Systems (ICS) within the Operation Technology (OT) domain is besieged by a plethora of diverse security threats, notably Distributed Denial-of-Service (DDoS) attacks, necessitating robust mechanisms for real-time detection and prevention of malicious incursions. In addition to conventional automated network security and management tools, an increasing body of research has turned to machine learning methodologies to fortify information security defenses. Machine learning techniques, adept at learning and discerning various data patterns and behavioral anomalies, hold promise in augmenting the protective capabilities of ICS [8]. However, traditional machine learning paradigms, which operate within a centralized computing framework, mandate the transfer of data to a central server for processing, incurring heightened communication costs and exposing vulnerabilities such as data leakage, single points of failure (SPOF), and potential misuse or abuse. To address these challenges, decentralized machine learning approaches have emerged, with Google's federated learning framework emerging as a prominent contender. Federated learning facilitates the distribution of requisite data across devices or servers for localized model training, thereby obviating the need for centralized data transmission while safeguarding data integrity [9]. This paradigm shift not only mitigates communication costs and enhances resource utilization efficiency through edge computing integration but also resolves the issue of data silos, rendering federated learning an adaptable solution to the evolving landscape of information security challenges [10, 11]. Moreover, beyond sharing model parameters among participating clients to bolster generalization capabilities, the augmentation of data can further enhance the efficacy of classification models. Data augmentation techniques diversify datasets, generating additional samples to enrich the model's learning capabilities and mitigate class imbalance, thereby averting biases towards majority categories and elevating prediction accuracy.

In the contemporary era of big data, the proliferation of data generated by mobile and IoT devices has prompted its upload to cloud servers or data centers for processing. Nonetheless, regulatory frameworks such as the EU's General Data Protection Regulation (GDPR) and the U.S. Consumer Privacy Bill of Rights Act impose constraints on the collection, storage, and utilization of such data. Moreover, the cloud-centric approach is often associated with implications for network performance. To address these challenges, mobile edge computing (MEC) has emerged as a potential solution. MEC leverages the computational and storage capabilities of end devices and edge servers to conduct model training in proximity to the data source. However, despite its potential benefits, MEC introduces privacy concerns during data transmission and processing [18]. As such, the adoption of MEC necessitates careful consideration of privacy implications alongside technological advancements.

In Chapter 2, this study will sequentially introduce the related work, including the existing research on DDoS attack detection and federated learning techniques; Chapter 3 describes the application scenarios, dataset processing, and research methodologies; Chapter 4 presents the experimental results and analyses comparing the performance of centralised deep learning, federated learning, and data augmentation coupled with federated learning; and finally, Chapter 5 concludes with a summary of the research contributions and discusses the possible directions for future research.

1.2. Purpose

In this study, we leverage the federated learning architecture, a decentralized machine learning framework, to train Multilayer Perceptron (MLP) deep learning models tailored for detecting Distributed Denial-of-Service (DDoS) attacks. Our training dataset originates from Edge-IIoTset [12], specifically designed to emulate attacks on targets including IoT devices, Industrial IoT (IIoT) devices, and edge servers. We conduct experiments using three physical hosts configured to simulate scenarios within smart manufacturing domains. Through the application of data augmentation techniques and the adoption of federated learning, we enhance the performance of detecting ICMP flood attacks-a task originally performed via centralized training on the local side of each node.

Table 1. provides a comparative analysis of various methodologies for anomaly detection and attack mitigation in networked systems

Files	Limitations	Methods	Advantages and Disadvantages
Online_Privacy-Preserving_Data-Driven_Network_Anomaly_Detection	Tradeoff between privacy and anomaly detection optimization	CUSUM algorithm, local outlierness scores	Advantages include the tradeoff between privacy and anomaly detection, along with theoretical approximations of lower bounds. Disadvantages encompass considerations such as unknown time-varying parameters and the impact of small network sizes.
iDAM_A_Distributed_MUD_Framework_for_Mitigation_of_Volumetric_Attacks_in_IoT_Networks	Inaccuracies and high computational costs in prior methods	MUD compliant IoT, OC-SVM models	iDAM efficiently detects and mitigates volumetric attacks in IoT networks, whereas MUD is limited by ACL constraints and cannot prevent such attacks effectively.
Collaborative_Learning_for_Cyberattack_Detection_in_Blockchain_Networks	Privacy risks in centralized learning	Collaborative learning model, BC-ID tool	Advantages of collaborative learning in blockchain network intrusion detection include efficient knowledge sharing among nodes for enhanced accuracy, privacy protection, and reduced network congestion with decentralized approaches. Centralized learning models, however, are hindered by their reliance on local datasets, limiting detection performance.
SDN-Based_Federated_Learning_Approach_for_Satellite-I	Lower accuracy in federated learning	SDN backbone, federated learning	SDN offers centralized control, efficient resource management, and network slicing, while federated

oT_Framework_to_Enhance_Data_Security_and_Privacy_in_Space_Communication	compared to traditional models		learning may provide lower accuracy than traditional ML methods.+
Blockchain-Empowered Decentralized Horizontal Federated Learning for 5G-Enabled UAVs https://ieeexplore.ieee.org/document/9551794	Model is not compressed.	Multi-signature smart contracts, distributed smart contracts, homomorphic encryption, multi-party computation	**Advantages:** Reduces the risks of centralized servers. Cross-domain privacy protection and identity authentication enhance system security. Model sharing instead of data sharing lowers the risk of data privacy breaches. **Disadvantages:** High communication costs and latency; limited model convergence efficiency.
Communication-efficient learning of deep networks from decentralized data https://arxiv.org/abs/1602.05629	High communication costs.	Federated learning based on iterative model averaging.	**Advantages:** Simple and easy to implement; robust to imbalanced and non-IID (independent and identically distributed) data distributions. **Disadvantages:** Does not consider security; weak against adversarial attacks.
Blockchained On-Device Federated Learning https://ieeexplore.ieee.org/document/8733825	Lacks authentication mechanisms, vulnerable to model information leakage attacks.	BlockFL, PoW (Proof of Work).	**Advantages:** Overcomes single point of failure issues; uses blockchain to ensure model validation, improving security. **Disadvantages:** PoW consumes substantial computing resources; lacks identity verification for communication requests.
Privacy-preserving blockchain-based federated learning for IoT devices https://ieeexplore.ieee.org/document/9170559	Communication vulnerabilities; relies on network connectivity.	Blockchain, Differential Privacy (DP), FL-based crowdsourcing scheme.	**Advantages:** Ensures model privacy, validated using blockchain and DP. **Disadvantages:** High communication overhead when involving millions of clients; unstable networks may lead to data loss or communication timeouts.
FL_GIoT: Federated Learning Enabled Edge-Based Green Internet of Things System https://ieeexplore.ieee.org/document/10323400	Privacy risks during model updates.	Edge computing, federated learning.	**Advantages:** Improves GIoT system efficiency and cost-effectiveness; reduces latency and energy consumption. **Disadvantages:** Frequent communication in federated learning increases energy consumption and latency; edge devices are more susceptible to attacks.
S2E-DECI: Secrecy and Energy-Efficient Dual-aware Device-Edge Co-Inference for AIoT	Limited exploration of physical-layer security against eavesdropping attacks; sensitive to dynamic communication	Split learning with physical layer security、Distributed Reinforcement Learning-based Joint Optimization	**Advantages** : Enhances energy efficiency by optimizing DNN partitioning and resource allocation 、Achieves near-optimal performance comparable to exhaustive search methods **Disadvantages** : Requires significant computational resources for training and inference

https://ieeexplore.ieee.org/document/10713243	and computation constraints.		
FedDDoS_An_Efficient_Federated_Learning-based_DDoS_Attacks_Classification_in_SDN-Enabled_IIoT_Networks	Privacy concerns in data transfer	FL-based model, PCC FS technique	Advantages include efficient DDoS classification achieving 98.37% accuracy and a privacy-preserving federated learning model for industrial data security. However, traditional DDoS classifiers degrade with increased features.
DDoS_Attacks_Mitigation_in_5G-V2X_Networks_A_Reinforcement_Learning-Based_Approach	Lack of 5G-specific datasets	RL-based approach, Q-learning	Advantages include RL-based approaches that outperform random actions in mitigating DDoS attacks, while disadvantages arise from DDoS attacks modifying communication protocols, posing challenges for mitigation.
Applying Federated Learning to ICMP Flood Attack Detection in Industrial Control System	Communication costs and data security risks in centralized ML	MLP model, data augmentation	Enhances recall rate for detecting ICMP flood attacks in ICS, federated learning mitigates communication costs and data security concerns, and data augmentation improves classification performance
Identifying Distributed Denial of Service Attacks through Multi-Model Deep Learning Fusion and Combinatorial Analysis https://dl.acm.org/doi/10.1007/s10922-024-09882-0	The need for large amounts of data, and the challenge of latency.	Four deep neural network models are developed, being fused using CFA	Inherent latency may hinder real-time response to threats, especially in scenarios where attacks need to be intercepted in a real-time environment.
Advancing DDoS Attack Detection: A Synergistic Approach Using Deep Residual Neural Networks and Synthetic Oversampling https://arxiv.org/abs/2401.03116	Class imbalance in DDoS datasets requires oversampling、Computationally intensive for large-scale datasets	ResNet / SMOTE	**Advantages** : Effectively balances datasets, reducing bias toward benign traffic、Detects subtle and low-volume attack patterns、Adaptable to dynamic cyber threats **Disadvantages :** Training requires significant computational resources 、Sensitive to parameter tuning
DDoS-Net: Classifying DDoS Attacks in Wireless Sensor Networks with Hybrid Deep Learning	The delay problem is compounded by the higher cost and effort required to deploy, operate and maintain these facilities.	LSTM / MLP	Large amount of computing resources, ineffective in dealing with real-time attacks or high latency.

https://ieeexplore.ieee.org/document/10534545			

2.Related Works

2.1. DDoS Attack Detection for ICS

DDoS attacks represent a significant threat to the availability of Industrial Control Systems (ICS) and have thus become a primary focus of detection efforts. In a recent study [13], machine learning and deep learning methods were applied, employing five algorithms, including multilayer perceptron (MLP), with the KDD-99 dataset for training and prediction. Another investigation [14] utilized the C++ programming language and ESP-01S device with a Wi-Fi module to generate both normal and malicious ICMP Echo requests. Subsequently, three machine learning algorithms were employed to construct a framework for detecting ICMP Flood attacks. Moreover, in a separate study [15], ICMP Flood attacks were conducted on sensors measuring temperature, humidity, gas, and water level. The K-means algorithm was employed for clustering, leading to the identification of two critical features for attack identification: the length of the Ethernet packet header and the flag of the IP packet header.These studies collectively underscore the growing interest and multifaceted approaches employed to combat DDoS threats in industrial settings.

2.2. Federated Learning

Federated learning, initially introduced by Google in 2016 [19], revolutionizes machine learning paradigms by leveraging local data on mobile devices to train machine learning models. Only the updates of model weights are transmitted to the aggregation server for computation, thus enhancing communication efficiency, reducing network latency, and bolstering security and privacy protection [20]. In domains spanning e-commerce, social platforms, financial services, and healthcare, federated learning offers avenues to surmount challenges related to intellectual property rights, privacy, security, and administrative processes. Moreover, federated learning facilitates the resolution of issues stemming from data silos and heterogeneity, thereby fostering collaborative data analysis and knowledge sharing [9]. In the realm of federated learning, recent research efforts have been dedicated to exploring its application. For instance, in [16], a CNN-MLP model was adopted, incorporating skip connection and factorized convolution techniques to mitigate gradient vanishing and reduce computational resource requirements. Furthermore, [17] introduced FLEAM (Federated Learning Empowered Architecture to Mitigate DDoS), which comprises four components: MPM (monitor and policing module), LAM (local analysis module), DPM (DDoS policy module), and aggregator. FLEAM integrates principles of Fog Computing and Edge Computing to bolster the detection and defense capabilities of decentralized Industrial IoT (IIoT) environments against DDoS attacks. As such, federated learning emerges as a transformative approach with far-reaching implications for diverse sectors and domains.

3. Conceptual Framework Design

3.1. Application scenario

In the illustrated application scenario depicted in Figure 1, the focus lies on a manufacturing enterprise that has undergone digital transformation to embrace smart manufacturing practices. Within this context, the company operates two distinct facilities, namely the legacy factory A and the modernized factory B, situated in disparate locations. These facilities are interconnected via Ethernet leased lines, facilitating data exchange and operational coordination. However, the legacy factory A faces considerable security vulnerabilities attributed to its reliance on outdated operational technology (OT) devices, which lack robust encryption and authentication mechanisms. Consequently, factory A becomes a prime target for malicious network traffic attacks orchestrated by hackers. These attacks, predominantly in the form of Distributed Denial-of-Service (DDoS) assaults, disrupt plant operations and undermine the availability of critical production resources. This scenario underscores the imperative for robust cybersecurity measures within smart manufacturing environments, particularly in addressing legacy infrastructure vulnerabilities and fortifying network defenses against evolving cyber threats.

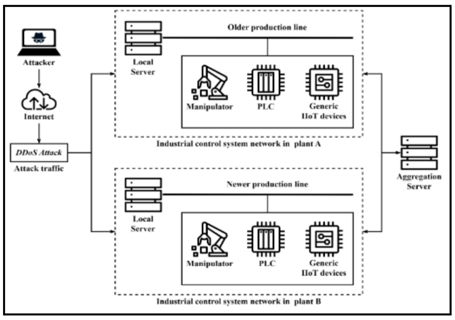

Figure 1. Application Scenario

3.2. Introduction to the Dataset

The dataset utilized for both training and testing in this study is derived from the Edge-IIoTset [12], a comprehensive dataset designed to emulate real-world IoT and Industrial IoT (IIoT) environments. The IoT data are generated from various IoT devices (more than 10 types) such as Low-cost digital sensors for sensing temperature and humidity, Ultrasonic sensor, Water level detection sensor, pH Sensor Meter, Soil Moisture sensor, Heart Rate Sensor, Flame Sensor, ...etc.). This dataset contains diverse data, including normal and malicious network traffic, generated by devices operating under protocols such as Hypertext Transfer Protocol (HTTP), Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP). These protocols are integral to understanding Distributed Denial-of-Service (DDoS) attack dynamics, making this dataset an ideal choice for evaluating detection mechanisms.

In alignment with the practical context of industrial control systems, this study broadens its scope beyond predefined scenarios. Alongside incorporating the DDoS_ICMP data representative of ICMP Flood attacks from the dataset, all data pertaining to DDoS attacks and normal network traffic are included for training and testing purposes. Table 1 illustrates the distribution of entries across the dataset to facilitate comprehensive analysis. Given the emphasis on assessing classification performance within data silos and the utilization of federated learning, the system configuration entails two physical hosts, designated as Client1 and Client2, simulating the roles of the old factory A and the new factory B, respectively. The system configuration for the new factory A and new factory B mirrors that of the old factory A and new factory B, respectively. This approach ensures a nuanced examination of classification capabilities under realistic conditions prevalent in industrial control settings, fostering a robust understanding of the effectiveness and adaptability of federated learning techniques.

3.3. Data preprocessing

Data preprocessing is one of the key steps before deep learning, which is used to organize the incorrect or incomplete data to improve the performance in the model training and prediction stage, and the data preprocessing in this study is mainly divided into the following steps:

The initial phase involves consolidating the .csv files containing normal network traffic data from both IoT and IIoT categories within the dataset, as outlined in Table 2 These files are merged into a unified .csv file, denoted as "Normal," under the newly introduced labeling field "Type". Subsequently, the network traffic associated with DDoS attacks, delineated across four distinct types as depicted in Table 1 within Section 3.2, undergoes a similar process. The corresponding .csv files are labeled with the respective attack types under the designated column "Type". This meticulous categorization procedure ensures systematic organization and classification of network traffic data, facilitating subsequent analyses and model training within the study's framework.

Table 2. Normal Network Traffic Files

File Name	
Distance.csv	phValue.csv
Flame_Sensor.csv	Soil_Moisture.csv
Heart_Rate.csv	Sound_Sensor.csv
IR_Receiver.csv	Temperature_and_Humidity.csv
Modbus.csv	Water_Level.csv

The subsequent phase of the process underscores the criticality of Data Cleaning, acknowledging the profound impact of data quality on model performance. This step prioritizes the eradication of any "Nan" values present in the dataset that deviate from a numeric type. Furthermore, in alignment with Ferrag et al. (2022) and the insights delineated in Table 3 [22], features closely correlated with the context of network traffic generation are selectively eliminated. This strategic refinement aims to enhance the model's generalization capabilities while mitigating the risk of overfitting.

Table 3. Features Removed

Name		
arp.dst.proto_ipv4	http.request.uri.query	tcp.dstport
arp.src.proto_ipv4	icmp.transmit_timestamp	tcp.options
frame.time	ip.dst_host	tcp.payload
http.file_data	ip.src_host	tcp.srcport
http.request.full_uri	mqtt.msg	udp.port

In the subsequent phase, the third step of the experimental process is initiated, which involves the standardization of data to expedite learning and convergence within the deep learning model predicated on the gradient descent algorithm. Initially, to preempt any misinterpretation stemming from the model's inability to directly process non-numeric features and to effectuate the conversion of feature fields into numeric values, features of string type, as delineated in Table 4, are amalgamated into a unified list. Subsequently, these features are encoded into binary vectors utilizing the One Hot Encoder methodology. To ensure homogeneity across disparate features and mitigate potential discrepancies affecting convergence speed and modeling algorithm performance, all features are subjected to processing alongside the remainder of the dataset using the Standard

Scaler technique. In this formulation, 'x' represents the original data, 'μ' signifies the mean value of the feature, 'σ' denotes the standard deviation of the feature, and 'X' represents the standardized data. Following standardization, the data distribution exhibits uniformity across various features, characterized by a mean of 0 and a standard deviation of 1. This standardization fosters enhanced model algorithm performance. The Standard Scaler formula is depicted below:

$$X\_StandardScaler = ((x - \mu))/\sigma \tag{1}$$

Table 4. Features of String Types

Name	
arp.hw.size	tcp.ack_raw
icmp.checksum	tcp.checksum
icmp.seq_le	tcp.connection.fin
http.request.method	tcp.flags
http.referer	tcp.len
http.request.version	dns.qry.name

In the concluding phase, the fourth step of the experiment incorporates the Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic (ADASYN) methods as delineated in Section 3.4 for DDoS_ICMP. This step aims to augment the dataset by generating synthetic samples equivalent to 20%, 40%, 60%, 80%, and 100% of the original data size. The objective is to address class imbalance arising from inadequacies in the original dataset size. Through this augmentation process, the original dataset is expanded to ensure robustness in handling variations in class distributions.

3.4. Data Augmentation

ADASYN (Adaptive Synthetic) sampling, introduced by He et al. [23], is a data augmentation technique designed to address class imbalance by drawing inspiration from SMOTE [24] and other similar methods. Class imbalance denotes situations where certain classes possess significantly more or fewer samples than others, potentially leading to model overfitting towards the majority class while neglecting minority classes during training.

At the heart of the ADASYN methodology lies the process of oversampling minority classes, particularly focusing on regions proximal to samples from the majority class in the feature space. By generating synthetic samples for underrepresented classes, ADASYN aims to enhance classifier performance. The generation of synthetic samples is contingent upon the proximity of minority class samples, thereby bolstering the classifier's ability to accurately categorize instances within boundary decision regions.

3.5. Research Method

This study uses the horizontal federated learning architecture to realize the configured application scenarios through three physical hosts: the central server, the regional server of the old domain and the new domain, and is represented by Server, Client 1, and Client 2.

3.5.1. Method of Centralized Deep Learning in Each Host

Firstly, we simulate the "data island" scenario, perform deep learning and evaluate the classification model performance on the local machine, as shown in Figure 2 and the following steps.

1) Step 1: Sample the dataset after preprocessing and split it into training data and test data, where the training data will be split and assigned to Client1 and Client2 respectively.
2) Step 2: Use the deep learning model MLP to train with the training data.
3) Step 3: Evaluate the classification performance of the model using the test data, which will be used as the basis for adjusting the structure of each model and the hyperparameters (batch size, epoch, learning rate, etc.).
4) Step 4: Based on the evaluation results of the previous step, configure the adjusted model and hyperparameters, and repeat the training and testing to obtain the best classification model.

3.5.2. Method of Federated Learning

Next, this study uses horizontal federated learning to investigate the effectiveness of the classification model by sharing the training of global model weights, as shown in Figure 3 and the following steps.

1) Step 1: After the two Clients are configured with the same deep learning model MLP, the Server and the two Clients are started sequentially.
2) Step 2: The two Clients send the local model weights obtained from each round of training to the Server.
3) Step 3: Server uses FedAvg algorithm to aggregate the weights from the two Clients.
4) Step 4: Server sends the aggregated weights to both Clients.
5) Step 5: The two Clients receive the weights from the global model to update their respective models and continue training using the gradient descent method.

After obtaining the performance of the federated learning classification model, ADASYN is used to enhance the data and repeat steps 1 to 5 to obtain the performance of the classification model after applying the data augmentation.

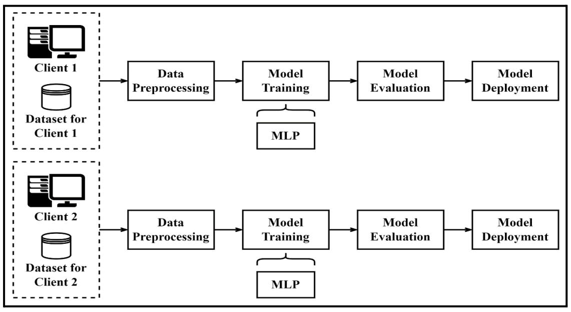

Figure 2. Method of Centralized Deep Learning in Each Host

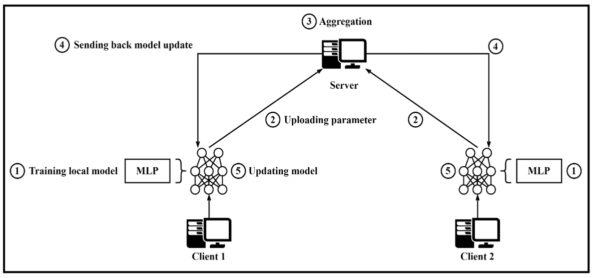

Figure 3. Method of Federated Learning

4. Experiment Results

4.1. Experimental Settings

The physical hosts utilized in this investigation, as delineated in Table 5, are equipped with the Windows 11 operating system and employ the Anaconda package management tool. These hosts operate on the Python programming language and leverage deep learning-related libraries, primarily developed using the Tensorflow and Keras frameworks. Tensorflow stands as an open-source library, while Keras functions as an API layered atop TensorFlow. *2.2.2. Federated Learning Framework*

Flower, an open-source framework, adopts a modular architecture design [21]. This modularization of aggregation policies, communication mechanisms, and training functions enhances scalability and flexibility, enabling adaptation to diverse machine learning frameworks and expanding the scope of federated learning applications. Moreover, to facilitate experimentation with the federated learning framework, all physical hosts have integrated the API code of the Flower framework for the implementation of federated learning functionalities.

Table 5. Experimental Software Configuration

Operating system	Microsoft Windows 11
Framework	Flower 1.5.0
Package management	Anaconda 23.3.1
Programming language	Python 3.11.4
Libraries and extensions	datetime, gc, itertools, logging, matplotlib, numpy, pandas, six, sklearn, tensorflow, time

In this study, the federated learning segment of the experiment involves the utilization of three physical hosts to simulate application scenarios: Server, Client 1, and Client 2, each with distinct hardware specifications outlined in Table 6 Client 1, representing the old factory, and Client 2, representing the new factory, are primarily tasked with training client models. Meanwhile, the Server is chiefly.

Table 6. Experimental Hardware Configuration

Server	CPU	Intel® Core™ i5-13400 4.60 GHz
	RAM	16GB
Client 1	CPU	AMD Ryzen™ 5 3600 4.1GHz
	RAM	32GB
Client 2	CPU	Intel® Core™ i7-12700 4.90 GHz
	RAM	32GB

The experimental setup depicted in Figure 4 facilitates communication among three physical hosts through an Ethernet network and a switch. Initially, these hosts are segmented into the same LAN (sub-network) and subsequently linked to a router providing Internet access. Moreover, communication among the three physical hosts is facilitated by gRPC, supported by Flower. This configuration enables all hosts to resolve IP addresses and port numbers, thereby facilitating the training of federated learning models within a decentralized environment.

For Hyperparameter, the configurations are shown in Table 7, Table 8 and Table 9

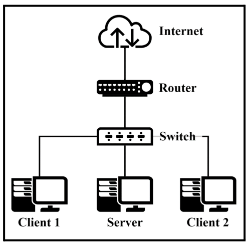

Figure 4. Experimental Network Configuration

Table 7. Configuration of Centralized Deep Learning in Each Host

Hyperparameter	Value
Batch Size	64
Epoch	25
Learning Rate	0.0005

Table 8. Configuration of Federated Learning

Hyperparameter	Value
Batch Size	64
Epoch	25
Round	30
Learning Rate	0.0005

Table 9. Other common configurations

Hyperparameter	Value
Activation Function	ReLU
Classification Function	Softmax
Hidden Layers	2
Optimizer	Adam

4.2. Performance Indicators

To assess and evaluate the performance of the experimental classification model in this study, a confusion matrix, as illustrated in Table 10, was employed to delineate the correspondence between the predicted outcomes of the classification model and the actual labels. These labels encompass true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

Table 10. Confusion Matrix

Confusion Matrix		Actual Condition	
		Negative	Positive
Predicted Condition	Negative	True Negative	False Negative
	Positive	False Positive	True Positive

The performance metrics such as Accuracy, Precision, Recall and F1-score obtained by confusion matrix computation are shown below.

The accuracy is the proportion of the number of samples correctly predicted by the model to the total number of samples, and is calculated as follows:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{1}$$

The precision is the proportion of samples predicted to be positive by the model that are also actually true positive (TP), and is calculated as follows:

$$Precision = TP/(TP + FP) \tag{2}$$

Recall is the proportion of all samples in which the model is actually positive that are correctly predicted to be positive and is calculated as:

$$Recall = TP/(TP + FN) \qquad (3)$$

The F1-Score is a comprehensive performance metric. The closer the value is to 1, the better the performance of the classification model, and the impact of extreme values is reduced by calculating the summed average of the precision and recall, which is calculated as follows:

$$F1 - score = ((2 * Precision * Recall))/((Precision * Recall)) \qquad (4)$$

4.3. Classification Model Performance

4.3.1. Applying Deep Learning in Each Host

According to Table 11, Client1, configured with a higher volume of DDoS attack traffic, exhibits superior performance with an accuracy of 93.13% compared to Client2's accuracy of 90.49%.

Table 11. The Accuracy of Deep Learning Method in Each Host

Client	Accuracy
1	0.9313
2	0.9049

As evidenced in Tables 12 and 13, the classification models deployed on Client1 and Client2 demonstrate remarkable precision of 100% in identifying TCP Flood attacks, exhibiting no instances of false positives. Furthermore, both models exhibit a recall rate of over 99% for classifying HTTP Flood, UDP Flood attacks, and normal network traffic. However, their recall rates for ICMP Flood attacks are notably lower, standing at 67.96% and 55.30% respectively, indicating a comparative inefficacy in identifying ICMP Flood attacks. This underscores the necessity for enhancements in the classification models to address the challenges associated with identifying ICMP Flood attacks effectively.

Table 12: Performance of Client1 for Deep learning.

Client1			
Label	Precision	Recall	F1-score
DDoS_HTTP	0.9994	0.9982	0.9988
DDoS_ICMP	0.9981	0.6796	0.8086
DDoS_TCP	1.0000	1.0000	1.0000
DDoS_UDP	0.7736	0.9994	0.8721
Normal	0.9975	0.9988	0.9982

Table 13: Performance of Client2 for Deep learning.

Client2			
Label	Precision	Recall	F1-score
DDoS_HTTP	1.0000	0.9994	0.9997
DDoS_ICMP	1.0000	0.5530	0.7121
DDoS_TCP	1.0000	1.0000	1.0000
DDoS_UDP	0.7101	0.9994	0.8303
Normal	0.9975	1.0000	0.9988

4.3.2. Applying Federated Learning

Upon applying federated learning and observing the convergence of Client1 and Client2, as depicted in Figure 5, it becomes apparent from Figure 6 that Client1 and Client2 achieve notable accuracy improvements. Specifically, Client1 demonstrates an accuracy of 99.53%, representing a significant increase of approximately 6.87%, while Client2 achieves an accuracy of 98.31%, reflecting an enhancement of about 8.64%.

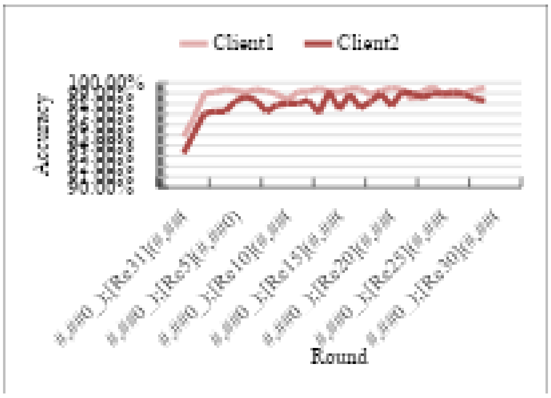

Figure 5. Accuracy of Federated Learning Method

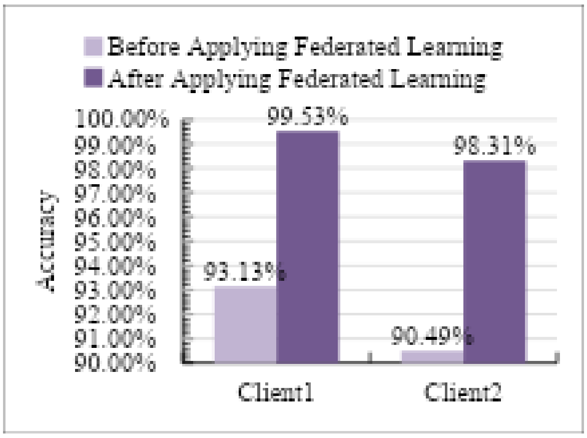

Figure 6. Accuracy Comparison Before and After Adopting Federated Learning

As indicated in Table 14 and Table 15, the recall metrics for Client1 witness a remarkable increase from 67.96% to 98%, while Client2 experiences an escalation from 55.30% to 92.25%. These enhancements denote substantial improvements of approximately 44.20% and 66.82% for Client1 and Client2, respectively. Such advancements notably fortify the recall capabilities of the classification model against ICMP Flood attacks through the application of federated learning.

Table 14. Comparison of Recall before and after Federated Learning in Client1

Client1		
Label	Recall	
	Before FL	After FL
DDoS_HTTP	0.9982	0.9994
DDoS_ICMP	0.6796	0.9800
DDoS_TCP	1.0000	1.0000
DDoS_UDP	0.9994	0.9994
Normal	0.9988	0.9988

FL (Federated Learning).

Table 15. Comparison of Recall before and after Federated Learning in Client2

Client2		
Label	Recall	
	Before FL	After FL
DDoS_HTTP	0.9994	0.9994
DDoS_ICMP	0.5530	0.9225
DDoS_TCP	1.0000	1.0000
DDoS_UDP	0.9994	0.9994
Normal	1.0000	0.9988

FL (Federated Learning).

4.3.3. Applying Data Augmentation and Federated Learning

The application of SMOTE and ADASYN for generating synthetic samples at varying proportions (20%, 40%, 60%, 80%, and 100%) and their incorporation with the original data for federated learning training is illustrated in Figure 7 Notably, the retraining outcomes demonstrate that ADASYN yields superior performance compared to SMOTE. Specifically, as depicted in Figure 8, the configuration employing ADASYN at 60% for both Client1 and Client2 attains the highest Accuracy rates of 99.97% and 99.96%, respectively. This represents an enhancement of approximately 0.44% and 1.68% in accuracy. Consequently, the study exclusively adopts ADASYN to achieve optimal accuracy in the experimental setting.

According to the findings presented in Table 16 and Table 17, the preemptive integration of ADASYN also enhances the recall performance of the classification model against ICMP Flood attacks. Specifically, for Client1, the recall rate increases from 98% to 100%, while for Client2, it rises from 92.25% to 99.94%. These improvements represent increments of approximately 2.04% and 8.34%, respectively.

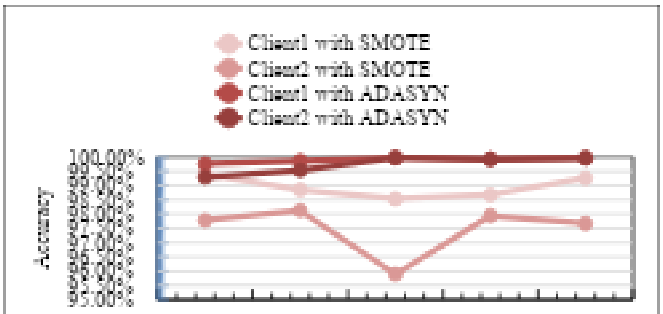

Figure 7. Proportion and Accuracy of Synthetic Samples Generated by Data Enhancement

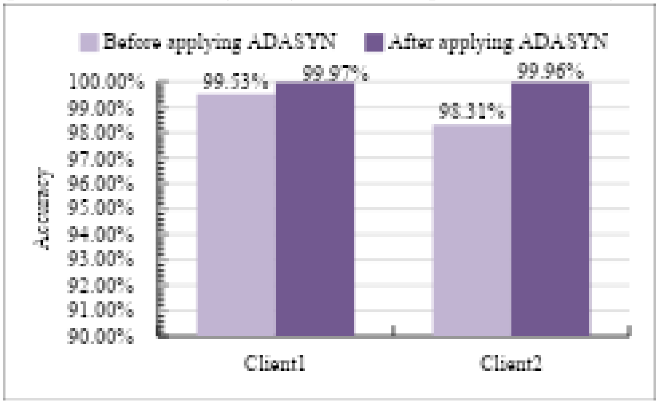

Figure 8. Comparison of Accuracy Methods Before and After Applying ADASYN

Table 16. Comparison of Recall Before and After Applying ADASYN in Client1

Client1		
Label	Recall	
	Before ADASYN	After ADASYN
DDoS_HTTP	0.9994	0.9994
DDoS_ICMP	0.9800	1.0000
DDoS_TCP	1.0000	1.0000
DDoS_UDP	0.9994	1.0000
Normal	0.9988	0.9988

Table 17. Comparison of Recall Before and After Applying ADASYN in Client2

Client2		
Label	Recall	
	Before ADASYN	After ADASYN
DDoS_HTTP	0.9994	1.0000
DDoS_ICMP	0.9225	0.9994
DDoS_TCP	1.0000	1.0000
DDoS_UDP	0.9994	0.9994
Normal	0.9988	0.9988

The findings delineated in Table 18 underscore the efficacy of the three experimental methodologies employed in this study. They demonstrate a progressive enhancement in the performance of the classification model, achieved through successive phases: centralized learning, federated learning alone, and preliminary integration of ADASYN followed by federated learning training. Notably, this progression culminates in a substantial increase in the recall rate of the classification model for detecting ICMP floods. Specifically, the recall rate of the classification model in detecting ICMP floods escalates from 67.96% to 100% for Client1 and from 55.30% to 99.94% for Client2. These improvements represent increments of approximately 47.15% and 80.72%, respectively.

Table 18. Comparison of Recall in Three Experimental Methods

Methods	Client	Recall
With Deep Learning in Each Host	1	0.6796
	2	0.5530
With Federated Learning	1	0.9800
	2	0.9225
With ADASYN and Federated Learning	1	1.0000
	2	0.9994

5. Conclusion

This study examines the performance disparities among three methodologies: centralized deep learning within individual hosts, federated learning, and the subsequent application of the data augmentation technique ADASYN to produce synthetic samples. It focuses on the detection of ICMP Flood attacks post-training in Industrial Control Systems (ICS) composed of Internet of Things (IoT) and Industrial Internet of Things (IIoT). Using the Flower framework and the Edge-IIoTset dataset, three physical hosts were set up to simulate application scenarios and implement the federated learning framework. Initially, ADASYN was employed to enrich the training data, followed by federated learning to update the weights of the overarching model and iterate the training process. This approach effectively enhances the recall rate of the classification model for detecting ICMP flood attacks. Specifically, Client1's recall rate increased from 67.96% to 100%, while Client2's rose from 55.3% to 99.94%.

References

[1] Craggs B, Rashid A, Hankin C, Antrobus R, Şerban O, Thapen N. A Reference Architecture for IIoT and Industrial Control Systems Testbeds. Living in the Internet of Things (IoT 2019), London, UK: IET; 2019. p. 1-8. https://doi.org/10.1049/cp.2019.0169.

[2] Khalil R A, Saeed N. Network Optimization for Industrial Internet of Things (IIoT). IEEE Sens. Lett. 2020;4(7):1-4. https://www.doi.org/10.1109/LSENS.2020.3002232.

[3] Garimella P K. IT-OT Integration Challenges in Utilities. 2018 IEEE 3rd Int. Conf. Comput., Commun., and Sec. (ICCCS), Kathmandu, Nepal.: IEEE; 2018. p. 199–204. https://www.doi.org/10.1109/CCCS.2018.8586807.

[4] Guo G, Zhuge J, Yang M, Zhou G, Wu Y. A Survey of Industrial Control System Devices on the Internet. 2018 Int. Conf. Internet of Things, Embed. Syst., and Commun. (IINTEC), Hamammet, Tunisia.: IEEE; 2018. p. 197-202. https://www.doi.org/10.1109/IINTEC.2018.8695276.

[5] Yilmaz E N, Ciylan B, Gönen S, Sindiren E, Karacayılmaz G. Cyber security in industrial control systems: Analysis of DoS attacks against PLCs and the insider effect. 2018 6th Int. Istanb. Smart Grids and Cities Congress and Fair (ICSG), Istanbul, Turkey: IEEE; 2018. p. 81–85. https://doi.org/10.1109/SGCF.2018.8408947

[6] Yaacoub J P A, Noura H N, Salman O, Chehab A. Ethical hacking for IoT: Security issues, challenges, solutions and recommendations. Internet Things Cyber-Phys. Syst. 2023;3:280-308. https://doi.org/10.1016/j.iotcps.2023.04.002.

[7] Chaudhary S, Mishra P K. DDoS attacks in Industrial IoT: A survey. Comput. Netw. 2023;236:no.110015. https://doi.org/10.1016/j.comnet.2023.110015.

[8] Koay A M Y, Ko R K L, Hettema H, Radke K. Machine Learning in Industrial Control System (ICS) Security: current landscape, opportunities and challenges. Journal of Intell. Inf. Syst. 2023;60:377-405. https://doi.org/10.1007/s10844-022-00753-1.

[9] Yang Q, Liu Y, Chen T, Tong Y. Federated Machine Learning: Concept and Applications. ACM Trans Intell. Syst. Technol. 2019;10(2):1-19. https://doi.org/10.1145/3298981.

[10] Staňo M, Hluchý L, Bobák M, Krammer P, Tran V. Federated Learning Methods for Analytics of Big and Sensitive Distributed Data and Survey. 2023 IEEE 17th Int. Symp. on Appl. Comput. Intell. and Inform. (SACI). Timisoara, Romania: IEEE; 2023. p. 705-10. https://www.doi.org/10.1109/SACI58269.2023.10158622.

[11] Truong N, Sun K, Wang S, Guitton F, Guo Y. Privacy Preservation in Federated Learning: An Insightful Survey from The GDPR Perspective. ArXiv Prepr. 2011. https://doi.org/10.48550/arXiv.2011.05411.

[12] 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: Centralized and Federated Learning. IEEE Dataport; 2023.Available: https://dx.doi.org/10.21227/mbc1-1h68.

[13] Patil P S, Deshpande S L, Hukkeri G S, Goudar R H, Siddarkar P. Prediction of DDoS Flooding Attack Using Machine Learning Models. 2022 Third Int. Conf. on Smart Technol. in Comput., Electri. and Electron. (ICSTCEE). Bengaluru, India: IEEE; 2022. p. 1-6. https://doi.org/10.1109/ICSTCEE56972.2022.10100083.

[14] Almorabea O M, Khanzada T J S, Aslam M A, Hendi F A, Almorabea A M. IoT Network-Based Intrusion Detection Framework: A Solution to Process Ping Floods Originating from Embedded Devices. IEEE Access. 2023;11:119118-45. https://www.doi.org/10.1109/ACCESS.2023.3327061.

[15] Stiawan D, et al. Ping Flood Attack Pattern Recognition Using a K-Means Algorithm in an Internet of Things (IoT) Network. IEEE Access. 2021;9:116475-84. https://www.doi.org/10.1109/ACCESS.2021.3105517.

[16] Zainudin A, Akter R, Kim D-S, Lee J-M. FedDDoS: An Efficient Federated Learning-based DDoS Attacks Classification in SDN-Enabled IIoT Networks. 2022 13th Int. Conf. on Inf. and Commun. Technol. Convergence (ICTC). Jeju Island, Korea, Republic of.: IEEE; 2022. p. 1279–83. https://www.doi.org/10.1109/ICTC55196.2022.9952610.

[17] Li J, Lyu L, Liu X, Zhang X, Lyu X. FLEAM: A Federated Learning Empowered Architecture to Mitigate DDoS in Industrial IoT. IEEE Trans. Ind. Inform. 2022;18(6):4059-68. https://www.doi.org/10.1109/TII.2021.3088938.

[18] Lim W Y B, Luong N C, Hoang D T, Jiao Y, Liang Y C, Yang Q, Niyato D, Miao C. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. IEEE Commun. Surv. Tutor. 2020;22(3):2031-63. https://www.doi.org/10.1109/COMST.2020.2986024.

[19] McMahan H B, Moore E, Ramage D, Arcas B A y. Federated Learning of Deep Networks using Model Averaging. ArXiv Prepr. 2016. https://doi.org/10.48550/arXiv.1602.05629.

[20] Li T, Sahu A K, Talwalkar A, Smith V. Federated Learning: Challenges, Methods, and Future Directions. IEEE Signal Process. Mag. 2020;37(3):50-60. https://www.doi.org/10.1109/MSP.2020.2975749.

[21] Beutel D J, Topal T, Mathur A, Qiu X, Fernandez-Marques J, Gao Y, Sani L, Li K H, Parcollet T, de Gusmão P P B, Lane N D. Flower: A Friendly Federated Learning Research Framework. ArXiv Prepr. 2020. https://doi.org/10.48550/arXiv.2007.14390

[22] 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;10:40281-306. https://www.doi.org/10.1109/ACCESS.2022.3165809.

[23] He H, Bai Y, Garcia E A, Li S. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. 2008 IEEE Int. Joint Conf. on Neural Netw. (IEEE World Congress on Comput. Intell.). Hong Kong, China: IEEE; 2008. p. 1322-8. https://www.doi.org/10.1109/IJCNN.2008.4633969.

[24] Chawla N V, Bowyer K W, Hall L O, Kegelmeyer W P. SMOTE: Synthetic Minority Oversampling Technique. Journal of A.I. Res. 2002;16:321-57.