CFL-IDS: An Effective Clustered Federated Learning Framework for Industrial Internet of Things Intrusion Detection

Yao Shan, *Graduate Student Member, IEEE*, Yu Yao, *Member, IEEE*, Xiaoming Zhou, Tong Zhao, Bo Hu, and Lei Wang

***Abstract*—The Industrial Internet of Things (IIoT) offers the manufacturing sector opportunities for transformation and upgrade but also carries significant security risks. Traditional federated learning (FL) as a potential security solution is challenging in complicated application environments with heterogeneous data, imbalanced data, and poisoning attacks. To address these challenges, we construct a clustered FL Framework for IIoT intrusion detection (CFL-IDS) based on local models' evaluation metrics (EMs). First, we designed an intrusion detection model with a dynamic focal loss (DFL) for all edge nodes (ENs). This model's performance is enhanced under various imbalanced data partitions by dynamically altering the focus on samples during the loss minimization training process. Second, the time series of EMs of local models to reflect the data distribution of ENs implicitly, and use clustering algorithms to facilitate knowledge sharing among those ENs with similar data distribution to co-optimize a common model for them. Finally, an intelligent cooperative model aggregation mechanism (ICMAM) adaptively adjusts each local model's weight distribution, which substantially improves the benefits of FL and alleviates subpar models' alleviates interference from subpar models to FL. Experiments demonstrate that CFL-IDS has stronger robustness and displays superior performance under data imbalance and non-independent and identically distributed (non-IID) situations while being effective against poisoning attacks.**

***Index Terms*—Clustered federated learning (FL), data imbalanced, evaluation metrics (EMs), Industrial Internet of Things (IIoT) Intrusion detection, non-independent and identically distributed (non-IID), poisoning attack.**

I. Introduction

THE INDUSTRIAL Internet of Things (IIoT) turns conventional linear manufacturing into dynamic, connected intelligent manufacturing, resulting in a new operating mode for enterprises. IIoT has wide applications in smart cities, equipment operation and maintenance, intelligent plants, and other domains by integrating big data analysis, artificial intelligence, edge computing, and other technologies. However, the IIoT faces significant security and privacy concerns due to the proliferation of smart industrial devices and the enormous volume of transaction data created.

Intrusion detection systems (IDSs) have been frequently employed in IIoT as an effective security measure. Numerous deep-learning-based IDSs have demonstrated their effectiveness in reducing security risks in IIoT thanks to the robust computing capabilities of ENs [1]. Currently, most IDSs are designed under the strong assumption that the data is centralized and that a significant number of cyberattack instances can enable the establishment of intrusion detection models. However, in practice, most enterprises are reluctant to share data, particularly confidential transaction data. Additionally, acquiring high-quality and substantial training data might be challenging, making it difficult to develop the model.

As a distributed machine learning framework, federated learning (FL) enables participants to create an ideal global model without sharing local data. However, the complicated IIoT environment presents substantial hurdles to the direct application of FL. First, due to the diversity and unpredictability of cyberthreats, there is stochastic heterogeneity in the data owned by each participant, i.e., the data are non-independent and identically distributed (non-IID). Although some methods claim to alleviate the issue [2], [3], [4], [5], [6], [7], they all ignore the additional information produced by model training (such as accuracy, F1-score, etc.), frequently implicitly reflecting the data distribution. Second, different data divisions for the same participant's data exist. One example is the imbalance where benign data are significantly more prevalent than attack data. Of course, participants can also selectively use local data, for example, by building models based solely on attack instances. The data imbalance might be mitigated. Faced with this complex data partitioning situation, the cross-entropy loss [8] and focal loss [9] frequently used by conventional classification models cannot provide efficient performance. Third, the participants may suffer from poisoning attacks in FL, which could create subpar models. Therefore, building an ideal intrusion detection model for IIoT in this complex situation is tricky.

To overcome the above challenges, we proposed a clustered FL framework for IIoT intrusion detection (CFL-IDS) in this article. Expressly, we first adopt the concept of curriculum

Manuscript received 10 July 2023; revised 28 August 2023 and 22 September 2023; accepted 10 October 2023. Date of publication 13 October 2023; date of current version 7 March 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB3101700. *(Corresponding author: Yu Yao.)*

Yao Shan, Yu Yao, and Tong Zhao are with the College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China (e-mail: 1810596@stu.neu.edu.cn; yaoyu@mail.neu.edu.cn; 1810597@stu.neu.edu.cn).

Xiaoming Zhou and Lei Wang are with the Digital Work Department, State Grid Liaoning Electric Power Supply Company Ltd., Shenyang 110169, China (e-mail: zhouxm@ln.sgcc.com.cn; wl@ln.sgcc.com.cn).

Bo Hu is with the State Grid Dalian Electric Power Supply Company Ltd., Dalian 116000, China (e-mail: hubo_dl@ln.sgcc.com.cn).

Digital Object Identifier 10.1109/JIOT.2023.3324302

learning [10] to design a deep learning model with a dynamic loss function for various data partitionings of edge nodes (ENs) to identify cyberthreats. A clustered FL framework is then developed for ENs with data heterogeneity that dynamically integrates ENs with similar data distributions to obtain personalized models. The three main contributions of this work are as follows.

1) An FL framework CFL-IDS for IIoT intrusion detection is proposed. The main idea is to perform cluster estimation on the time series of evaluation metrics (EMs) generated from local training to mine the ENs' data relevance and aggregate ENs with similar data distributions to optimize a common intrusion detection model cooperatively. Experiments have verified that CFL-IDS delivers the best outcomes in non-IID situations.
2) A new loss function called dynamic focus loss (DFL) is employed to guide the creation of local intrusion detection models, which consider the various data partitioning situations at the ENs. By incorporating the idea of curriculum learning [10] to partition the training process and dynamically adjusting the model's focus on the samples during the loss minimization training process, the DFL improves the model's performance on different data partitions, demonstrating a stronger robustness.
3) We provide an intelligent collaborative model aggregation mechanism (ICMAM) that adaptively reduces the aggregation weights of those subpar models by comparing the correlation of model parameters among different ENs to maximize the benefits of model aggregation. In this way, it can effectively resist poisoning attacks and guarantee model convergence.

The remainder of this article is organized as follows. Section II discusses related work. Section III presents some preliminary knowledge for comprehending our work. Section IV elaborates on our proposed CFL-IDS framework. Section V presents an evaluation of the effectiveness of CFL-IDS. Finally, Section VI concludes this article.

II. Related Work

A. Federated Learning for Intrusion Detection

FL for IDSs has drawn more attention in recent years. For example, Li et al. [8] proposed an FL intrusion detection scheme called DeepFed, which constructed a base classifier using convolutional neural networks (CNNs) and gated recurrent units (GRUs), and used the cross-entropy loss to guide model training to identify cyberthreats. The Paillier cryptosystem was used to ensure the security and privacy of the model parameters during the FL training. Cui et al. [11] created a decentralized asynchronous FL framework based on blockchain and incorporated controlled noise into local models using generative adversarial networks to satisfy differential privacy constraints. In order to identify zero-day botnet attacks, Popoola et al. [12] developed an FL scheme that employs FedAVG [13] to update the global model and deep neural network as a base classifiers. Experimental results demonstrate that this scheme outperforms centralized, localized, and distributed DL methods. Tahir et al. [14] designed a decentralized FL framework to detect false data injection attacks in IoT energy trading systems. The scheme utilizes a hierarchical attention aggregation mechanism to construct the optimal global model and achieves excellent scores in the IEEE 39 bus system. FedAGRU [15] is an FL detection scheme for wireless edge networks that avoids unimportant model updates by applying an attention mechanism, which improves communication efficiency and effectively prevents poisoning attacks. Hamdi [16] constructed a reliable IoT network attack detection scheme using six FL schemes, such as FedAVG, FedProx [2], etc., which proposed to validate the FL framework using the client-side evaluation. The experimental results show that FL achieves a higher detection rate than individual learning. Fed-ANIDS [17] combines autoencoders and FedProx to achieve intrusion detection of network traffic, and experimental results on three data sets show that the autoencoder-based model outperforms the generative adversarial network model.

B. Federated Learning for Non-IID

Non-IID is a significant issue in IIoT and one of the critical factors impacting the effectiveness of FL models, which has drawn considerable attention from industry and academics. Local personalization of the global model is an effective way to solve the non-IID. FedProx [2] introduces a proximal term for the local model based on the FL average algorithm FedAVG to reduce the deviation of the global model from the local model. The local training will also dynamically alter the model's number of iterative rounds to ensure FL can accommodate system heterogeneity. An analogous concept can be found in SCAFFOLD [3]. However, the non-IID situations make building a robust global model especially challenging, which is a prerequisite for such methods.

Similar to our idea, the similarity-based approach is less preoccupied with building a global model and instead attempts to learn personalized local models by modeling client relationships. CPFL [4] is a personalized FL scheme for Industry 4.0, which adaptively adjusts the aggregation weights of each client during FL by an intelligent cognitive mechanism to reduce model drift. It is worth emphasizing that CPFL creates a personalized model for each client. The starting point of clustering FL is to locate individuals with similar data distributions among multiple clients. IFCA [5] is an iterative federated clustering algorithm. The server uses loss function minimization in each iteration to estimate the groups to which each client belongs. All clients in the same group share a set of average parameters to generate a personalized model. PFA [6] uses the sparsity of neural networks to represent clusters of clients with different data distributions and executes the FedAVG [13] algorithm in each cluster to construct a personalized model. EEFED [7] proposed an FL intrusion detection scheme with an execution–evaluation dual network capable of generating both global and local models. The execution network is responsible for the federation training process, and the evaluation network generates personalized local models by calculating the cosine similarity of the models. However, these schemes ignore the role of the additional information that comes from FL. In our work, we substitute multiple rounds of model EMs

for the model parameters to guide clustering for similar node knowledge sharing and reciprocal promotion.

C. Data Imbalance in Intrusion Detection

In the realm of intrusion detection, data imbalance is a prominent topic and one of the main causes of deep learning models' inability to predict rare classes effectively. According to the current research, solving the data imbalance problem can be considered at the data level, model level, and loss function level. At the data level, [18], [19], and [20] introduced various oversampling or undersampling approaches in IDSs to balance the data set. Bedi et al. [21] suggested an I-SiamIDS neural network structure at the model level to identify majority and minority classes through a hierarchical mechanism. The CSE-IDS [22] is a three-layer IDS that uses cost-sensitive and integrated algorithms. The first layer separates attack samples from normal samples, the second layer identifies rare attack classes, and the third layer performs multiple classifications on a few classes to complete the detection. Although all of the methods above have achieved satisfactory results, they all have the flaws of overfitting and changing the data distribution.

Focal loss [9] has been demonstrated to effectively reduce data imbalance in IDS at the level of the loss function. FL-NIDS [23] uses focal loss to reweight rare class samples throughout the training process to minimize the impact of data imbalance on detection performance. By integrating the gradient coordination mechanism and improving focal loss, CNN-IDMDI [24] decreased the error between the expected and actual values and identified cyberthreats of high-dimensional and imbalanced data. However, the overweighting of hard samples by focal loss may be counterproductive in practice, particularly when the positive and negative classes ratio is not too disparate. In this article, we proposed a DFL based on focal loss combined with the idea of curriculum learning [10] to improve the robustness of the DL model.

III. Preliminary

A. Non-IID and Imbalance Data

FL frequently encounters non-IID issues when put into reality. The non-IID in FL can be summed up as follows. For a specific EN i, its data distribution $(x, y) \sim P_i$ for all sample features x and labels y, does not represent the global data distribution, i.e., $P_i \neq P_{\text{global}}$. Specifically, we consider the following two non-IID situations.

1) *Feature Distribution Skew:* The feature distributions $P_i(x)$ of various ENs varied, but $P(y|x)$ is the same. For instance, the distribution of features for a particular attack may fluctuate depending on the attack means and targets.
2) *Label Distribution Skew:* The label distributions $P_i(y)$ of various ENs varied, but $P(x|y)$ is the same. For instance, certain ENs gather more DoS data, while others gather more information about reconnaissance attacks.

When there is a significant difference in the quantity of samples from one category to another for a specific EN, this is referred to as data imbalance. A simple example is that the normal data collected by most ENs will be much higher than the attack data. We frequently use $r = n_{\min}/n_{\max}$ to calculate the imbalance ratio for imbalanced data, where $n_{\min}$ and $n_{\max}$ are the categories with the smallest and largest sample sizes, respectively. A smaller r indicates more imbalanced data.

Non-IID and data imbalance can affect model's performance since it is difficult to fit all data distributions with a uniform global model. In our work, we consider these concerns concurrently to create an effective FL intrusion detection framework.

B. Threat Model

Cyberthreats against the IIoT and poisoning attacks against the FL framework are taken into consideration by CFL-IDS. The attacks that IIoT suffers are frequently more focused than those that affect traditional IT information networks. In addition to more general attacks (e.g., DoS attacks), attackers frequently attempt to interfere with industrial processes to change system behavior.

We concentrate on the following cyberthreats to IIoT in this study.

1) *Reconnaissance Attack:* Reconnaissance attacks usually occur at the beginning of an attack and aim to collect basic details about the system, including network architecture, device model, device storage mapping, etc.
2) *Response Injection Attack* interferes with monitoring industrial processes by fabricating response messages and forwarding them to the party making the query, giving false information about the system state.
3) *Command Injection Attack* changes the system behavior by injecting false control and configuration commands to control the device to perform its unintended operation.
4) *DoS Attack:* Attacks that repeatedly target links and system programs over a short period cause resource exhaustion, which may prevent the system from offering standard services.

Poisoning attacks aim at compromising the robustness of FL models [25] and can be mainly classified as data poisoning and model poisoning.

1) *Data Poisoning* is when a participant or outside attacker corrupts the model by contaminating the local data (e.g., label inversion or feature replacement) during the FL.
2) *Model Poisoning* is a way for attackers to impact the performance of FL models by tampering with the local model parameters and uploading them to the server.

It is crucial to stress that our goal is to lessen the impact of poisoning attacks on FL, not to detect poisoning attacks.

C. Time Series of Evaluation Metrics

The data distribution of ENs is not directly accessible due to privacy protection. As a result, we desire certain parameters that can implicitly indicate the data distribution relationship among ENs. Similar data distribution patterns on ENs will result in similar EMs for their local models. Two important things should be noted. First, the EMs for a single communication are episodic and poorly reflect the distribution of ENs' data. We can expand these EMs in terms of temporal dimension through the FL process. Each round of communication for

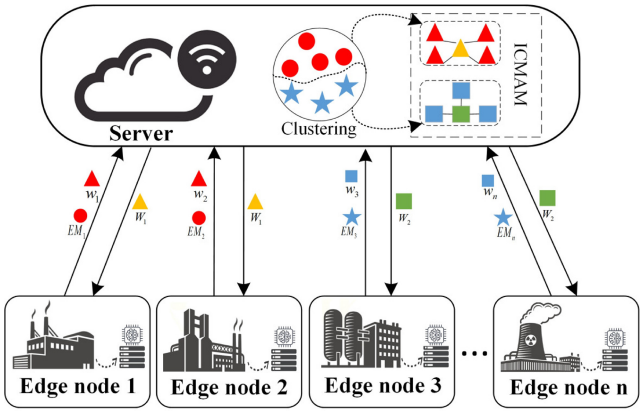


Fig. 1. CFL-IDS framework under consideration.

FL will be considered a time node. To show the distribution of their data, we shall record the EMs for several communications of ENs. The second concern is the choice of EMs. We recommend that appropriate EMs be chosen based on various tasks. Our work concentrates on IIoT intrusion detection and selects the EMs each EN should upload: *accuracy, precision, recall, F1-score, false positive rate (FPR)*, and *loss*. All EMs are defined as follows.

1) *Accuracy:* Portion of the model that correctly predicts.
2) *Precision:* Proportion of samples predicted to be attacks that also indeed attack.
3) *Recall:* Proportion of attack samples that can be accurately identified.
4) *F1-Score:* A weighted average of precision and recall is generally considered a comprehensive metric of the classifier.
5) *FPR:* Proportion of normal samples were identified as attack samples.
6) *Loss:* The loss utilized to measure the difference between predicted and true values is illustrated in (4) in this study.

IV. Proposed CFL-IDS Framework

In this section, the workflow of CFL-IDS is illustrated. The intrusion detection model, the time-series clustering of EMs, and the ICMAM are then detailed.

A. Workflow of CFL-IDS

The proposed CFL-IDS framework consists of two main types of entities, i.e., ENs and server, as shown in Fig. 1.

ENs: ENs are responsible for developing local models and identifying IIoT cyberthreats. Specifically, the ENs will train the intrusion detection models issued by the server on the local data set (network traffic collected by industrial agents) and upload the updated model parameters and EMs to the server. Deploying the trained models on each EN can complete the intrusion detection for industrial agents.

Server: The server is in charge of creating several personalized intrusion detection models. By applying time-series clustering (see below for details) on numerous rounds of EMs uploaded by ENs, ENs are organized into clusters. A personalized model that belongs to each EN cluster is created via model aggregation and sent to the appropriate ENs. It takes several rounds of communication between the server and the ENs to get an intrusion detection model with full performance.

Algorithm 1: CFL-IDS Framework Workflow

Input: ENs set $C = \{c_1, c_2, \cdots c_K\}$, Number of communication rounds R, Local training epoch E, Learning rate η, Clustering Length L.
Output: Personalized models $\{w_i^R\}_{i\in K}$.

Initialization: Initial model parameters w^0;
Producer:
for $r \le R$ **do**
  (1) For ENs:
  for $\forall i \in K$ **do**
    The EN c_i calculates the model parameters w_i^r according to w_i^{r-1} using Algorithm 2 and records EMs on the test set $P_i^r = [p_{i,Acc}^r, p_{i,Pre}^r, p_{i,Rec}^r, p_{i,F1}^r, p_{i,FNR}^r, p_{i,Loss}^r]$
  end
  Upload w_i^r and M_i^r to Server
  (2) For Server:
  Divide ENs into N clusters according to L, P and Algorithm 3.
  for $\forall n \in N$ **do**
    for $\forall n_j \in n_m$ **do**
      Calculate the average similarity $A_n = [a_{n_1}, a_{n_2}, \cdots, a_{n_m}]$ of all ENs in cluster n by Eq. (12) - Eq. (13)
    end
    Calculate the importance $\alpha_n = [\alpha_{n_1}, \alpha_{n_2} \cdots, \alpha_{n_m}]$ of all ENs in cluster n by $\alpha_{n_j}^r = a_{n_j}^r / \sum_{j=1}^m a_{n_j}^r$
    Calculate aggregation parameters for cluster n: $W_n^{r+1} = \sum_{j=1}^{m} \alpha_{n_j}^r w_{n_j}^r$
  end
  Send $W = \{W_1, W_2, \cdots W_N\}$ to ENs
end
Return personalized models $\{w_i^R\}_{i\in K}$.

Algorithm 1 presents the workflow of CFL-IDS, which can be described in three stages as follows.

System Initialization: The server first delivers the initial intrusion detection model w^0 to each EN. Then, it defines FL-related hyperparameters, such as the number of communication rounds R, the number of local training epochs E, and the learning rate η. These parameters will guide the FL process.

ENs Update: The EN i will execute Algorithm 2 on the local data set during each round of communication to update the model using the model w_i^{r-1} (w^0 for the first round) and hyperparameters distributed by the server. The updated model w_i^r and EMs are then uploaded to the server.

Server Update: The EMs for the EN i uploaded during the rth communication round are P_i^r. In this way, we can consider the EMs as a time series with multiple attributes for multiple

communication rounds. The EMs sequences of all ENs are subsequently clustered by Algorithm 3. In this way, ENs with similar data distribution can be integrated. There are two points to note. First, to prevent the FL from being impacted by a single inaccurate estimation, we implement Algorithm 3 in each round. Second, we do not use the entire sequence of EMs when executing Algorithm 3. We believe that historical information from too far back is not helpful or even counterproductive for discriminating the data distribution. Therefore, we set a clustering length L, which means that the model's ability to distinguish between the data distribution depends solely on the L rounds of communication before this. Finally, we perform the ICMAM in each cluster of ENs to create a unique model for each EN cluster. Through multiple rounds of communication between the server and ENs, the model formed by the last round of ICMAM (i.e., $\{W_n\}_{n\in N}$ in Algorithm 1, N is the number of clusters) will be sent down to the ENs and used to perform the intrusion detection task.

For a federated system consisting of K ENs, the global objective function is shown in (1) at the bottom of the page.

N is the number of clusters of ENs, and m is the number of ENs in cluster n. They are dynamic and are obtained from Algorithm 3. $g_n(\cdot)$ and $F_i(\cdot)$ represent the objective functions of cluster n and a specific EN i, respectively. The first term of the expression is the training loss of the K ENs, and the second term represents the execution of ICMAM on the server side for all EN clusters. The FL process can be facilitated by alternating optimization of ENs and the server. Specifically, for EN i, the optimization objective in the rth round is

$$w_i^r = \arg\min\left(F_i\left(w_i^{r-1}\right)\right). \tag{2}$$

This optimization process will be described in detail in Algorithm 2. For the server, the optimization objective for the ENs cluster n is

$$\begin{aligned} W_n^r &= \arg\max\left(\mathrm{ICMAM}\left(w_{n_1}^{r-1}, w_{n_2}^{r-1}, \ldots, w_{n_m}^{r-1}\right)\right) \\ &= \arg\max\left(\sum_{j=1}^{m} \alpha_{n_j}^{r-1} w_{n_j}^{r-1}\right). \end{aligned} \tag{3}$$

$\alpha_{n_j}^{r-1}$ is the aggregate contribution of the jth EN in cluster n, which will facilitate cooperation between similar models and presented in Section IV-D.

B. Intrusion Detection Model

A novel intrusion detection model for imbalanced data is designed in this section, and it primarily consists of a neural network and a training loss module, as depicted in Fig. 2.

Neural Network Module: Similar to [26], the hybrid neural network module consists of three parallel CNN, a GRU, and two full connection (FC) layers. For the input vector x, spatially localized features at various scales are slidingly extracted

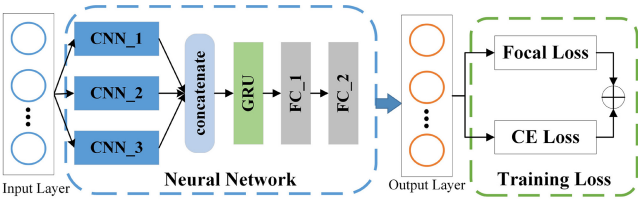


Fig. 2. Intrusion detection model guided by DFL.

by CNNs with convolutional kernels of different sizes

$$\begin{aligned} v_1 &= \mathrm{CNN}_1(x), v_2 = \mathrm{CNN}_2(x), v_3 = \mathrm{CNN}_3(x) \\ u &= \mathrm{Concate}(v_1, v_2, v_3). \end{aligned} \tag{4}$$

where CNN_i, $i \in \{1, 2, 3\}$, is the ith CNN layer, v_i denotes the corresponding CNN layer's hidden vector, and v_1, v_2, and v_3 are combined to produce u and then fed into a GRU

$$h_1 = \mathrm{GRU}(u), o_1 = \mathrm{FC}(h_1), o_2 = \mathrm{FC}(o_1). \tag{5}$$

h_1 is the hidden vector of the GRU layer, and FC_j, $j \in \{1, 2\}$, denotes the jth FC layer. o_1 and o_2 represent the outputs of the two FC layers.

Training Loss Module: Data imbalance is a frequent occurrence in intrusion detection, and it has been demonstrated that Focal loss [9], calculated by (6), is an excellent method to address this issue

$$\begin{aligned} L_{\mathrm{FL}} &= \left(1 - p_{y_i}\right)^{\gamma} L_{\mathrm{CE}} \\ &= \left(1 - p_{y_i}\right)^{\gamma} \sum_{C} -y_i \log\left(p_{y_i}\right) \end{aligned} \tag{6}$$

where L_{CE} is the cross-entropy loss frequently employed for classification tasks, and C is the total number of samples. The true and predicted values of the sample i are y_i and p_{y_i}, respectively. A smaller p_{y_i} indicates lower sample confidence.

The cross-entropy-based tuning factor $(1 - p_{y_i})^{\gamma}$ that the focal loss introduces increases the model's attention to samples with poor confidence by giving them a large weight during training. However, this reweighting strategy will negatively impact when the data are relatively balanced. In this research, we provide a more reliable loss function called DFL. The core concept of DFL derived from curriculum learning [10], which states that the focus on low-confidence samples should be gradually increased during training and fine-tuned in the final stage to enhance the generalization ability of the model. DFL is calculated by

$$\begin{aligned} L &= \alpha L_{\mathrm{CE}} + (1 - \alpha) L_{\mathrm{FL}} \\ \alpha &= \begin{cases} 1 - \beta \frac{e_i}{\mathrm{ER}}, & \mathrm{ER} \geq \beta e_i \\ \left(\beta \frac{e_i}{\mathrm{ER}} - 1\right)/2(\beta - 1), & \mathrm{ER} < \beta e_i. \end{cases} \end{aligned} \tag{7}$$

α is a dynamic weight that will change with the FL process. ER is the total number of iteration epochs, and e_i is the number of current iteration epochs of the local model. The dynamic factor β regulates how the training period is divided.

$$\min_{W_n} G(w_1, w_2, \ldots, w_K) = \min_{W_1, W_2, \ldots, W_N} \left(\sum_{n=1}^{N} g_n(W_n)\right) = \min\left(\sum_{i=1}^{K} F_i(w_i)\right) + \max\left(\sum_{n=1}^{N} \mathrm{ICMAM}\left(w_{n_1}, w_{n_2}, \ldots, w_{n_m}\right)\right) \tag{1}$$

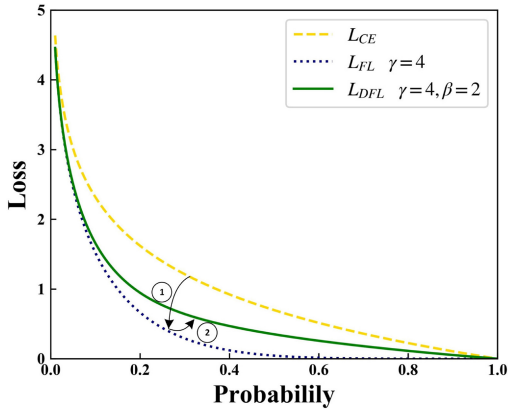


Fig. 3. Trend of DFL.

Algorithm 2: DFL Training Algorithm

Input: Training data $\{x_i\}$ in c_i. Model parameters w_i^{r-1} of neural network. Hyperparameter β, γ, and learning rate η. Training epoch E.

Output: The parameters w_i^r.

for *every epoch* $e = 1, 2 \ldots, E$ **do**
  Compute the DFL by $L = \alpha L_{CE} + (1-\alpha)L_{FL}$
  Compute the backpropagation error for each i by $\frac{\partial L^e}{\partial x_i^e} = \alpha \frac{\partial L_{CE}^e}{\partial x_i^e} + (1-\alpha)\frac{\partial L_{FL}^e}{\partial x_i^e}$
  Update the model parameters by $w_i^r = w_i^{r-1} - \eta \sum_i^m \frac{\partial L^e}{\partial x_i^e} \cdot \frac{\partial x_i^e}{\partial w_i^{r-1}}$
end

For instance, when $\beta = 2$, the first half of training will be a progressive shift from cross-entropy loss to focal loss (see ① in Fig. 3), and the second half of training will be a rise and fall in the cross-entropy loss and the focal loss, respectively, before reaching equilibrium(see ② in Fig. 3). Model training under the guidance of DFL is shown in Algorithm 2.

C. Time-Series Clustering of Evaluation Metrics

The presence of EN natural groupings based on their local data distributions is the fundamental premise for putting this step into practice. For each homogenous group of ENs, we can train an FL model in this manner cooperatively. We divide EN groups using the classic dynamic time warping (DTW) distance-based k-medoids algorithm [27], which mainly consists of two stages: 1) DTW-based k-medoids and 2) determination of the optimal k, as demonstrated in Algorithm 3.

DTW-Based k-Medoids: First, k of all time series are randomly selected as initial centers, and the DTW [28] distance of each time series to each center is calculated. For any two ENs with EMs time series $q_i = [P_i^{r-L+1}, P_i^{r-L+2}, \ldots, P_i^r]$ and $q_j = [P_j^{r-L+1}, P_j^{r-L+2}, \ldots, P_j^r]$, their DTW distances can be calculated by

$$\mathrm{DTW}(q_i, q_j) = \sum_{k=1}^{d} \mathrm{DTW}\left(q_i^k, q_j^k\right)$$

Algorithm 3: Time-Series Clustering of EMs

Input: The time series $Q = [q_1, q_2, \ldots q_N]$ contains the ENs' individual EMs, Number of ENs N.

Output: The optimal number of clusters k, and the set of edge node clusters.

(1) DTW-based k-medoids:
for $k = 1, 2, \ldots, N$ **do**
  Choose k sequences randomly from all time series Q as initial centers
  while SSE_k *not converge* **do**
    The DTW distance of each time series to each cluster center is calculated by Eq. (8);
    Move the sequence to the nearest cluster;
    Find the center sequence in each cluster. i.e., the sequence with the smallest average DTW distance from all sequences in the cluster;
    Calculate SSE_k by Eq. (9).
  end
  Record the SSE_k and the set of clusters for the current k
end
(2) Determine the optimal k:
The linear function between the two endpoints of the elbow joint curve is solved by Equation Eq. (10)
for $k = 1, 2, \ldots, N$ **do**
  $s_k = |SSE_k - f(k)|$
end
$S = [s_1, s_2, \ldots s_N]$
Returns the subscript k of the largest distance $s_k = \max(S)$, and the set of clusters

$$\mathrm{DTW}\left(q_i^k, q_j^k\right) = \min\left\{\sqrt{\sum_{t=1}^{T} w_t^k / T}\right\} \tag{8}$$

where $d = 6$ is the dimension of the EM time series. $\sum_{t=1}^{T} w_t^k, k \le d$ is the warping path of the distance metric square with lengths $L = 20$ created by the EM's time series q_i^k and q_i^k in a single dimension. w_t^k is the element that makes up the warping path, and T is the total number of elements. The similarity between the two sequences increases with decreasing DTW distance.

Subsequently, each sequence is moved to its nearest cluster, the central sequence of each cluster is updated, and the cost function of the k-medoid is calculated by

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{r \in C_i} \mathrm{DTW}(r - m_i) \tag{9}$$

where m_i is the center of cluster C_i, and r represents the sequence within cluster C_i. The clustering is completed when the cost function no longer changes.

Determination of the Optimal k: The elbow joint curve's inflexion point serves as the optimal cluster number in the k-medoid. We will employ this method to find the optimum cluster number in FL automatically. The relationship between k and SSE can be accurately plotted for all possible ways of dividing the ENs, and a linear function between the two

endpoints $(1, \mathrm{SSE}_1)$ and (N, SSE_N) can be obtained

$$f(x) = \frac{(x-1)(\mathrm{SSE}_N - \mathrm{SSE}_1)}{N-1} + \mathrm{SSE}_1. \tag{10}$$

The elbow curve's inflexion point should thus be located at the point furthest from this function, i.e., $s_k = \max(|\mathrm{SSE}_k - f(k)|)$.

D. Intelligent Collaboration Model Aggregation Mechanism

In traditional FL, the model's aggregation depends only on the amount of data at the ENs, as shown in

$$W_n^{r+1} = \sum_{j=1}^{m} \frac{D_{n_j}}{D_n} w_{n_j}^r \tag{11}$$

where W_n^{r+1} is the model parameter for round $r+1$ of EN cluster n, and $w_{n_j}^r$ is the intrusion detection model for round r of the jth EN in cluster n. D_{n_j} is the data volume of the jth EN, and the total data volume of the EN cluster n is D_n.

However, the FL process is not entirely reliable. Poisoning attacks may upset some ENs, resulting in a subpar local model and irreversible impacts on FL. Therefore, we introduced an intelligent collaborative model aggregation mechanism in FL to increase the aggregation contribution of similar models to obtain a more comprehensive personalized intrusion detection model (lines 11–17 of Algorithm 1). We calculate the Earth mover's distance (EMD) to compare the similarity of local models, as in [4]. The average distance between model $w_{n_j}^r$ and the other models in its cluster is first determined by

$$\begin{aligned} D_{n_j}^r &= \frac{\sum_{k=1}^{m} \mathrm{EMD}\left(w_{n_j}^r, w_{n_k}^r\right)}{m} \\ \mathrm{EMD}\left(w_{n_j}^r, w_{n_k}^r\right) &= \inf_{\gamma \in \prod\left(w_{n_j}^r, w_{n_k}^r\right)} E_{(x,y)\sim\gamma}[\|x-y\|] \end{aligned} \tag{12}$$

where $\prod(w_{n_j}^r, w_{n_k}^r)$ is the set of all joint probability distributions of $w_{n_j}^r$ and $w_{n_k}^r$. $\gamma(x, y)$ is the probability that x will appear in $w_{n_j}^r$ and y in $w_{n_k}^r$. From this, the distance expectation $E(x, y)$ of all x and y can be obtained, and its infimum is the EMD distance. The average similarity of each model is then calculated by

$$a_{n_j}^r = \frac{1}{1 + D_{n_j}^r}. \tag{13}$$

The larger the $D_{n_j}^r$, the greater its difference from other models in this cluster and the smaller the average similarity. Each model's aggregate contribution is calculated as

$$\alpha_{n_j}^r = \frac{a_{n_j}^r}{\sum_{j=1}^{m} a_{n_j}^r}. \tag{14}$$

EMD can be considered an optimal linear programming problem with a time complexity of $T(mn^2)$, where m is the number of ENs, and n is the neural network parameter size. Although EMD introduces additional overheads, the algorithm is deployed on the server, and with the help of the server's computational resources, it does not affect the FL process [4]. In addition, the uncomplicated neural network structure and concurrent similarity computation can alleviate the computational burden. Ultimately, the aggregation model for cluster n of ENs is

$$W_n^{r+1} = \sum_{j=1}^{m} \alpha_{n_j}^r w_{n_j}^r. \tag{15}$$

This intelligent collaboration mechanism can effectively facilitate knowledge sharing among those more similar models and prevent the negative effects of subpar models on FL.

V. Experimental Evaluations

In this section, we conduct extensive experiments to evaluate the performance of CFL-IDS. First, we describe the data set. Second, the effectiveness of DFL is evaluated in various imbalanced data partitioning instances using a centralized training strategy. Third, the performance of CFL-IDS is evaluated in the non-IID situation. Finally, it is proven that CFL-IDS is effective at defending against poisoning attacks.

The model for this work was built using the PyTorch API, and our experiments were conducted on a device equipped with an Intel Core i7-10700 and an NVIDIA Geforce RTX3080 Ti (12G). Specifically, the server program runs on the main thread, and the EN programs are executed in parallel by multiple subthreads, all of which are managed uniformly by the thread pool. It should be noted that the maximum number of threads supported by the central processing unit (CPU) used in this experimental platform is 16, which means up to 16 ENs can be allowed to execute in parallel in each round of training. When the number of ENs exceeds the maximum number of threads, the exceeding part will wait for calling in the thread queue.

A. Data Set Description

We comprehensively evaluated the proposed method on the Gas pipeline (GP) [29] and UNSW-NB15 [30] intrusion detection data sets. The GP data set includes seven types of attacks and benign samples, each consisting of 26 features and 1 label. UNSW-NB15 contains nine different attacks and benign samples, each consisting of 42 features and 1 label. Each attack in both data sets has a unique attack strategy for a distinct target. The imbalance ratios of the two data sets reached 573/61 156 and 117/93 000, which are extremely imbalanced.

B. Dynamic Focal Loss Performance Evaluation

Data Resource Partitioning: We set five partitions for each of the two data sets: Balanced, 100-imbalanced, 200-imbalanced, 500-imbalanced, and Extreme-imbalanced. Specifically, each class has 573 examples in the Balanced case. 100-imbalanced means that the number of samples in each class is an arithmetic progression with a common difference of 100. In the 200-imbalance and 500-imbalance cases, the common difference will rise to 200 and 500, respectively. The complete data set will be used for the Extreme-imbalance. The sample size for the smallest category in all data partitions in the GP and UNSW-NB15 data sets are 573 and 117. For each partition, 80% of the data will be used for training and 20% for testing.

TABLE I
INTRUSION DETECTION MODEL HYPERPARAMETERS

Hyperparameters	Count
CNN kernel size	(2, 4, 6)
GRU hidden layer	64
Learning rate	0.01
Batch size	32
Epoch	50

Hyperparameters Setting: Before evaluating the model, we need to ascertain the hyperparameters β and γ of the DFL. We apply the default value of 2 for focal loss to γ. We utilized a grid search to identify the optimal value of β, where the range of β was set to $[2, 6]$, step = 1. The best accuracy was finally attained at $\beta = 4$. We do not recommend overly complex neural network structures for computational resource considerations. The intrusion detection model's additional hyperparameters were specified as shown in Table I. The weight parameters of the model are 88 064 and 88 192 on the two data sets, and the average overhead of ENs to perform one local training is 22.47 and 46.81 s, respectively, which is acceptable.

Baseline Methods: We compare DFL with other state-of-the-art intrusion detection models. Cross-entropy loss and focal loss, respectively, are used by DeepFed [8] and FL-NIDS [23] to guide the training of intrusion detection models. All baseline methods used in our experiments have neural network modules identical to those in this article, and they all employ centralized training to create the models.

Comparison With State-of-the-Art Methods: Tables II and III display the experimental outcomes of various data division strategies. DFL produces significant results on almost all data partitions. Specifically, in the case of relatively balanced data, DFL performs comparably to cross-entropy and is significantly higher than focal loss. The performance attained with focal loss outperforms cross-entropy loss and comes close to the DFL as the imbalance increases. In particular, DFL delivers the best outcomes across all metrics in the case of the 200-imbalance in the GP data set. Since the imbalance ratio drops sharply in Table III, the performance degradation of the cross-entropy loss is even more pronounced. This experiment shows that the intrusion detection model driven by DFL has stronger robustness.

C. CFL-IDS Performance Evaluation

Data Resource Partitioning: Both data sets used in this work have a natural phenomenon of feature distribution skews because they used different means for a specific attack, as mentioned in Section III-A. For label distribution skew, we consider the non-IID situation when some ENs have insufficient global data classes. Specifically, for the GP data set, we put up 100 ENs in our experiments. Each EN contains 7 or 8 classes of data, and the amount of data is 1000–1500. ENs with different classes of samples are non-IID. Additionally, we set the imbalance ratio of each EN as $1/r$ and r as any value among $\{1, 2, 5, 10, 100\}$ to simulate different data partitioning situations. Two categories of data are then randomly designated as the minimum and maximum categories. The ratio of the remaining categories to the minimum category is set $r_{\text{other}} \in [1, r]$ where $r_{\text{other}} \in N^+$, so the specific number of each category can be calculated. It is worth noting that when dividing the data, it is necessary to ensure that both the training and test sets contain data from the selected category. Similarly, 80% of the data in each EN is used for training and 20% for testing. The requirements are the same for the UNSW-NB15 data set, except that each EN now has 9 or 10 classes.

Hyperparameters Setting: We make the number of FL communication rounds $R = 150$ and the number of ENs training epochs $E = 2$. The hyperparameters of the intrusion detection model are the same as in Table I.

Baseline Methods: We compare the proposed CFL-IDS with the current mainstream non-IID solutions. FedAVG [13] is a well-known federated average algorithm, which is applied in many IIoT intrusion detection schemes [8], [12], [16]. Idrissi et al. [17] employed the FedProx [2] FL scheme, which eliminates the client-side drift issue introduced by non-IID by introducing a regularized term in the local model. The closest to our concept is IFCA [5], which uses loss function minimization to determine the cluster to which each EN belongs. The local models for all FL schemes are the same as in this work.

Comparison With State-of-the-Art Methods: We have counted the performance metrics of the different methods and the additional time overhead due to cluster estimation and model aggregation in each round of communication, as shown in Table IV. The traditional FedAVG method performs the worst due to the influence of heterogeneous data. CFL-IDS performs excellently by promoting the cooperation of similar ENs through the clustering mechanism, while FedProx and IFCA are in the middle. As illustrated in Figs. 4 and 5, CFL-IDS exhibits more dependable stability relying on the accurate clustering. At the same time, the other methods fluctuate to a greater extent, especially FedAVG and FedProx, since they both need to jointly maintain a global model, which is hard to come in non-IID cases. In addition, CFL-IDS has a faster convergence speed, which helps reduce the FL system's communication consumption. In terms of time overhead, FedAVG will serve as a benchmark. FedProx mitigates the local model bias by introducing a proximal term deployed on each EN and thus imposes more time overhead. On the contrary, the estimation of clusters and ICMAM by CFL-IDS are deployed on the server, with a slight increase in time overhead. It is worthwhile to note that while IFCA achieves the lowest time overhead, the models are allocated through a broadcast strategy, which introduces extra communication overhead.

D. Evaluation of the DFL on FL

To evaluate the effect of DFL on FL, we employ the baseline methods in Section V-B. The data partitioning and hyperparameters are maintained consistently across methods, as in Section V-C. The experimental results on the GP data set are shown in Fig. 6.

TABLE II
PERFORMANCE OF DIFFERENT METHODS ON DIFFERENT DATA PARTITIONS ON THE GP DATA SET

Partitions	Cross-entropy loss [8]					Focal loss [23]					The proposed DFL				
	Accuracy	Precision	Recall	F1-score	FPR	Accuracy	Precision	Recall	F1-score	FPR	Accuracy	Precision	Recall	F1-score	FPR
Balanced	**0.8157**	0.8146	0.7889	0.8016	**0.0272**	0.8146	0.7749	0.8078	0.7910	0.0293	0.7863	**0.8392**	**0.8176**	**0.8283**	0.0348
100-imbalanced	0.9418	0.9504	**0.9381**	0.9442	0.0084	0.9296	0.9619	0.9197	0.9404	0.0098	**0.9447**	**0.9674**	0.9318	**0.9493**	**0.0077**
200-imbalanced	0.9697	0.9766	0.9561	0.9663	0.0043	0.9609	0.9612	0.9433	0.9522	0.0058	**0.9765**	**0.9797**	**0.9695**	**0.9746**	**0.0035**
500-imbalanced	0.9576	0.9753	0.9376	0.9517	0.0059	**0.9708**	0.9762	0.9430	0.9541	**0.0040**	0.9604	**0.9817**	**0.9532**	**0.9647**	0.0054
Exreeme-imbalanced	0.9467	0.8526	0.8105	0.8310	0.0105	**0.9489**	**0.8536**	0.8150	0.8339	0.0103	0.9479	0.8526	**0.8257**	**0.8389**	**0.0098**

TABLE III
PERFORMANCE OF DIFFERENT METHODS ON DIFFERENT DATA PARTITIONS ON THE UNSW-NB15 DATA SET

Partitions	Cross-entropy loss [8]					Focal loss [23]					The proposed DFL				
	Accuracy	Precision	Recall	F1-score	FPR	Accuracy	Precision	Recall	F1-score	FPR	Accuracy	Precision	Recall	F1-score	FPR
Balanced	0.6264	**0.6713**	**0.6408**	**0.6557**	0.0592	0.5805	0.6119	0.5969	0.6044	0.071	**0.6322**	0.602	0.6342	0.6177	**0.057**
100-imbalanced	0.6819	0.6388	0.5876	0.6121	0.0462	0.6843	**0.6959**	0.5946	**0.6413**	0.0452	**0.7035**	0.6526	**0.6252**	0.6386	**0.042**
200-imbalanced	0.7806	0.6605	0.615	0.6369	0.0467	0.7969	**0.7636**	**0.6474**	**0.7007**	0.0434	**0.8044**	0.7584	0.6438	0.6964	**0.0419**
500-imbalanced	0.802	0.7402	0.7095	0.7245	0.0425	0.8168	**0.7837**	0.7262	**0.7539**	0.0421	**0.8029**	**0.7505**	0.7332	0.7418	**0.0409**
Exreeme-imbalanced	0.8845	0.6988	0.6932	0.696	0.0302	**0.9003**	**0.8679**	0.7202	**0.7872**	0.0284	0.8921	0.7453	**0.7272**	0.7362	**0.0269**

TABLE IV
PERFORMANCE OF DIFFERENT METHODS ON DIFFERENT DATA SETS

Methods	GP Dataset [29]						UNSW-NB15 Dataset [30]					
	Accuracy	Precision	Recall	F1-score	FPR	Time(s)	Accuracy	Precision	Recall	F1-score	FPR	Time(s)
FedAVG [13]	0.7411	0.6632	0.6802	0.6717	0.0326	–	0.6934	0.519	0.621	0.5649	0.136	–
FedProx [2]	0.8791	0.6971	0.6848	0.6909	0.0239	68.47	0.7813	0.6728	0.6949	0.6837	0.0847	84.85
IFCA [5]	0.9201	0.7346	0.6836	0.7081	0.0121	**19.62**	**0.8693**	0.6371	0.6955	0.665	0.0683	**29.73**
CFL-IDS	**0.9407**	**0.8435**	**0.7766**	**0.8087**	**0.0108**	62.14	0.842	**0.7147**	**0.723**	**0.7188**	**0.0436**	75.11

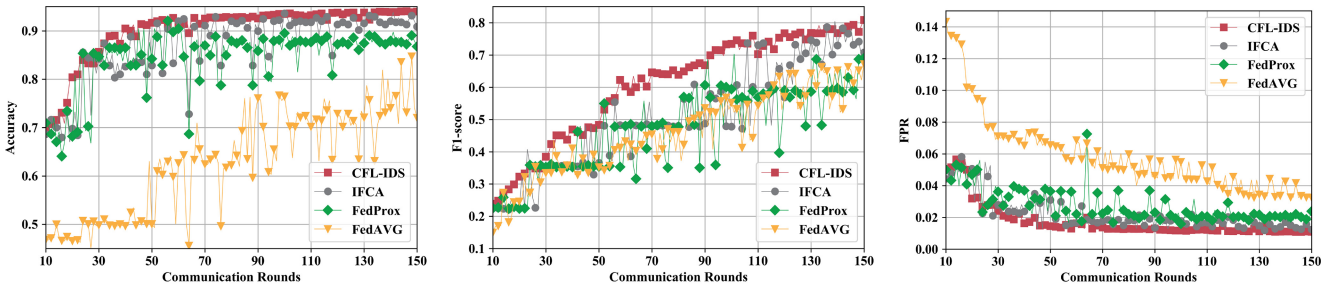


Fig. 4. Trends in EMs for different methods on the GP data set. (a) Accuracy. (b) F1-score. (c) FPR.

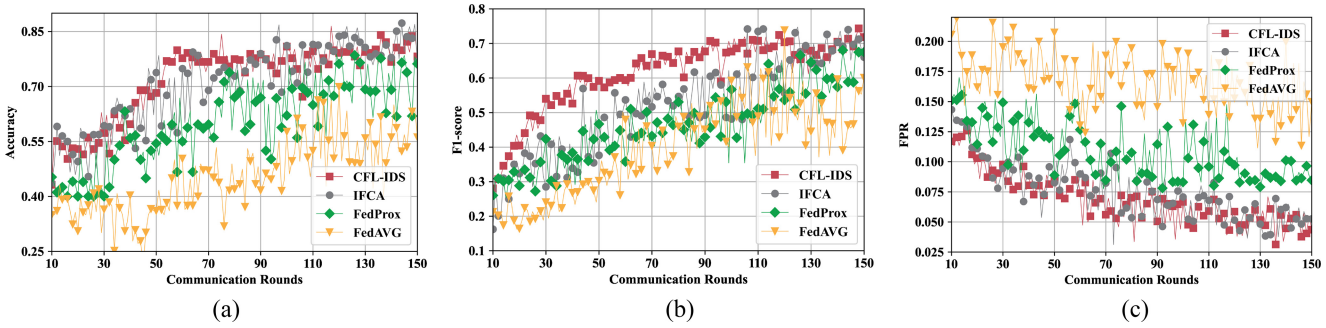


Fig. 5. Trends in EMs for different methods on the UNSW-NB15 data set. (a) Accuracy. (b) F1-score. (c) FPR.

Since the data in most ENs are imbalanced, the method using cross-entropy loss has the lowest accuracy. Benefiting from the adjustment of sample attention during FL, DFL finally achieves optimal results and can be applied to FL with complex data partitioning situations.

E. Clustering Length L Evaluation

By a simple experiment, we explore the effect of clustering length on FL. We set five clustering lengths of $L = \{1, 10, 20, 50, \infty\}$, where $L = \infty$ indicates that the EMs of all current rounds are used. For each case, we recorded their

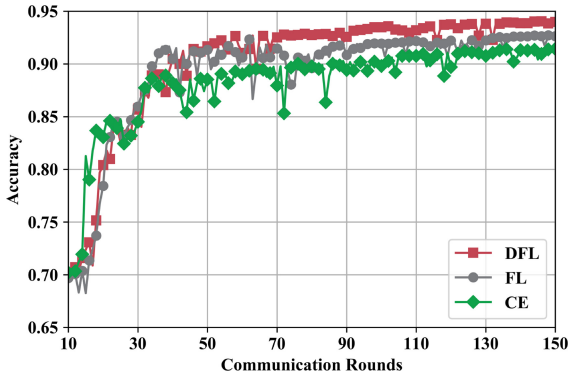


Fig. 6. Evaluation of the effect of DFL on FL.

TABLE V
PERFORMANCE OF DIFFERENT CLUSTERING LENGTHS

Lengths	Acc	Acc=0.85	Acc=0.88	Acc=0.91	Time(s)
$L = 1$	0.9106	103	120	138	**3761.59**
$L = 10$	0.9388	42	64	84	3841.97
$L = 20$	**0.9407**	31	36	45	3883.29
$L = 50$	0.9316	**30**	**35**	**38**	4513.16
$L = \infty$	0.9212	84	105	140	8544.2

accuracy, the number of rounds required to reach the specified accuracy (0.85, 0.88, 0.91), and the total time spent on the clustering operation. The results of the GP data set are shown in Table V.

When $L = 20$, the model's accuracy is at its highest level. It likewise performs well in terms of convergence speed, coming in second only to the case of $L = 50$, but it takes around 630 s less. The model performs poorly at $L = 1$ and $L = \infty$, suggesting that the EMs on a single round and much historical information are not beneficial for clustering. Additionally, we observe that the time consumed by the clustering operation is close when $L \leq 20$, indicating that the calculation of the DTW distance brings little extra cost. However, as L grows, its demand for computational resources will soar rapidly. Table V also means that the average additional cost of considering all possible division scenarios of ENs in each round of communication is 25.9 s, which is perfectly acceptable.

F. Poisoning Attack Defense

Poisoning Attack Simulation: We fully consider the two cases of data poisoning and model poisoning attacks. We randomly choose ten ENs to simulate the data poisoning attack and invert their local data labels (e.g., label 0 becomes 7, 1 becomes 6, etc.). For the model poisoning attack, we randomly selected ten ENSs and truncated their model parameters after local testing (setting those less than 0 to 0), which is uploaded to the server as a subpar model.

Poisoning Attack Defense Effects: We compared traditional FL FedAVG [13] and FedAGRU [15] during experiments, which uses an attentional mechanism to avoid poisoning model updates. We separately counted the average accuracy of ENs that experienced poisoning attacks and those that did not, as shown in Figs. 7 and 8. ENs with dirty data in FedAVG will maintain a low detection level for data poisoning attacks and adversely influence normal ENs through model aggregation. In contrast, normal ENs in CFL-IDS and FedAGRU are less impacted and continue to move upward steadily. For model poisoning attacks, clustering operations are ineffective at separating these low-quality models because the attack occurs after local testing. However, benefitting from ICMAM, CFL-IDS was able to promote knowledge sharing among ENs that were not attacked and provide a high-quality model for the attacked ENs, enabling all ENs to benefit from FL. Similarly, FedAGRU can identify ENs under poisoning attacks and suppress their model uploads through the attention mechanism, so that all ENs end up with similar performance and display stronger fairness. However, FedAGRU's performance on non-IID data is unstable and less efficient overall than CFL-IDS since it optimizes a common global model for all ENs.

TABLE VI
DETAILS OF DATA DIVISION FOR GP

Group	Label set	EN set
1	{0, 1, 2, 3}	{1, 2, 3, 4}
2	{4, 5, 6, 7}	{5, 6, 7, 8}
3	{0, 1}	{9, 10}
4	{0, 7}	{11, 12}

G. Clustering Effectiveness Evaluation

Data Resource Partitioning: We set up a more extreme non-IID distribution on the GP data set to evaluate the clustering effect. Specifically, we divided the 12 ENs into four non-IID groupings, each with 1000 samples: 1) for groups 1 and 2 have data with labels 0–3 and 4–7, respectively and 2) for groups 3 and 4 have data with labels 0,1 and 0,7, respectively. We created these two groups to observe the impact of overlapping labels on the clustering effect. ENs belonging to the same group are IID. To assign data to each EN, we randomly sampled the entire data set. Details of the data division are shown in Table VII.

Clustering Effects: We compared with IFCA [5] during the experiments, and the outcomes are displayed in Table VI. Both methods are capable of correctly clustering ENs over time. However, CFL-IDS has a lower error rate and produces accurate estimates after 11 rounds. In contrast, IFCA is more volatile and needs additional communication rounds to ensure the FL optimizes correctly. This is because CFL-IDS incorporates more historical information, allowing FL to make improvements more quickly, whereas IFCA takes each round of cluster estimate as an independent event. Additionally, both methods reveal more inaccurate estimates for Group 3, which could result from the high proportion of benign samples included in Group 3 and Group 1.

H. ICMAM Effectiveness Evaluation

Through a straightforward design, we validate the aggregation mechanism's efficacy. Specifically, at the 30th communication cycle in Section V-G, we choose the four ENs in Group 1. They are now correctly grouped together. After that, EN 1 is subjected to the data poisoning and model poisoning attacks, which are carried out in the manners described in Section V-F, respectively. The average similarity of the four ENs in the following round is then calculated, as seen in Fig. 9.

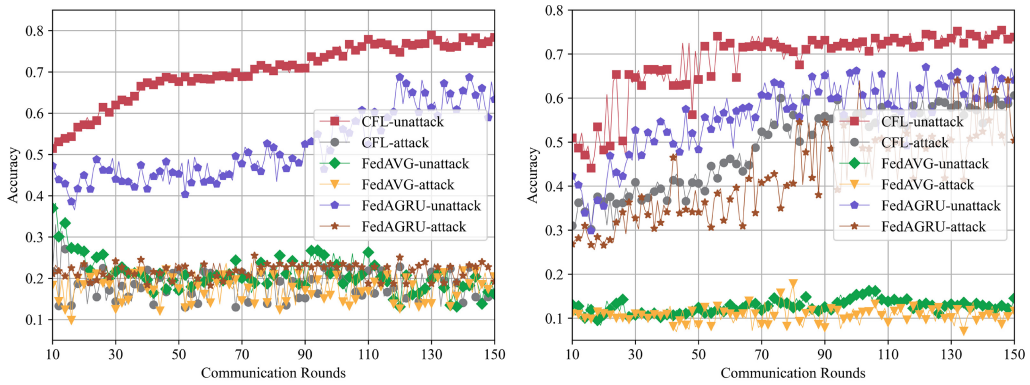


Fig. 7. Effectiveness of FL frameworks under poisoning attacks on the GP data set. (a) Accuracy of the model when subjected to the data poisoning attack. (b) Accuracy of the model when subjected to the model poisoning attack.

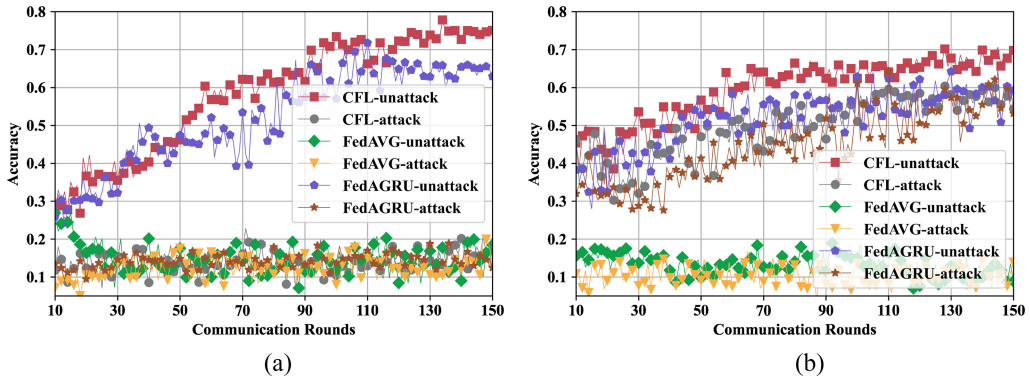


Fig. 8. Effectiveness of FL frameworks under poisoning attacks on the UNSW-NB15 data set. (a) Accuracy of the model when subjected to the data poisoning attack. (b) Accuracy of the model when subjected to the model poisoning attack.

TABLE VII
COMPARISON OF CLUSTERING EFFECTS

	CFL-IDS								IFCA							
Round \ EN	Group 2 (*)				Group 3 ($)		Group 4 (!)		Group 2 (*)				Group 3 ($)		Group 4 (!)	
	5	6	7	8	9	10	11	12	5	6	7	8	9	10	11	12
1	*	!	*	*	#	#	!	!	*	!	*	!	#	#	!	*
2	*	!	*	*	#	#	!	!	!	!	*	!	#	#	!	*
3	*	*	*	*	#	#	!	!	*	!	*	!	#	#	!	*
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7	*	*	*	*	$	#	!	!	*	!	*	*	#	#	!	!
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11	*	*	*	*	$	$	!	!	*	!	*	!	$	#	!	!
12	*	*	*	*	$	$	!	!	*	!	*	*	$	#	!	!
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19	*	*	*	*	$	$	!	!	*	*	*	!	$	#	!	!
20	*	*	*	*	$	$	!	!	*	*	*	*	$	#	!	!
21	*	*	*	*	$	$	!	!	*	*	*	*	$	$	!	!
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
30	*	*	*	*	$	$	!	!	*	*	*	*	$	$	!	!

1. Group 1 (#) is removed because it is always correctly clustered in both methods.
2. Red symbols indicate being incorrectly clustered.

After implementing the poisoning attack, the average similarity between EN 1 and other ENs rapidly decreases, which can assist FL in selecting the ENs that are more similar to collaborate and reduce the harmful effects of subpar models on FL. On the other hand, model poisoning has less of an impact on similarity than data poisoning, suggesting that it could be more detrimental to FL, in line with the findings of [31].

data points collected over time intervals

VI. CONCLUSION

In this article, we suggest the CFL-IDS FL scheme for IIoT intrusion detection. We first developed an intrusion detection model guided by DFL for each EN, which performed well with different data partitions. Second, we propose to use a time series of local model EMs during FL to characterize the data distribution of ENs and enable the DTW-based k-medoids

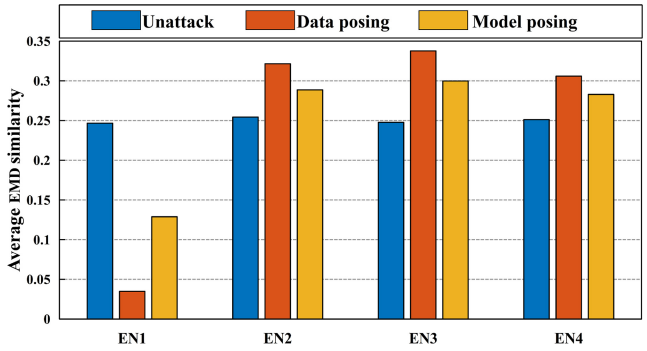


Fig. 9. Effect of poisoning attacks on average similarity.

clustering algorithm to achieve the partitioning of similar edge clusters, which improves the performance of the model in the non-IID case. Finally, we propose an intelligent collaboration mechanism to create a common intrusion detection model for each cluster, thus mitigating the impact of poisoning attacks on federation learning. Numerous experiments demonstrate the effectiveness of the CFL-IDS scheme. In the future, we intend to compare CFL-IDS with more baseline methods and validate its performance in various kinds of actual IIoT applications. We will also explore more lightweight neural network models to detect cyber threats.

REFERENCES

[1] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in Industrial Internet of Things: Architecture, advances and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2462–2488, 4th Quart., 2020.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450

[3] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[4] J. Wang, G. Xu, W. Lei, L. Gong, X. Zheng, and S. Liu, "CPFL: An effective secure cognitive personalized federated learning mechanism for industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7186–7195, Oct. 2022.

[5] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 19586–19597, 2020.

[6] B. Liu, Y. Guo, and X. Chen, "PFA: Privacy-preserving federated adaptation for effective model personalization," in *Proc. Web Conf.*, 2021, pp. 923–934.

[7] X. Huang, J. Liu, Y. Lai, B. Mao, and H. Lyu, "EEFED: Personalized federated learning of execution and evaluation dual network for CPS intrusion detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 41–56, 2022.

[8] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5615–5624, Aug. 2021.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.

[10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[11] L. Cui et al., "Security and privacy-enhanced federated learning for anomaly detection in iot infrastructures," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3492–3500, May 2022.

[12] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day Botnet attack detection in IoT-edge devices," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3930–3944, Mar. 2022.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[14] B. Tahir, A. Jolfaei, and M. Tariq, "Experience-driven attack design and federated-learning-based intrusion detection in industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6398–6405, Sep. 2022.

[15] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion detection for wireless edge networks based on federated learning," *IEEE Access*, vol. 8, pp. 217463–217472, 2020.

[16] N. Hamdi, "Federated learning-based intrusion detection system for Internet of Things," *Int. J. Inf. Security*, 2023, pp. 1937–1948.

[17] M. J. Idrissi et al., "Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems," *Expert Syst. Appl.*, vol. 234, 2023, Art. no. 121000.

[18] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *J. of Big Data*, vol. 8, no. 1, pp. 1–41, 2021.

[19] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and gaussian mixture model for intrusion detection in imbalanced dataset," *Comput. Netw.*, vol. 177, 2020, Art. no. 107315.

[20] A. A. Alfrhan, R. H. Alhusain, and R. U. Khan, "SMOTE: Class imbalance problem in intrusion detection system," in *Proc. Int. Conf. Comput. Inf. Technol. (ICCIT-1441)*, 2020, pp. 1–5.

[21] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: An improved siam-IDS for handling class imbalance in network-based intrusion detection systems," *Appl. Intell.*, vol. 51, no. 2, pp. 1133–1151, 2021.

[22] N. Gupta, V. Jindal, and P. Bedi, "CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems," *Comput. Security*, vol. 112, 2022, Art. no. 102499.

[23] M. Mulyanto, M. Faisal, S. W. Prakosa, and J.-S. Leu, "Effectiveness of focal loss for minority classification in network intrusion detection systems," *Symmetry*, vol. 13, no. 1, p. 4, 2020.

[24] B. Gan, Y. Chen, Q. Dong, J. Guo, and R. Wang, "A convolutional neural network intrusion detection method based on data imbalance," *J. Supercomput.*, vol. 78, pp. 19401–19434, Jun. 2022.

[25] L. Lyu et al., "Privacy and robustness in federated learning: Attacks and defenses," 2020, *arXiv:2012.06337*.

[26] Y. Shan, Y. Yao, T. Zhao, and W. Yang, "NeuPot: A neural network-based honeypot for detecting cyber threats in industrial control systems," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 10512–10522, Oct. 2023.

[27] Y. Chen et al., "Delineating urban functional areas with building-level social media data: A dynamic time warping (DTW) distance based k-medoids method," *Landsc. Urban Plan.*, vol. 160, pp. 48–60, Apr. 2017.

[28] M. Müller, "Dynamic time warping," in *Information Retrieval for Music and Motion*. Heidelberg, Germany: Springer, 2007, pp. 69–84.

[29] T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," in *Proc. Int. Conf. Crit. Infrastruct. Prot.*, 2014, pp. 65–78.

[30] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.

[31] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. 29th USENIX Security Symp. (USENIX Security 20)*, 2020, pp. 1605–1622.

Yao Shan (Graduate Student Member, IEEE) received the M.Sc. degree in computer technology from Northeastern University, Shenyang, China, in 2018, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Engineering.

He joined the Engineering Research Center of Security Technology of Complex Network System, Liaoning Cyberspace Security Professional Technology Innovation Center and Shenyang Industrial Big Data Technology Innovation Center from 2018 to 2022. He founded and served as the CEO of Liaoning Diting Information Technology Company, Shenyang, in 2020. His research topics of interest include network intrusion detection, network threat intelligence analysis, and cybersecurity big data.

Yu Yao (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 1998, 2001, and 2005, respectively.

He has been a Professor and a Ph.D. Tutor with Northeastern University since 2011. And he was the Deputy Director of Shenyang Big Data Administration Bureau from 2015 to 2018. His main research direction is the security of industrial control systems, data analysis and modeling, data visualization, and nonlinear dynamic system analysis.

Xiaoming Zhou received the Ph.D. degree from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2009.

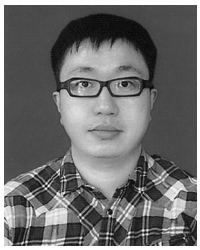
He is currently working with State Grid Liaoning Electric Power Supply Company Ltd., Shenyang. His main research direction is cybersecurity big data, information system operation, data security, and energy big data.

Tong Zhao received the M.Sc. degree in signal and information processing from Northeastern University, Shenyang, China, in 2018, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Engineering.

His work focuses on security of cyber–physical system and machine learning.

Bo Hu received the B.Sc. degree in computer science from Shenyang University of Technology, Shenyang, China, in 1994.

He is currently working with State Grid Dalian Electric Power Supply Company Ltd., Dalian, China. His main research direction is teaching and research work in marketing informatization, network security, and information operation.



Lei Wang received the Ph.D. degree from Harbin Institute of Technology, Harbin, China, in 2011.

He is currently working with State Grid Liaoning Electric Power Supply Company Ltd., Shenyang, China. His main research direction is distributed trusted intelligent security protection technology, new power system network security protection, and trusted data security technology.