EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems
Research Article EAI.EU

A Multimodal Swarm Learning Approach for DDoS Detection in Internet of Things Infrastructure

Thuat Nguyen-Khanh[1,2,*], Anh Pham-Nguyen-Hai[1,2], Luan Van-Thien[1,2], Quan Le-Trung[1,2]

[1]Faculty of Computer Networks and Communications, University of Information Technology, Ho Chi Minh City, Viet Nam
[2]Vietnam National University, Ho Chi Minh City, Vietnam (VNU-HCM)

Abstract

The Internet of Things (IoT) has emerged as a foundational platform for driving intelligent solutions, playing a central role in the Fourth Industrial Revolution. Its potential lies in enabling seamless connectivity and real-time data exchange among diverse devices and systems, thereby powering advanced applications such as intelligent transportation, smart healthcare, precision agriculture, and automated manufacturing. These solutions promise to improve efficiency, optimize resource utilization, and enhance decision-making across various sectors. However, this potential is challenged by some issues, including security vulnerabilities, privacy concerns, and significant heterogeneity arising from the vast diversity of devices, communication protocols, and data formats. In this paper, we develop a multimodal deep learning solution to detect DDoS attacks on IoT infrastructure based on two data types: packet-based data and flow-based data. Firstly, the datasets containing packets labeled as benign or attack are processed into two branches: packet-based and flow-based features. Then, each branch is trained using two independent CNN models. Finally, the feature information extracted from both modalities is fused and fed into a concatenation-based classifier for DDoS attack detection. Experimental results on Edge-IIoTset and CiCIoMT2024 datasets indicate that the multimodal deep learning model within a decentralized machine learning architecture achieves performance comparable to centralized machine learning. In addition, our proposal is also robust to non-independent and identically distributed (non-IID) data in decentralized machine learning architecture.

Received on 15 August 2025; accepted on 22 December 2025; published on 29 December 2025

Keywords: DDoS, Decentralized Machine Learning, multimodal, CNN, Swarm Learning

doi:10.4108/eetinis.131.9961

1. Introduction

The global demand for Internet of Things (IoT) devices grows annually. According to a recent report by IoT Analytics[1], the global IoT market is projected to expand from USD 269 billion in 2023 to USD 690 billion by 2030. Alongside this rapid growth, IoT systems face increasing security threats, notably Distributed Denial of Service (DDoS) attacks. In response to such threats, machine learning (ML) and deep learning (DL) techniques have been extensively explored for cyberattack detection. With effective feature extraction from packet-level and flow-level data, recent studies have demonstrated improved accuracy and efficiency in detecting attacks on resource-constrained devices [1], including real-time detection capabilities [2]-[3]. However, most existing DL-based intrusion detection models, particularly DDoS attack detection, focus on a single aspect of network traffic, limiting their generalizability and effectiveness. Multimodal deep learning has emerged as a promising solution to address this limitation by leveraging multiple data representations. By integrating features from different modalities, multimodal models can deliver more accurate and comprehensive predictions compared to unimodal approaches [4]. Meanwhile, centralized learning (CL) architectures pose challenges in maintaining data privacy, as raw data must be transmitted from edge devices to third-party servers

*Corresponding author. Email: thuatnk@uit.edu.vn
[1]https://iot-analytics.com/iot-market-size/

for training. The volume of data involved further exacerbates this concern. Federated Learning (FL) has been proposed to mitigate these issues by enabling decentralized training across devices while preserving data locality [5]. However, FL remains dependent on a central coordinating server, introducing potential vulnerabilities. A single point of failure at the server can impact the entire system, and the presence of a central authority raises ongoing concerns about data security, malicious participants, and potential manipulation of shared model parameters [6]. To address these limitations, this paper proposes a DDoS detection system for IoT that integrates a multimodal deep learning model with the Swarm Learning (SL) framework [7]. The proposed architecture removes the need for a central server by utilizing blockchain technology for decentralized authentication and coordination. The multimodal approach enables the system to exploit both packet-based and flow-based features, enhancing detection accuracy and robustness in distributed IoT environments.

Table 1. List of abbreviations

Abbreviation	Description
AI	Artificial Intelligence
CL	Centralized Learning
CNN	Convolutional Neural Network
DDoS	Distributed Denial of Service
DL	Deep Learningv
FL	Federated Learning
IID	Independent and Identically Distributed
IIoT	Industrial Internet of Things
IoT	Internet of Things
ML	Machine Learning
MMDTL	Multimodal Deep Transfer Learning
MQTT	Message Queuing Telemetry Transport
NIDS	Network Intrusion Detection Systems
Non-IID	Non-Independent and Identically Distributed
SDN	Software-Defined Networking
SL	Swarm Learning
SN	Swarm Network

The main contributions of this paper are summarized as follows:

- The design and implementation of a distributed infrastructure leveraging edge computing to support the Swarm Learning architecture for training deep learning models;
- The development of multimodal deep learning models tailored to capture diverse characteristics of network traffic for DDoS attack classification;
- The experimental evaluation and analysis of distributed deep learning models in the context of DDoS attack detection.

The remainder of this paper is organized as follows: Section 2 reviews the relevant literature and examines existing technologies related to DDoS attack detection and machine learning architectures. Section 3 focuses on the design of decentralized machine learning and deep learning models that support multimodal data. Section 4 presents and discusses the experimental results concerning the effectiveness and performance of the proposed architectures. Finally, Section 5 concludes the paper and outlines potential directions for future work.

2. Related Works

This work lies at the intersection of two key research areas: Multimodal deep learning for DDoS attack detection, and machine learning architectures. The following section reviews prior studies in these domains, with a focus on multimodal deep learning approaches for NIDS and machine learning architectures relevant to our proposed framework.

2.1. Multimodal deep learning for DDoS attack dettection

Recent research on DDoS attack detection in IoT often focuses on refining and integrating deep learning models to improve their accuracy and effectiveness. RSG-MJ: Alshdadi et al. [8] introduced the ResNeSt-GRU Ensembler Approach (RSG-MJ), which utilizes a ResNeSt model with Split-Attention for intricate feature extraction and a GRU for temporal analysis, optimized using the Jaya Algorithm to achieve reduced computational complexity and superior accuracy across complex datasets, such as the NSL-KDD and CIC-IDS datasets. MLDNN: Abid et al. [9] proposed a Multilevel Deep Neural Network (MLDNN) approach, a hybrid CNN-LSTM architecture for detecting and classifying DDoS attacks across different protocols in SDN-supported IoT networks. This method focuses on combining the spatially learned features from the CNN module with the temporally learned features from the LSTM module to achieve superior accuracy compared to individual models.

On the other hand, the author of [10] and [11] focused on defining a new type of data and then using different modeling architectures on these newly created data types. Chang and Cao [10] proposed enhancing detection efficiency by recognizing that network traffic is multimodal (containing structured data like protocol text and nonstructured data like packets). Their approach uses lightweight feature extractors (e.g., language models and linear models) and fuses the

resulting features using a cross-attention mechanism. This parallel processing maintains inference speed while improving classification accuracy. DEEPShield: Saiyed and Al-Anbagi [11] developed the Deep Ensemble learning with Pruning (DEEPShield) system, a DL-based ensemble model integrating CNN and LSTM, explicitly designed for high- and low-volume DDoS detection on resource-constrained edge devices. This work contributed a new, flow-based dataset, HL-IoT, created from a real testbed to address the lack of publicly available data covering both high- and low-volume attacks. All of the studies above use centralized data sources to train deep learning models.

2.2. Machine Learning Architectures

Traditional centralized learning approaches aggregate data from multiple clients into a central server to facilitate the training of a global machine learning model. While this strategy enables the development of high-performance models, it introduces critical challenges related to data privacy, security risks, and system scalability due to centralized data handling and potential processing bottlenecks.

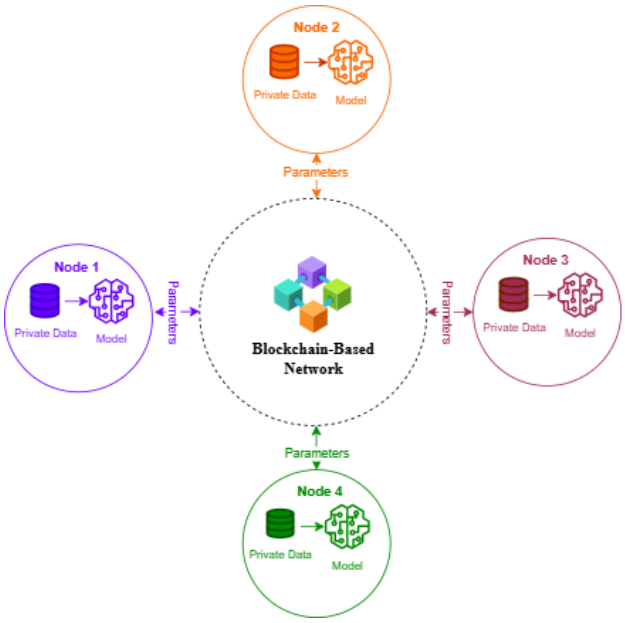

Figure 1. Overview of Swarm Learning Architecture

To address these limitations, decentralized and federated learning paradigms have emerged. Federated learning [12] allows clients to train local models on their private data and only share model updates with a central coordinator, thus mitigating privacy risks. However, it still relies on a central server to orchestrate communication and aggregation, which may become a single point of failure or control.

In contrast, decentralized learning eliminates the need for a central authority by enabling peer-to-peer coordination and collaborative model training across distributed nodes. Swarm Learning framework [7] further enhance this approach by combining blockchain-based consensus with privacy-preserving learning mechanisms, making it well-suited for sensitive domains such as healthcare and cybersecurity. Figure 1 illustrates the architecture of Swarm Learning.

The Swarm Learning framework supports three distinct model aggregation strategies for global model synthesis at each training round [13]: (1) *Mean*, based on the FedAVG approach, computes the weighted average of local model parameters; (2) *Coordmedian* sorts the model parameters across participants and selects the coordinate-wise median; and (3) *Geomedian* estimates the geometric median using Weiszfeld's algorithm. Table 2 shows the comparison of three merge methods on SL.

3. Methodology and Implementation

3.1. System Architecture

The architecture of the proposed Swarm Learning-based DDoS attack detection, depicted in Figure 2, is designed to detect cyberattacks within IoT environments. User data, generated by end devices (e.g., local Wi-Fi routers), is retained at the edge to preserve data privacy and enhance system security. A blockchain layer, composed of Swarm Network (SN) nodes, enforces participant legitimacy by permitting only authorized entities to join the decentralized learning network. Furthermore, each participant operates under a training smart contract that regulates their contributions and governs the distribution of rewards.

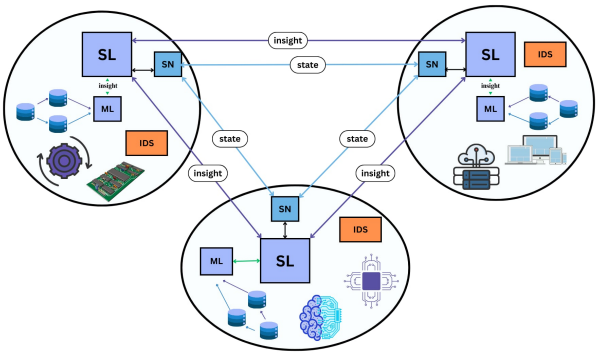

Figure 2. Overview of The Proposed Architecture

At each edge node, a Machine Learning (SL) component is paired with a corresponding Swarm Learning (SL) node to train a local model using locally available data. The parameters and weights resulting from this local training process are shared

Table 2. Comparison of Merge Methods in Swarm Learning [13]

Merge Method	Method Name	Aggregation Strategy	Key Characteristics	Advantages	Limitations
Mean	Weighted Federative Averaging	Sequentially aggregates pairs of model weights and parameters, then divides by the total sum of weights	Simple implementation; effective on IID data; sensitive to outliers	Fast computation; performs well in homogeneous data distributions	Prone to noise and biased nodes; degrades under non-IID conditions
Coordmedian	Coordinate-wise Median	Sorts and computes the median (50th percentile) for each parameter coordinate after transposing intermediate parameters	More robust to outliers than mean; independent of data distribution assumptions	Improved robustness to noisy updates; simpler than geometric median	May underperform in heterogeneous data; not globally optimal
Geomedian	Geometric Median (Weiszfeld Algorithm)	Iteratively minimizes the total distance to all nodes' parameters using Weiszfeld's algorithm	Highly resilient to outliers; minimizes impact of extreme values; computationally intensive	Most robust to adversarial nodes and non-IID data distributions	Computationally expensive; less scalable for large models or high node counts

across the SL network. During each training round, a designated leader node aggregates the received local models and redistributes the resulting global model to all participating clients for the next iteration.

Algorithm 1 outlines the execution flow of the proposed system. Edge devices join the SL blockchain network by submitting their IP addresses and port number. Upon joining, SL nodes register with their corresponding Swarm Network (SN) nodes. Raw data training is performed locally at the edge nodes, ensuring that data privacy is maintained throughout the learning process.

The SL library customizes the default model callback—supporting both Keras and PyTorch—through the SwarmCallback API to manage the model aggregation process. A key parameter of this API is the synchronization frequency, which defines the number of local training batches completed before model weights and parameters are shared.

At the end of each training round, a node is designated as the leader. This leader is responsible for collecting local models from participating nodes and aggregating them into a single global model by averaging their parameters and weights. In the event of a leader failure, a new leader is dynamically elected to ensure continuity of the aggregation process. If the previously failed leader recovers after a new leader has been assigned, it resumes participation as a regular SL node, contributing its parameters and weights to subsequent global model updates. The model is then sent back and updated at the edge nodes. This process repeats until the end of the last epoch.

Algorithm 1: The working process of Swarm Learning architecture

Input: *Set of SL nodes:* $C = \{C_i \mid i = 1, 2, \ldots, n\}$
Set of weightages of SL nodes: $W = \{W_i \mid i = 1, 2, \ldots, n\}$
F: the number of batches before sync
f: merging function
E: maximum epochs
Output: *Aggregated model for network intrusion detection*
Swarm on train begin:
  for *each node i* **do**
    Initialize model m_i from its dataset.
    Elect $C_k, k \in \{1, 2, \ldots, n\}$ as a leader.
  Leader collects and merges models and weightages of all nodes using f.
  Leader returns the merged model to all nodes.
while *maximum epochs are not reached* **do**
  Local training:
    $r = 1$.
    for $r \leq F$ **do**
      for *each node i* **do**
        Train its model m_i from its dataset.
      $r = r + 1$.
  Elect a leader, aggregate and update model parameters:
    for *each node i* **do**
      Elect $C_k, k \in \{1, 2, \ldots, n\}$ as a leader
      Leader collects and merges all models and weightages of all nodes using f.
      Leader returns the merged model to all nodes.
Return final global model.

3.2. Proposed Multimodal Deep Learning Approach

This paper proposes a multimodal deep learning-based DDoS detection system that utilizes both flow-based and packet-based traffic features, as illustrated in Figure 3. The proposal comprises two sub-branches: one processes flow-level data, while the other handles packet-level information. These complementary branches capture distinct aspects of network traffic to enhance detection accuracy in real-world scenarios. Both branches operate in parallel, each responsible for extracting and analyzing specific features. The outputs from these branches are subsequently fused at a hidden layer, which serves as the decision-making point of the model.

This paper uses two types of data, including packet data and stream data. The flow data was extracted and reconstructed from packet data using the LUCID [14] tool. The goal of this study is to evaluate the multimodal model on different branches with two input data sources. Therefore, we experimented with multimodal with a late fusion (concatenate layer) setup to evaluate the efficiency and compatibility of the model, especially on secondary data (flow data).

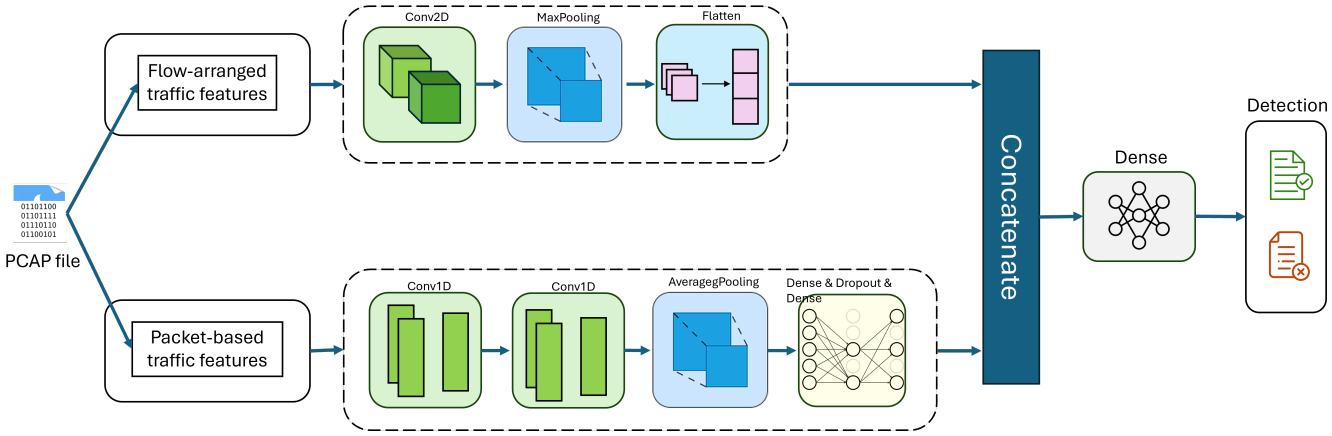

Figure 3. Overview of the deep learning model

Flow-based Branch. This branch extracts traffic features organized by flow, where each flow is identified by a 5-tuple: *source IP address*, *source TCP/UDP port*, *destination IP address*, *destination TCP/UDP port*, and *IP protocol*. Traffic data is processed using the LUCID tool introduced in [14]. Each flow is represented as a two-dimensional array of size n × f (10 × 11), as illustrated in Figure 4. Each row corresponds to a packet, described by 11 features defined in the LUCID framework: *Time*, *Packet Length*, *Higher Protocol*, *IP Flags*, *Protocols*, *TCP Length*, *TCP Ack*, *TCP Flags*, *TCP Window Size*, *UDP Length*, and *ICMP Type*.

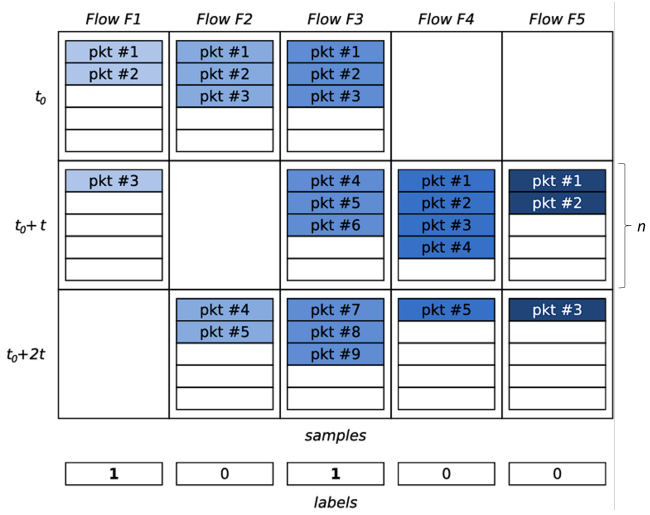

Figure 4. Graphical representation of flow-based data [14]

Flows with fewer than n packets are zero-padded. Packets are ordered chronologically, and the timestamp represents the time offset from the first packet in the array. Since packet-level attributes are extracted using TShark[2], high-level features such as the highest protocol detected and the protocol stack can be incorporated.

[2]https://www.wireshark.org/docs/man-pages/tshark.html

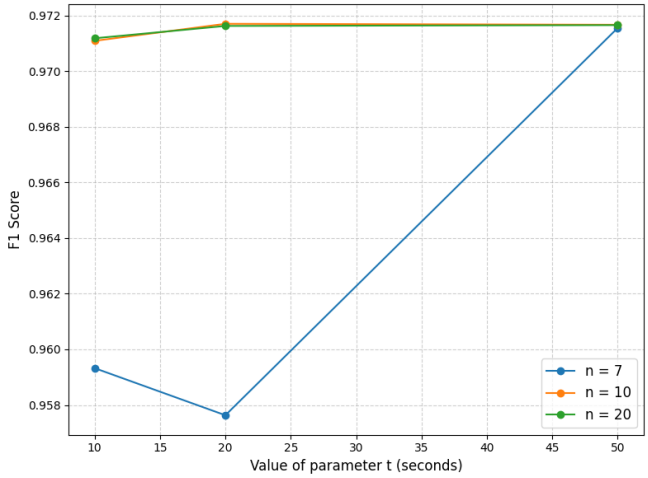

Figure 5. Sensitivity of flow-based model to time and n

The LUCID parser segments each flow into smaller subsets of packets, generating samples suitable for real-time detection scenarios where algorithms operate on flow fragments within predefined time windows. While shorter windows enable faster detection, they may also reduce classification accuracy due to increased segmentation. Following the setup in [15], a 10-second time window is applied to both benign and attack traffic. Figure 5 illustrates the experimental results of evaluating the F1-score under different configurations of window size (t - in seconds) and the number of packets (n) used to generate data for the Flow-based branch. Accordingly, when t = 10 and n = 20, the F1-score reaches 0.971, which is comparable to configurations with larger window sizes. However, as the window size increases, the number of packets contained in each flow becomes inversely proportional, resulting in a larger padding portion.

Consequently, larger window sizes lead to reduced model effectiveness, accompanied by a degradation in overall system performance.

To reduce model complexity and execution time for deployment on resource-constrained devices, a lightweight detection system is developed using a CNN model. The CNN architecture leverages parameter sharing by reusing kernel weights, unlike traditional fully connected networks where each weight is utilized only once. This design significantly lowers storage and memory demands, enhancing the model's suitability for edge deployment within IoT environments. Table 3 illustrates the summary of deep learning model for flow-based data.

Table 3. The deep learning model for flow-based data

Layers	Output	Parameter
Conv2D	(None, 8, 1, 64)	2176
Activation	(None, 8, 1, 64)	0
GlobalMaxPooling2D	(None, 64)	0
Flatten	(None, 64)	0
Dense	(None, 1)	65

Packet-based Branch. The packet-based data processing starts with feature extraction by Tshark, followed by data grouping, duplicate and missing data removing. Then additional stream features, such as *frame.time*, *ip.src_host*, *ip.dst_host*, *arp.src.proto_ipv4*, *arp.dst.proto_ipv4*, *http.file_data*, *http.request.full_uri*, *mqtt.msg*, *http.request.uri.query*, *tcp.options*, *udp.port*, *tcp.payload*, *tcp.srcport*, *tcp.dstport*, and *icmp.transmit_timestamp* are removed. Next, the data is encoded by using dummy encoding and Z-score normalization which defined by $\{x - \mu\}/\sigma$, where x is the feature value, μ is the mean value and σ is the standard deviation.

Table 4. The deep learning model for packet-based data

Layers	Output	Parameters
Conv1D	(None, 48, 74)	444
Conv1D	(None, 44, 50)	18550
GlobalAveragePooling1D	(None, 50)	0
Dense	(None, 60)	3060
Dropout	(None, 60)	0
Dense	(None, 60)	3660
Dropout	(None, 60)	0
Dense	(None, 1)	61

To address the diversity and complexity of DDoS attack patterns in IoT environments, CNN model is employed as a suitable deep learning approach. The proposed model, designed to capture the distinct characteristics of DDoS traffic, consists of 25,775 trainable parameters, as outlined in Table 4. The architecture includes a one-dimensional convolutional layer (Conv1D) for feature extraction, followed by fully connected (Dense) layers for classification. Additionally, the model integrates GlobalAveragePooling1D and Dropout layers to enhance feature robustness, improve generalization, and mitigate overfitting.

The final fully connected layer will make a decision between 0 and 1 with 0 for benign sample and 1 for attack sample.

4. Experiment and Result Analysis

4.1. Datasets and Data Processing

In this study, we used Edge-IIoTset and CiCIoMT2024 as two major datasets for the experimental process.

Edge-IIoTset. The Edge-IIoT dataset [16] is designed to facilitate intrusion detection research, enabling the evaluation of intrusion detection systems using federated deep learning and centralized learning methods. Table 5 lists the attack and benign labels in the dataset. It includes 14 types of attacks related to IoT and IIoT protocols, categorized into five threat types: information gathering, DoS and DDoS attacks, injection attacks, malware-based attacks, and man-in-the-middle attacks. The dataset contains 1,176 attributes, 61 of which are highly correlated. It consists of 20,952,648 samples, including 11,223,940 benign samples and 9,728,708 attack-related samples.

Table 5. The Edge-IIoTset Dataset Distribution

Class	Attack	Count	Total
Benign	-	11223940	11223940
Attack	Backdoor	24862	9728708
	DDoS_HTTP	249022	
	DDoS_ICMP	2914354	
	DDoS_TCP	2200120	
	DDoS_UDP	3201626	
	Fingerprinting	1007	
	MITM	1229	
	Password	1053385	
	Port Scanning	225643	
	Ransomware	10925	
	SQL injection	51203	
	Uploading	37634	
	Vul scanner	145869	
	XSS	15915	
Total		20952648	20952648

CiCIoMT2024. CiCIoMT2024 dataset [17] was created by the Canadian Cybersecurity Institute and provides a real-world benchmark dataset for developing and evaluating IoMT security solutions. Eighteen cyberattacks were conducted on a testbed of 40 IoMT devices (25 physical and 15 simulated), utilizing widely recognized

protocols such as Wi-Fi, MQTT, and Bluetooth. The attacks were categorized into five types: DDoS, DoS, Reconnaissance, MQTT-based attacks, and Spoofing. This dataset establishes a baseline benchmark that complements existing research, supporting researchers in developing advanced security solutions for healthcare systems using machine learning. The statistics for this dataset are described in Table 6.

Table 6. The CiCIoMT2024 Dataset Distribution

Class	Category	Attack	Count
Benign	–	–	230,339
Attack	Spoofing	ARP spoofing	17,791
	Recon	Ping sweep	926
		Recon VulScan	3,207
		OS scan	20,666
		Port scan	106,603
	MQTT	Malformed data	6,877
		DoS connect flood	15,904
		DDoS publish flood	36,039
		DoS publish flood	52,881
		DDoS connect flood	214,952
	DoS	DoS TCP	462,480
		DoS ICMP	514,724
		DoS SYN	540,498
		DoS UDP	704,503
	DDoS	DDoS SYN	974,359
		DDoS TCP	987,063
		DDoS ICMP	1,887,175
		DDoS UDP	1,998,026

Data Processing. This study focuses on DDoS attack detection. Therefore, we filter DDoS-related and benign samples from the two datasets mentioned above for preprocessing before training the proposed system. Table 7 presents the number of data samples after filtering. Specifically, DDoS HTTP Flood, DDoS TCP SYN Flood, DDoS ICMP Flood, and DDoS UDP Flood samples are extracted from the Edge-IIoTset dataset, while MQTT DDoS Publish Flood and MQTT DDoS Connect Flood samples are taken from the CiCIoMT2024 dataset. Additionally, benign samples are selected from both datasets.

4.2. Evaluation Metrics

We use four common metrics to evaluate the machine learning architectures during training: accuracy, precision, recall, and F1-score. Table 8 presents the evaluation criteria for the DDoS attack detection models in this study. These metrics are derived from four types of values in the confusion matrix:

- True Positive (TP): The number of correctly predicted attack instances.
- False Positive (FP): The number of instances incorrectly classified as attacks.
- True Negative (TN): The number of correctly predicted non-attack instances.
- False Negative (FN): The number of attack instances incorrectly classified as non-attacks.

4.3. Experiment Scenarios

Table 7 presents the distribution of labels used in the experiment. The labels *DDoS HTTP Flood*, *DDoS TCP SYN Flood*, *DDoS ICMP Flood*, and *DDoS UDP Flood* are derived from the **Edge-IIoTset** dataset, while the remaining labels are sourced from the **CiCIoMT2024** dataset. Notably, the benign label contains samples from both datasets. For experimental purposes, both datasets were partitioned into smaller subsets, each containing a single type of DDoS attack. The convergence behavior of the proposed multimodal deep learning model was evaluated under various aggregation functions within the Swarm Learning framework. The evaluation was conducted under a worst-case scenario, where a union of at least three and at most five clients was formed, with a one-to-one mapping between clients and DDoS attack types. Additionally, a separate experiment was conducted in which three clients were assigned two attack types, each in their respective local datasets. These experimental setups introduced data imbalance and ensured non-Independent and Identically Distributed (non-IID) data across clients.

We conduct three scenarios to evaluate the training results on a distributed environment compared to a centralized environment:

- Scenario 1: 3 hosts, each host contains only one attack label of the Edge-IIoTset dataset
- Scenario 2: 3 hosts, each host contains two attack labels from both Edge-IIoTset and CiCIoMT2024 dataset.
- Scenario 3: 5 hosts, each host contains one attack labels from both Edge-IIoTset and CiCIoMT2024 dataset

Table 9, 10, and 11 describe the data distribution for three above scenarios.

4.4. Result Analysis

Scenario 1. Figure 6 illustrates the average accuracy per epoch of the proposed multimodal deep learning model when trained with three different aggregation

Table 7. The statistics of experimental data

Label	Train	Val	Test	Total
DDoS HTTP Flood (1)	39161	4481	4902	48544
DDoS TCP Syn Flood (1)	40459	4499	5104	50062
MQTT DDoS Publish Flood (2)	47026	9157	9652	65835
DDoS ICMP Flood (1)	55021	6104	6814	67939
DDoS UDP Flood (1)	98433	10675	12459	121567
MQTT DDoS Connect Flood (2)	117776	13581	15809	147166
Benign (1, 2)	396999	39822	43392	480213
Total	**794875**	**88319**	**98132**	**981326**

(1) - Edge-IIoTset dataset (2) - CiCIoMT2024 dataset

Table 8. Metric Performance

Metric	Formula	Definition
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Proportion of correctly classified traffic, considering both normal and attack traffic.
Precision	$\frac{TP}{TP+FP}$	Proportion of predicted normal traffic that are actually normal, measuring model reliability for normal predictions.
Recall	$\frac{TP}{TP+FN}$	Proportion of actual normal traffic correctly identified, also known as sensitivity or true positive rate.
F1-score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$	Harmonic mean of precision and recall, providing a balance between them.

Table 9. Data distribution in Scenario 1

	Lables	Train	Val	Test	Sum
Host 1	HTTP, ICMP, benign	187368	20818	23132	231318
Host 2	TCP Syn Flood	81691	9077	10085	100853
Host 3	UDP Flood	196618	21846	24274	242738
Sum		465677	51741	57491	574909

Table 10. Data distribution in Scenario 2

	Labels	Train	Val	Test	Sum
Host 1	HTTP, ICMP, benign	187368	20818	23132	231318
Host 2	TCP Syn Flood, UDP Flood, benign	278309	30923	34359	343591
Host 3	MQTT Publish Flood, MQTT Connect Flood, benign	329198	36578	40641	406417
Sum		794875	88319	98132	981326

Table 11. Data distribution in Scenario 3

	Labels	Train	Val	Test	Sum
Host 1	HTTP, ICMP, benign	187368	20818	23132	231318
Host 2	TCP Syn Flood, benign	81691	9077	10085	100853
Host 3	UDP Flood, benign	196618	21846	24274	242738
Host 4	MQTT Publish Flood, benign	106663	11852	13168	131683
Host 5	MQTT Connect Flood, benign	222535	24726	27473	274734
Sum		794875	88319	98132	981326

methods under Scenario 1. As observed, both the geomedian and coordmedian methods exhibit similar performance, outperforming the mean method in early epochs—96.98% vs. 96.83% in epoch 2 and 99.29% vs. 99.14% in epoch 3. However, after 10 epochs,

Table 12. Model Training Parameters

Parameter	Value
Learning Rate	0.001
Batch Size	128
Number of Epochs	10
Optimizer	Adam
Loss Function	binary_crossentropy
No. of System Frequencies	1024
No. of Nodes	3–5

the mean method achieves the highest accuracy at 99.87%, compared to 99.55% for both geomedian and coordmedian.

Figure 7 presents the average loss per epoch under the same experimental conditions. The loss curves corroborate the accuracy trends, showing comparable convergence behavior across the three aggregation methods. The results are similar in Figure 8; the model also achieved convergence on the validation dataset for all three merge methods during the experiment in scenario 1. In particular, the mean performed slightly better than the other two methods.

Figure 9 illustrates the average validation accuracy of the three merging methods under Scenario 1. As observed, all methods show rapid performance improvement during the early epochs. At epoch 1, the validation accuracy is approximately 98.86%, 98.83%,

Figure 6. Average Training Accuracy – Scenario 1

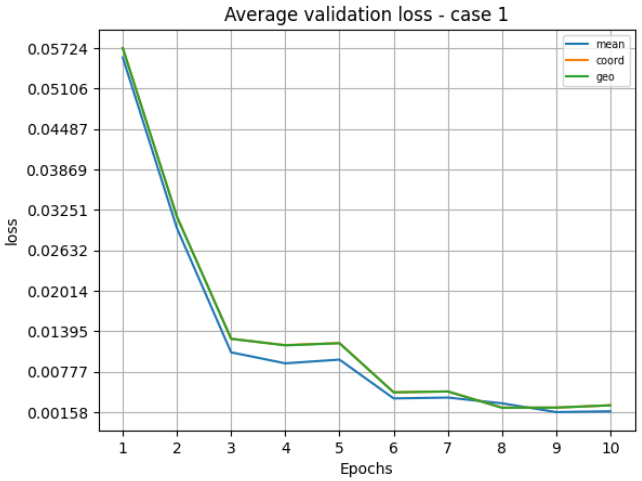

Figure 8. Average Validation Loss – Scenario 1

and 98.85% for the mean, coord, and geo methods, respectively. By epoch 3, the mean method slightly outperforms the other two, reaching approximately 99.67%, compared to 99.60% for coord and 99.59% for geo. This trend continues through the mid-training phase.

After 10 epochs, all three methods converge to nearly identical validation accuracy, reaching approximately 99.80%, indicating comparable final generalization performance. Overall, while the mean merging method demonstrates a marginal advantage in early convergence, all three approaches achieve similar performance at convergence under Scenario 1.

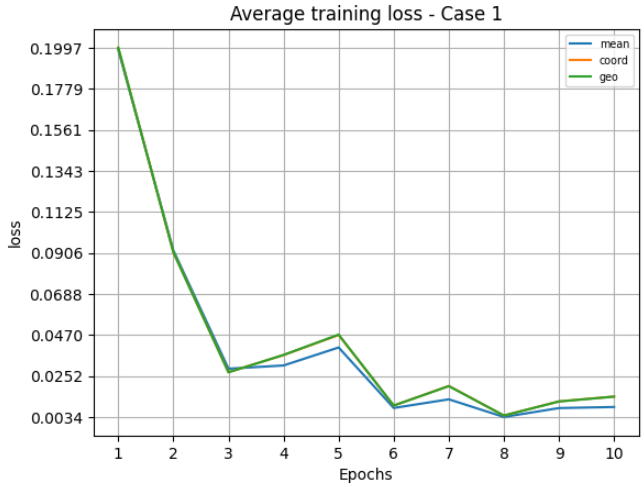

Figure 7. Average Training Loss – Scenario 1

Figure 10 displays the confusion matrices for binary attack detection. The results indicate high and consistent classification accuracy, with most predictions concentrated along the main diagonal.

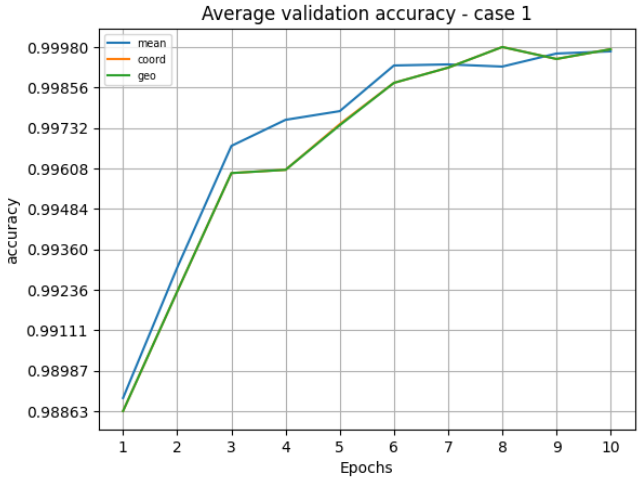

Figure 9. Average Validation Accuracy – Scenario 1

This demonstrates the model's ability to correctly distinguish between attack and benign traffic.

Table 13 compares the effectiveness of the proposed multimodal deep learning model against each individual sub-branch and the CL baseline. Under the CL setting, the multimodal model achieves perfect scores across all four evaluation metrics (100%), along with the single-modal model based on packet-level features. Both outperform the flow-based model and previously reported results in [18], confirming the superiority of the multimodal architecture in capturing both packet and flow characteristics.

Furthermore, when integrated into the Swarm Learning framework, the performance of the multimodal model remains robust. Across all aggregation methods, the maximum performance drop compared to the centralized setting is within 0.5%, demonstrating

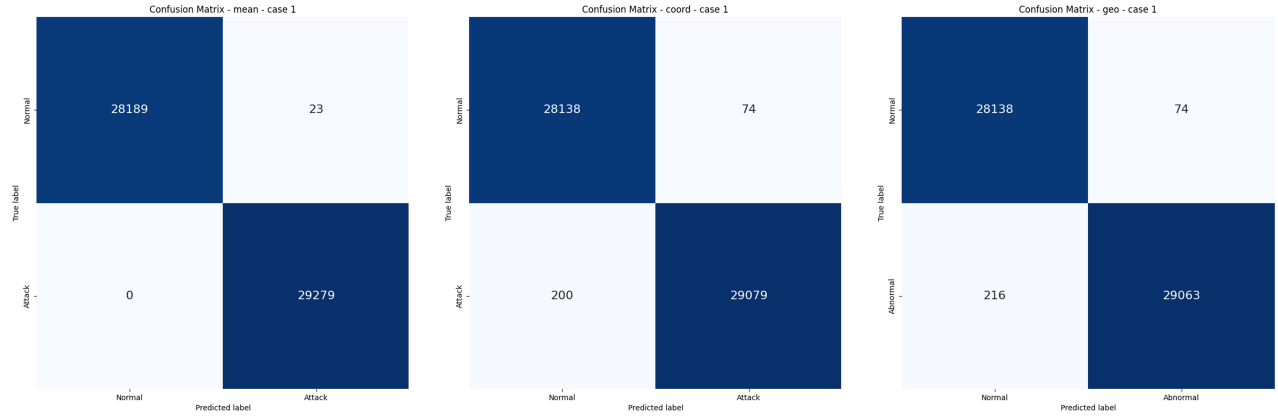

Figure 10. Confusion matrix of three merge methods – Scenario 1

Table 13. Experimental results – Scenario 1

	Approach	Accuracy	Precision	Recall	F1_score
Taraf [18]	**CL / Packet**	99.55	99.50	99.50	99.50
Centralized Learning	**Packet**	**100.00**	**100.00**	**100.00**	**100.00**
	Flow	96.85	94.28	99.88	97.00
	Multimodal	**100.00**	**100.00**	**100.00**	**100.00**
SL - Packet [8]	**Mean**	**100.00**	**100.00**	**100.00**	**100.00**
	Coord	**100.00**	99.99	**100.00**	99.99
	Geo	**100.00**	**100.00**	**100.00**	**100.00**
SL - Flow	**Mean**	96.59	93.96	99.70	96.75
	Coord	96.27	93.94	99.06	96.43
	Geo	96.27	93.94	99.06	96.43
SL - Multimodal	**Mean**	99.96	100.00	99.92	99.96
	Coord	99.52	99.32	99.75	99.53
	Geo	99.50	99.26	99.75	99.50

Swarm Learning (SL), Centralized Learning (CL), coord (coordmedian), geo (geomedian)

the model's stability and effectiveness in decentralized learning environments.

Scenario 2. Figures 11 and 12 show the trend of accuracy and loss of the model on the validation dataset in this scenario. Unlike Scenario 1, where performance was tracked per epoch, Scenarios 2 and 3 adopt round-based evaluation due to the increased complexity introduced by training on two distinct datasets. In such settings, per-epoch metrics do not adequately capture the convergence behavior of the aggregation functions.

Overall, the mean aggregation method demonstrates superior performance compared to the coordmedian and geomedian methods. In terms of accuracy, all three methods begin with a similar starting loss value of 0.5402. After 27 rounds of aggregation, the mean method reaches an accuracy of approximately 0.98, while the coordmedian and geomedian methods converge more slowly, reaching only around 0.85.

Regarding validation loss, the mean method consistently achieves the lowest values across all aggregation rounds. It exhibits a rapid decline in the initial rounds and continues to decrease steadily, ultimately reaching

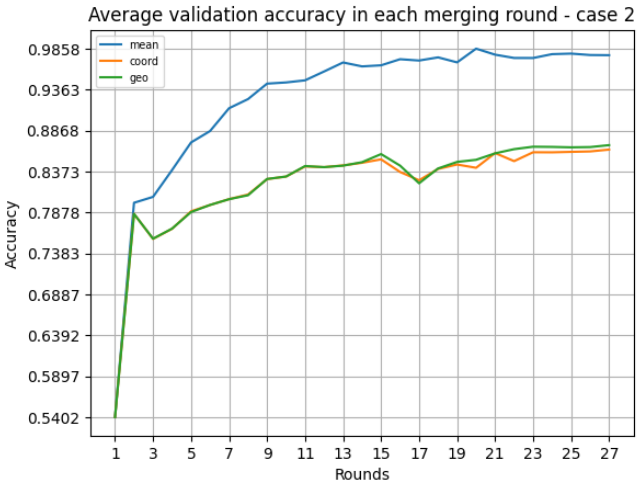

Figure 11. Average Validation Accuracy – Scenario 2

a final loss of 0.05 from an initial value of 0.6949. This indicates effective and stable convergence. In contrast, the coordmedian method yields higher validation loss, stabilizing early and exhibiting minor fluctuations in later rounds, reflecting limited improvement. The geo-median method follows a similar trend with slightly better stability but still lags behind the mean method.

The performance gap between the mean method and the other two is further underscored by the higher variability observed in the coordmedian and geomedian methods, with fluctuations exceeding 5% in some rounds. These results confirm the superior convergence behavior and robustness of the mean aggregation strategy in complex, decentralized learning scenarios.

Figure 13 presents the confusion matrices of the ensemble aggregation methods within the Swarm Learning framework under Scenario 2. In general,

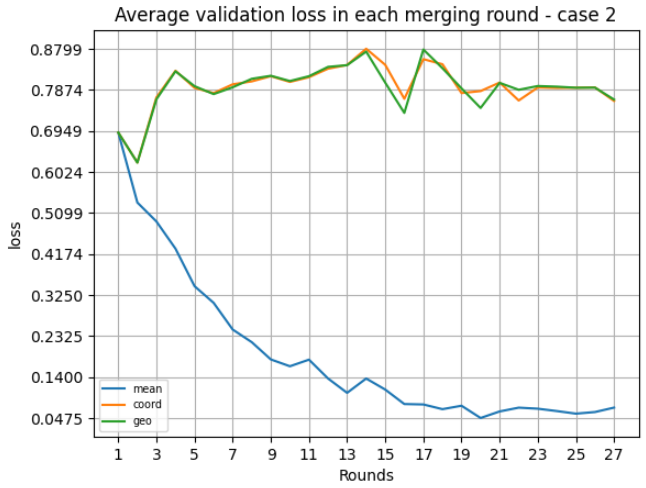

Figure 12. Average Validation Loss - Scenario 2

most values are concentrated along the main diagonal, indicating strong classification performance across methods. The mean aggregation method yields the highest classification accuracy, with minimal false positives and false negatives, resulting in an overall accuracy of approximately 98.6%. In contrast, the coordmedian and geomedian methods demonstrate reduced performance, exhibiting a greater number of misclassifications—particularly for benign samples. Among the two, the geomedian method performs slightly better, achieving a more balanced distribution between true positives and false negatives.

Scenario 2 poses a greater challenge for the proposed model due to the combination of two datasets—Edge-IIoTset and CiCIoMT2024. As summarized in Table 14, the proposed multimodal deep learning model consistently outperforms individual single-modal models and maintains strong performance within the SL framework. Under the CL setting, the multimodal model achieves perfect scores across all four evaluation metrics (100%), along with the single-modal packet-based model. Both outperform the flow-based model, reaffirming the superior performance of packet-level features in DDoS detection.

Despite being trained on two heterogeneous datasets, the flow-based model experiences only a marginal performance drop—less than 4%—compared to its counterpart trained solely on the Edge-IIoTset in Scenario 1. This result highlights the generalizability of both the multimodal and single-modal models in centralized training settings.

When deployed within the SL framework, the ensemble functions show varying performance levels. The mean aggregation method remains the most robust, with performance reductions of less than 1.4% across all evaluation metrics compared to the centralized

Table 14. Experimental results - Scenario 2

	Approach	Accuracy	Precision	Recall	F1_score
Centralized Learning	Payload	100	100	100	100
	flow	95.46	95.78	96.10	95.94
	Multimodal	100	100	100	100
Case 2 - SL - Multimodal	Mean	98.62	98.81	98.71	98.76
	Coord	83.26	84.83	85.24	85.03
	Geo	83.91	84.87	86.59	85.72

Swarm Learning (SL), Centralized Learning (CL), coord (coordmedian), geo (geomedian)

baseline. Conversely, the coordmedian and geomedian methods experience more significant drops in accuracy. These results confirm that the mean aggregation method provides the most stable and effective performance for the proposed multimodal deep learning model in decentralized learning scenarios.

Scenario 3. Figures 14 and 15 illustrate the average accuracy and loss of the proposed model across the three ensemble methods under Scenario 3. Among these, the mean aggregation method consistently outperforms the coordmedian and geomedian methods. Initially, all methods begin with an accuracy of 0.5624. In the first three ensemble rounds, coordmedian and geomedian slightly outperform mean (e.g., coordmedian reaches approximately 0.70, while mean attains 0.69 in round 2 and 0.6858 in round 3). However, in subsequent rounds, the accuracy of the mean method steadily improves and surpasses the others, demonstrating better convergence behavior. The coordmedian and geomedian methods converge earlier but stabilize at lower accuracy levels. All three methods show an upward trend in accuracy across pooling rounds, confirming the effectiveness of the training process.

Regarding validation loss, all methods start at 0.683. After 18 rounds, the mean method reduces its loss to 0.2, whereas coordmedian and geomedian only reach 0.5. The mean method consistently achieves the lowest loss across all rounds, with a nearly linear decrease, indicating a more efficient optimization process. In contrast, coordmedian and geomedian exhibit higher losses and greater fluctuations, stabilizing at suboptimal values.

Figure 16 presents the confusion matrices of the three ensemble methods under Scenario 3. The mean-based model yields the highest overall performance, with lower false positive and false negative rates. The coordmedian method performs the worst in all key metrics, while the geomedian method performs better than coordmedian but does not surpass the mean method. From the perspective of minimizing false positives and false negatives, which is critical in intrusion detection, the mean method is preferable.

The experimental results in Scenario 3 are described in Table 15. Under the SL framework in Scenario 3, the unimodal packet-based model achieves the highest performance, followed by the proposed multimodal

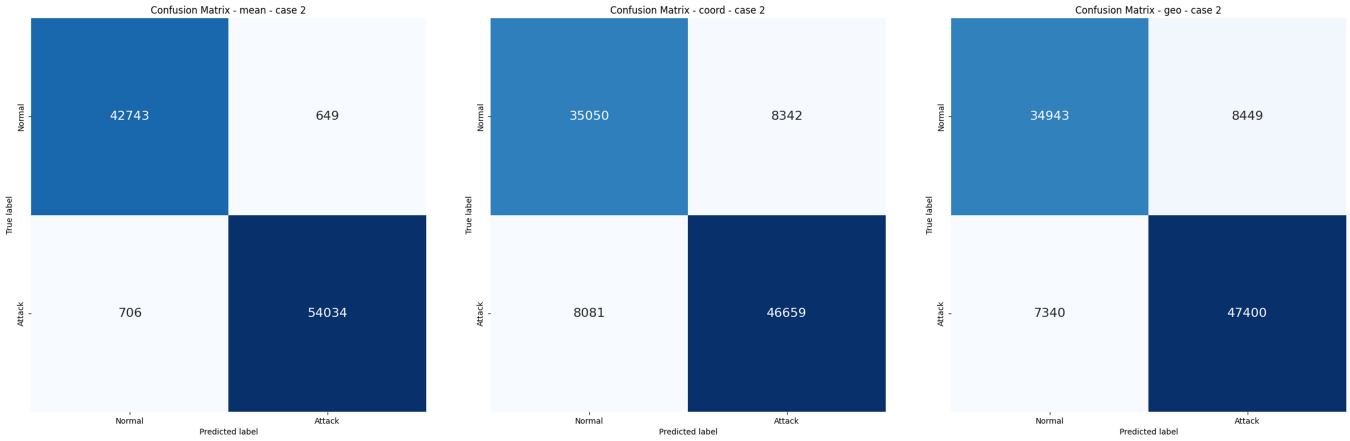

Figure 13. Confusion matrix of three merge methods – Scenario 2

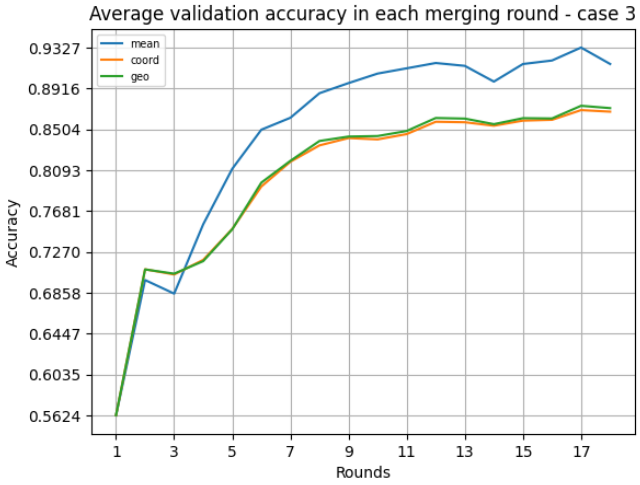

Figure 14. Average Validation Accuracy – Scenario 3

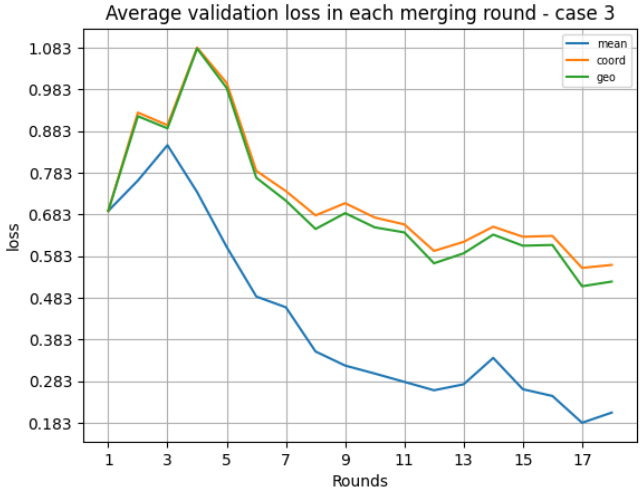

Figure 15. Average Validation Loss – Scenario 3

model, while the flow-based unimodal model performs the worst. These results affirm the robustness of the proposed multimodal architecture across both packet and flow contexts. For the packet-based models, performance degradation across ensemble methods remains below 10%. The mean aggregation method delivers the best results, with accuracy of 97.17%, precision of 99.82%, recall of 95.09%, and F1-score of 97.40%. Conversely, the geomedian method yields the lowest performance in this group (accuracy 95.00%, precision 99.02%, recall 91.95%, and F1-score 95.35%).

Table 15. Experimental results – Scenario 3

	Approach	Accuracy	Precision	Recall	F1_score
CL	**Packet**	**100.00**	**100.00**	**100.00**	**100.00**
	Flow	95.46	95.78	96.10	95.94
	Multimodal	**100.00**	**100.00**	**100.00**	**100.00**
Multimodal - Case 2 - SL	**Mean**	98.62	98.81	98.71	98.76
	Coord	83.26	84.83	85.24	85.03
	Geo	83.91	84.87	86.59	85.72
Packet - Case 3 - SL	**Mean**	97.17	99.82	95.09	97.40
	Coord	94.05	99.03	90.22	94.42
	Geo	95.00	99.02	91.95	95.35
Flow - Case 3 - SL	**Mean**	70.70	87.93	55.02	67.69
	Coord	73.65	81.87	67.76	74.15
	Geo	75.01	82.38	70.22	75.82
Multimodal - Case 3 - SL	**Mean**	92.62	99.78	86.95	92.93
	Coord	89.20	96.58	83.60	89.62
	Geo	89.69	96.88	84.24	90.12

Swarm Learning (SL), Centralized Learning (CL), coord (coordmedian), geo (geomedian)

Table 16. Average test loss in all scenarios

Method	Case 1	Case 2	Case 3
Mean	0.0039	0.0352	0.1968
Coord	0.0181	0.7622	0.5151
Geo	0.0177	0.7655	0.4760

Table 16 reports the average test loss of the three aggregation methods across all experimental scenarios. Overall, the mean aggregation method consistently

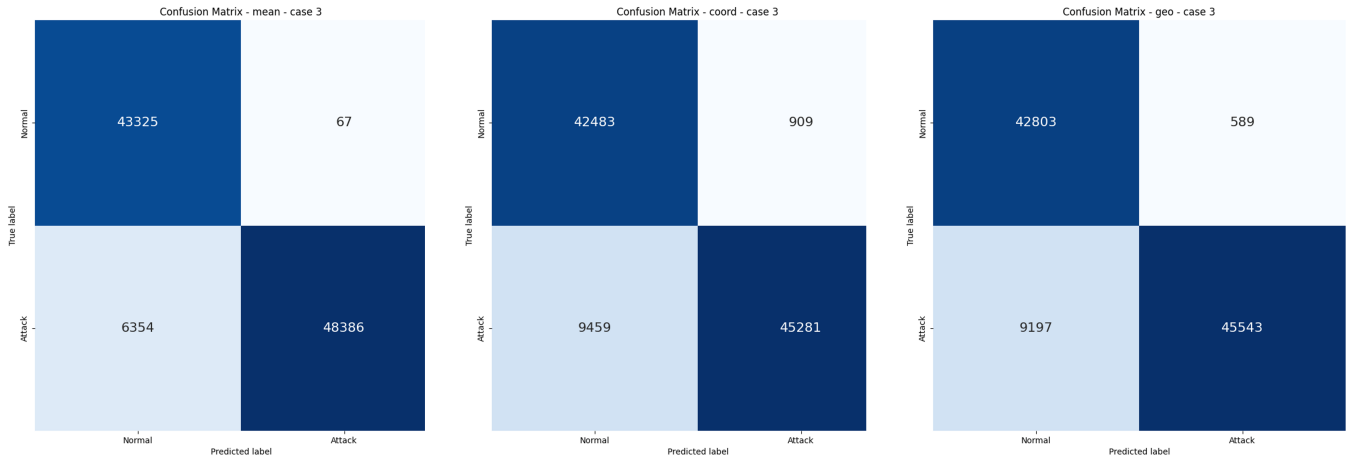

Figure 16. Confusion matrix of three merge methods – Scenario 3

achieves the lowest test loss, demonstrating superior generalization performance under diverse scenarios.

The flow-based models in Scenario 3 exhibit significant performance degradation compared to their counterparts in Scenarios 2 and 1, and when benchmarked against models trained using the CL approach. The highest-performing ensemble method in this setting is geomedian, with accuracy of 75.01%, precision of 82.38%, recall of 70.22%, and F1-score of 75.82%. However, this remains substantially lower than the corresponding CL-based flow model, which achieves 95.46%, 95.78%, 96.10%, and 95.94% for the respective metrics.

Notably, the multimodal model with the mean aggregation function achieves the highest precision (99.78%) among all SL-based models in Scenario 3, closely followed by the unimodal packet-based model using mean (98.82%). These results are comparable to the CL-based packet model, which records a precision of 99.82%, and approach the ideal (100%). This underscores the effectiveness of the mean aggregation method in minimizing false positives—an essential factor in operational environments where incorrectly blocking legitimate traffic can negatively impact users and services.

Overall, in both Scenarios 2 and 3—when trained on the Edge-IIoTset and CiCIoMT2024 datasets—the mean aggregation method delivers the most stable and accurate performance among the three evaluated merge methods.

Compute Performance Analysis. Compared to the CL approach, the Swarm Learning framework demonstrates significantly lower bandwidth consumption. This efficiency arises because nodes in SL exchange only model parameters rather than transferring raw training data to a central server. As detailed in Section 3.2, the proposed multimodal model has an approximate size of 0.611 MB. In the most bandwidth-intensive setting—Scenario 3, with five participating nodes and 18 training rounds—four nodes transmit their models to a designated leader during each round. This results in a total network load of approximately $4 \times 0.611 \times 18 = 43.992$ MB. This volume is considerably smaller than the data transfer required in CL, where several hundred megabytes of raw training data must be sent to a central node. The results highlight SL's communication efficiency, making it more suitable for distributed environments with limited bandwidth, such as edge computing or IoT systems.

5. Conclusion

This study proposes a multimodal distributed deep learning model for DDoS detection in IoT systems built upon the Swarm Learning framework. The architecture consists of two unimodal branches: one based on packet-level traffic features and the other on flow-level features. The experimental results demonstrate that SL is a promising framework for deploying deep learning models, particularly in multimodal settings. Across all evaluated scenarios, the multimodal and unimodal proposed models achieve competitive performance. The packet-based branch consistently yields high accuracy, while the multimodal model approaches the performance of its centralized learning counterpart. Among the evaluated aggregation strategies, the mean method exhibits the most stable and reliable performance across all configurations.

Future research will focus on extending the multimodal deep learning approach within the Swarm Learning framework by incorporating additional deep learning architectures. Moreover, developing vertically collaborative or hybrid (vertical and horizontal) multimodal learning models is identified as a promising

direction to further enhance learning effectiveness in decentralized environments.

Acknowledgements. This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number C2024-26-11

References

[1] Leonel Santos, Ramiro Gonc¸alves, Carlos Rabadao, and José Martins. (2023) *A flow-based intrusion detection framework for internet of things networks.* In: Cluster Computing (2023), pp. 1–21.

[2] Roberto Doriguzzi-Corin, Stuart Millar, Sandra Scott-Hayward, Jesus Martinez-del-Rincon, and Domenico Siracusa. (2020) *LUCID: A practical, lightweight deep learning solution for DDoS attack detection.* In: IEEE Transactions on Network and Service Management 17.2 (2020), pp. 876–889.

[3] Xuan-Ha Nguyen, Xuan-Duong Nguyen, Hoang-Hai Huynh, and Kim-Hung Le. *Realguard: A lightweight network intrusion detection system for IoT gateways.* In: Sensors 22.2 (2022), p. 432.

[4] Aklil Kiflay, Athanasios Tsokanos, Mahmood Fazlali, (2024) and Raimund Kirner. *Network intrusion detection leveraging multimodal features.* In: Array 22 (2024), p. 100349.

[5] Rashid, M. M., Khan, S. U., Eusufzai, F., Redwan, M. A., Sabuj, S. R., and Elsharief, M. (2023). *A federated learning-based approach for improving intrusion detection in industrial internet of things networks.* Network, 3(1), 158-179.

[6] Li, Y., Chen, C., Liu, N., Huang, H., Zheng, Z., and Yan, Q. (2020). *A blockchain-based decentralized federated learning framework with committee consensus.* IEEE Network, 35(1), 234-241.

[7] Warnat-Herresthal, S., Schultze, H., Shastry, K. L., Manamohan, S., Mukherjee, S., Garg, V., ... and Schultze, J. L. (2021). *Swarm learning for decentralized and confidential clinical machine learning.* Nature, 594(7862), 265-270.

[8] Abdulrahman A. Alshdadi, Abdulwahab Ali Almazroi, Eesa Alsolami, and Nasir Ayub., *Enhanced IoT Security for DDOS Attack Detection: Split Attention-Based ResNeXt-GRU Ensembler Approach.* In in IEEE Access, vol. 12 (2024), pp. 112368-112380.

[9] Yawar Abbas Abid, Jinsong Wu, Guangquan Xu, Shihui Fu, and Nasir Ayub., *Multilevel Deep Neural Network Approach for Enhanced Distributed Denial-of-Service Attack Detection and Classification in Software-Defined Internet of Things Networks.* in IEEE Internet of Things Journal, vol. 11, no. 14 (2024), pp. 24715-24725.

[10] Liyuan Chang, and Bin Cao., *Toward Efficient Network Traffic Classifications via Multimodal Learning.* in IEEE Internet of Things Journal, vol. 12, no. 24 (2024), pp. 51812-51820.

[11] Makhduma F. Saiyed, and Irfan Al-Anbagi., *A Genetic Algorithm- and t-Test-Based System for DDoS Attack Detection in IoT Networks.* in IEEE Internet of Things Journal, in IEEE Access, vol. 12 (2024), pp. 25623-25641.

[12] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). *Federated learning: Strategies for improving communication efficiency.* arXiv preprint arXiv:1610.05492.

[13] Giri Sai Ram Ragam, Swarm Learning Team. (2024) *Whitepaper on merge methods in swarm learning.* Hewlett Packard Enterprise. https://github.com/HewlettPackard/swarm-learning/blob/master/docs/HPE_Merge_Methods_Whitepaper.pdf. Accessed: 2025-04-25

[14] Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martinez-del-Rincon, J., Siracusa, D. (2020) *LUCID: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection.* IEEE Transactions on Network and Service Management. Advance online publication. https://doi.org/10.1109/TNSM.2020.2971776.

[15] Doriguzzi-Corin, R., and Siracusa, D. (2024) *FLAD: adaptive federated learning for DDoS attack detection.* Computers & Security, 137, 103597.

[16] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., and 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.

[17] Dadkhah, S., Neto, E. C. P., Ferreira, R., Molokwu, R. C., Sadeghi, S., and Ghorbani, A. A. (2024) *CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT.* Internet of Things, 28, 101351.

[18] Al Nuaimi, T., Al Zaabi, S., Alyilieli, M., AlMaskari, M., Alblooshi, S., Alhabsi, F., ... & Al Badawi, A. (2023) *A comparative evaluation of intrusion detection systems on the edge-IIoT-2022 dataset.* Intelligent Systems with Applications, 20, 200298.