EEFED: Personalized Federated Learning of Execution&Evaluation Dual Network for CPS Intrusion Detection

Xianting Huang, Jing Liu, *Member, IEEE*, Yingxu Lai, *Member, IEEE*, Beifeng Mao, and Hongshuo Lyu

***Abstract*— In the modern interconnected world, intelligent networks and computing technologies are increasingly being incorporated in industrial systems. However, this adoption of advanced technology has resulted in increased cyber threats to cyber-physical systems. Existing intrusion detection systems are continually challenged by constantly evolving cyber threats. Machine learning algorithms have been applied for intrusion detection. In these techniques, a classification model is trained by learning cyber behavior patterns. However, these models typically require considerable high-quality datasets. Limited attack samples are available because of the unpredictability and constant evolution of cyber threats. To address these problems, we propose a novel federated *Execution&Evaluation* dual network framework (EEFED), which allows multiple federal participants to personalize their local detection models undermining the original purpose of Federated Learning. Thus, a general global detection model was developed for collaboratively improving the performance of a single local model against cyberattacks. The proposed personalized update algorithm and the optimizing backtracking parameters replacement policy effectively reduced the negative influence of federated learning in imbalanced and non-i.i.d distribution of data. The proposed method improved model stability. Furthermore, extensive experiments conducted on a network dataset in various cyber scenarios revealed that the proposed method outperformed single model and state-of-the-art methods.**

***Index Terms*— Federated learning, cyber-physical system (CPS), intrusion detection, cyber security, personalized model.**

I. Introduction

CYBER-PHYSICAL systems (CPSs), a type of computing system integrated with physical devices, are widely used in many key areas such as manufacturing, traffic control, energy, and safety management. As one of the major enablers for intelligence industry, the combination of cloud computing and CPSs has been the general trend with several practical cases, e.g. cloud manufacturing service platform, support small and medium enterprises (SMEs) with close business cooperation and SMEs supporting industrial cluster collaboration [1], [2], [3], [4]. With the aid of cloud computing, more optimization methods can be created to enhance the reliability and robustness of system, collaboration to expand limited information and efficiency of functions for CPSs.

The rapid integration of advanced network and computing technology has considerably expanded the range of cyber threats. A high-profile CPS security incident occurred in May 2021, when systems of Colonial, the largest oil pipeline operator in the United States, were implanted with ransomware, which resulted in shut down of key fuel network supplying oil to the eastern states [5]. This event revealed that the rapid improvement of network-operating technologies poses new challenges in maintaining the high-level security of CPS systems. NIST Guide to ICS Security [6] revealed the importance of cyber security for the modern industry.

Many AI-based intrusion detection methods have been proposed to ensure CPS security [7], [8], [9]. However, although most of the proposed algorithms exhibit satisfactory performance, they are based on the assumption that the datasets reflect the actual scenario of cyberattacks. However, in practice, datasets with limited samples of cyberattacks are available to users. Because of security considerations and privacy policies, CPS users tend to not share their private samples of multiple attack. Furthermore, the unpredictability and rapid evolution of unknown cyberattacks increases the difficulty of acquiring samples and retraining the models [10]. In this case, CPS users who are lack of samples and who are relatively sufficient samples intend to gain more efficiency through security compliance without sharing their private data. Federated learning (FL) is a secure method for industrial-cooperation-based CPSs which only transmits model parameters by encrypting, as displayed in Fig. 1. FL [11] was proposed by Google in 2016 to collaboratively improve model performance while keeping their data private. For example, Huong et al. combined FL with machine learning for anomaly detection in industrial control systems to prove that the FL framework can achieve superior detection accuracy and reduce transmission link bandwidth consumption [12].

Although the use of FL is a solution for the limited availability of cyberattack samples, other problems remain unresolved. In 2020, Li et al. [13] proposed an FL framework

Manuscript received 22 February 2022; revised 5 June 2022 and 13 August 2022; accepted 29 September 2022. Date of publication 14 October 2022; date of current version 7 December 2022. This work was supported in part by the National Key R&D Program of China (Key Technologies and Applications of Security and Trusted Industrial Control System, No. 2020YFB2009500) and Beijing Natural Science Foundation (No. L192020). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mika Ylianttila. *(Corresponding author: Yingxu Lai.)*

Xianting Huang, Jing Liu, Beifeng Mao, and Hongshuo Lyu are with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: huangxt@emails.bjut.edu.cn; jingliu@bjut.edu.cn; maobf19@emails.bjut.edu.cn; lvhs@emails.bjut.edu.cn).

Yingxu Lai is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China, and also with the Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education, Beijing 100124, China (e-mail: laiyingxu@bjut.edu.cn).

Digital Object Identifier 10.1109/TIFS.2022.3214723

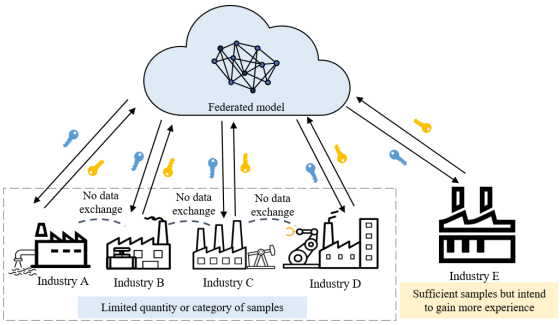


Fig. 1. Motivation and requirement of industrial cooperation through federated learning.

to collaboratively construct the CPS intrusion detection model. Chatterjee et al. [14] proposed a FL-based intrusion detection system in which federated average (FEDAVG) and noise tolerant are used to address tag noise. Nguyen et al. [15] used FL to collect aggregate behavior profiles to build anomaly detection systems. The above works introduced FL into CPS intrusion detection and improved the detection model structure by using the existing FL algorithm. But all of works are based on the assumption that distributed data is evenly distributed. Preuveneers et al. [16] proposed to audit and hold accountable FL federated learning model updates using blockchain technology to prevent sample damage to model training. The differences in collected data between various participants due to conflicts with various cyber scenarios are not considered. Most damages to model training are causing by such differences but not poison samples. Such differences lead to statistical nonindependent and identically distributed data distribution in which the distribution quantity and probability of each sample category on the same distributed client differs considerably. As a major FL problem, this problem considerably affects FL performance. Moreover, limited personalization results in the construction of an intensive common model, which results in a dissatisfied accuracy performance confronted with local cyber scenarios. Our work is to enable the model to learn selectively to minimize negative impacts rather than ignoring or providing a one-for-all solution to these problems.

To construct a personalized model and mitigate FL inherent inadequacies, we first designed a novel FL framework, which was combined with our personalized optimization update algorithm based on the FEDAVG [17] algorithm. This technique allows multiple participants to collaboratively construct the intrusion detection global model in the execution network. Furthermore, the asynchronous computation of the dual network designed in the framework was combined with the proposed optimal backtracking replacement algorithm to ensure the sustainable stability of the model and reduce the consumption of the system in the evaluation network. The main contributions of this paper are follows:

1. To overcome the weakness of FL, we propose a dual $Execution\&Evaluation$ network FL framework (EEFED), which generates both global model and personalized local model. The $Execution$ network obtains the ideal of updating models without moving data, which not only ensures data privacy, but also better absorbs sample knowledge and greatly improves the accuracy of participants who have limited data or no samples. The personalized local model generated by the $Evaluation$ network can not only learn from the global model to detect unknown attacks, but also better adapt to local application scenarios and improve detection accuracy. The asynchronous computation between dual networks reduces computing time through making using of the idle time of the participants.

2. We proposed an optimized personalized update algorithm and optimal backtracking replacement algorithm in EEFED. To satisfy the personalization requirement, the $Environment\ Similarity$ parameter was introduced in the optimized personalized update algorithm to dynamically update the model. The personalized update algorithm alleviates the accuracy degradation of FL performance caused by data imbalance and the non-i.i.d distribution problem. To ensure FL stability and sustainability, an optimal backtracking replacement policy was implemented to ensure the optimality of every model parameter update process.

3. We used two CPS traffic datasets and a TCP/IP traffic dataset to conduct experiments. We demonstrated that EEFED was effective in both local and global scenarios and could be adapted to additional participants and unknown attacks. Moreover, compared with three state-of-the-art studies, we proved that EEFED exhibits superior adaptability and effectiveness to a complex cyber scenario. In the experiments, EEFED achieved an accuracy improvement of approximately 3% over comparable methods. Furthermore, the stability and timecost of the proposed method were higher and lower, respectively, than those of conventional methods. A gain of at least 13.19% local unknown attack accuracy was achieved over the single local model.

The rest of this paper is organized as follows: In Section 2, we elaborate and analyze the problems of FL to be solved. In Section 3, we review studies related to FL and intrusion detection and summarize the existing problems with FL applications in building collaborative intrusion detection. In Section 4, we introduce the system model and threat model considered in our method. In Section 5, we describe the proposed method in detail. In Section 6, we discuss the experiments conducted to validate the proposed method using two different CPSs traffic and a TCP/IP traffic datasets. Finally, we present the conclusion in Section 8.

II. Challenges of the FL Environment

Before focusing on improving FL detection performance, we first consider the unique characteristics that distinguish FL from other distributed training settings such as parallel training. In FL, the distribution of both training data and computational resources is a fundamental and fixed property of the learning environment of each participant. Participants have absolute control over their devices and data which means they can stop their devices from participating in computing and communication at any time. The following challenges are prevalent:

1) **Unbalanced and non-i.i.d data:** Because the training data on the individual clients are collected by the clients based on their local network traffic environment and usage

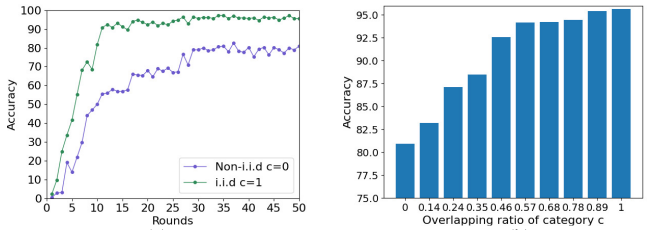


Fig. 2. Preliminary experimental result of FL with no optimized solution in various data category distributions. (a): Comparison of the accuracy result on SWaT dataset of non-i.i.d distribution with c = 0 and i.i.d distribution with c = 1 in 50 rounds of communication. (b): Accuracy on the SWaT dataset of various overlapping ratios of category in 50 rounds of communication.

patterns, both the size and distribution of the local data categories typically vary considerably for various participants. Fig. 2 shows that the optimal global model target of the non-i.i.d data is far different from the local model than i.i.d data. In Fig. 2, c is defined as the overlapping ratio of category of all participants local data. s_i is defined as participant i the number of category of its local data. The overlapping ratio of category can be described as:

$$c = \frac{s_i \cap s_j}{\sum_{k=1}^{N} s_j} \tag{1}$$

where N is the total number of all participants, and i and j are the subscripts of any participant. Here, we set the c to 10 different levels. When $c = 0$, it means that any of local dataset is of different categories. When $c = 1$, it means that all local datasets of participants are distributed by i.i.d.

The characteristic of the existence of FL considerably affects performance. Thus, the general averaged model may be far different from the global optima, especially when the local data distribution differs considerably. Eventually, the converged global model exhibits lower accuracy than that of the i.i.d setting.

2) **Indeterminate number of participants:** FL environments may constitute of multiple participants with various computing capacities. Generally, in the FL for CPS detection, not all participants may participate in each communication round. Participants can lose their connection, run out of battery, or seize to contribute to the collaborative training for other reasons. Furthermore, because the quality of the collaboratively learned model is determined by the combined available data of all participants, collaborative learning environments exhibit a natural tendency to grow or temporary tendency to decrease.

3) **Personalized requirement for multiple participants:** The optimization goal of FL is global optimality for all participants. In this mode, the desired results cannot be achieved in scenarios applicable to local participants. Participants should have the ability to detect the personalized traffic of a local scene while gaining shared experience to obtain local unknown traffic.

Based on the aforementioned characterization of the FL environment, an efficient distributed training method for FL should satisfy the following requirements:

1) Obtain fast model convergence and high performance in a small number of training rounds by sharing experiences to gain local unknown knowledge while not sharing private local data. The model should be robust to non-i.i.d unbalanced traffic data both in unbalanced data size and categories.

2) Establish a more stable global model for participants lacking or even without data.

3) Develop and improve a personalized model and fine-tune the personalized model to confront with cyber scenarios having local particularity of different participants.

III. Related Work

In this section, we briefly review relevant studies on FL and address the non-i.i.d data distribution challenges and personalized requirements in FL.

FL was first proposed by Google in 2016 to collaboratively develop a machine learning model to solve the ùdata island ì problem by using the data in the distributed environment while preventing data leakage. Yang et al. [11] proposed three types of security federation learning frameworks, among which the horizontal federation framework is widely adopted when the datasets share the same characteristics but distinct sample space. However, a basic horizontal federated framework cannot guarantee individual requirements of the participants. We designed and improved the network security scenario based on the horizontal federation learning framework. The proposed methods are an improved version of the horizontal federation learning framework for various cyber security scenarios.

Because training data on an individual are collected by the participants at different times based on their local cyber scenarios and patterns of utilization, the size and category distribution of the local datasets of participants tend to vary considerably. Non-i.i.d data can markedly influence FL accuracy. To overcome this statistical challenge, McMahan et al. [17] proposed the FEDAVG algorithm in which each participant executes multiple SGD iterations to calculate weight updates, rather than updating immediately after each iteration. However, Sattler et al. [22] performed preliminary experiments and proved that the FEDAVG algorithm has limited effect. Yao et al. [23] adopted a feature fusion operator to reduce communication rounds and achieved a higher accuracy than that of the FEDAVG algorithm.

Furthermore, because of the discrepancy between the datasets distributed among the participants, the benefits of participating in FL are debatable for participants who have sufficient data or only encounter a single cyber scenario. Yu et al. [24] proposed that for different tasks, the global intensive model is not as accurate as a single local model trained by itself; thus, some participants may not benefit in any way from FL. Hanzely et al. [25] queried the utility of the global model in which the local daily security requirements differ considerably. Thus, training a single global model suitable for all participants becomes difficult.

To address statistical heterogeneity and non-i.i.d distribution challenges of data and satisfy the personalized requirements of local participants, the global model must be personalized.

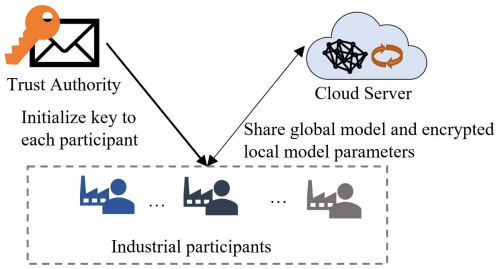


Fig. 3. System architecture.

Masour et al. [26] clustered similar participants to train a separate model for each group. Jiang et al. [27] fine-tuned the initial model to a personalized model based meta learning and indicated that optimizing the performance of the global model alone will deteriorate subsequent personalization. To achieve individualization and solve the problem of heterogenous data distribution, most current methods use prior knowledge to train models in a group. Yang et al. [28] developed a FL framework that aggregated and updated model parameters in an asynchronous way. The problem of personalized resource management of UAV is solved by reinforcement learning and the excution time of FL is shortened effectively by asynchronous aggregation. With the increase in federated participants, the federated system has low scalability because of the difficulty of obtaining prior knowledge. We used meta learning to fine-tune the model internally to overcome scalability problems. Based on reinforcement learning through the evaluation network, feedback on the evaluation results of learning provides the system the ability to continuously learn. Based on prior studies [13], [23], [27], [28], our methods solve the aforementioned problems through the optimization of the framework and improve the deployed algorithm.

IV. System Model and Threat Model

In this section, we introduce the system model and threat model considered in this manuscript.

A. System Model

As shown in Fig. 3, our system model consists of three entities: Trust Authority, cloud server and industrial participants.

- Trust Authority (TA): TA assigns public and private keys to each participant as the basis for FL encryption. TA would not participate in the entire implementation process of FL until disputes arise. TA is an authoritative third-party certification enterprise.
- Cloud server: Cloud server aggregates the model gradients uploaded by participants in current FL system and sends the global model of the aggregation results to each participant. During the entire implementation process of FL, the cloud server only access to the encryption gradients and aggregation results. The cloud server is a mature FL service platform.
- Industrial participant: Industrial participants are responsible for building detection models based on their local data and uploading the encryption model gradients to update FL model.

B. Threat Model

In our consideration of FL, TA would not participate any operations with local data or models. Further, stealing data or key exposures is of no benefit to TA and may even have a significant impact on the reputation of enterprise. Therefore, we assume that TA are completely honest and trusted. Both cloud server and industrial participants are semi-honest and not colluding entities. They are honest in their compliance with agreements but also curious about the private data of other industrial participants. However, the cloud server only processes the processed model parameters. And it is difficult for cloud server to infer the specific individuals from lots of participants which needs specific identification and differentiation. More attacks are directed at local models. For example, [33] demonstrate attack from insider participants which use generative adversarial nets (GAN) to mimic prototypical samples of the other participants' training set. Therefore, we focus on attacks on local models and consider two threats in the following. First is model corruption by uploading malicious model gradients from internal participants. Second is the possibility that external attackers could inject malicious model gradients through communication links.

Our FL method calculates the similarity between the updated model and the local model through personalized update algorithm, and to some extent identifies whether non-i.i.d difference or injected malicious model gradient exists. If malicious participants damage the model, the update algorithm will selectively update the model. Further, the proposed Optimizing Backtracking Parameters Replacement Policy will dynamically evaluate the model parameters of each update. If the model parameters caused damage to the model training of the FL system, the model parameter update is invalid. The model parameters with better evaluation in the saved history were selected for the next round of communication.

V. Proposed Methods

In this section, we first introduce the proposed framework, then elaborate on our proposed personalized update algorithm and optimize backtracking parameters replacement policy.

In EEFED, multiple CPS participants are combined to confront various cyber scenarios and gain experience to detect local unknown attacks. To achieve this goal, we improved the horizontal federated framework including both central global and local personalized models.

The proposed FL framework consists of two main components as cloud server and local FL participants. Cloud server is defined as G, which maintains a secure channel, and the trained global model as M_g. Suppose N local participants are defined as $L = \{l_1, \ldots l_N\}$, and they all want to acquire unknown knowledge by sharing local datasets $D_k (k \in N)$ in a privacy-preserving manner to contribute a global model M_g while obtaining local personalization models $M_l = \{m_{l_0}, \ldots, m_{l_N}\}$. The other key notations used in this section are summarized in Table I.

Two types of participants typically exist in a real-world FL scenario. As the main contributor to the FL model, this type of participants should ensure gaining experience without decline

TABLE I
LIST OF KEY NOTATIONS

Symbol	Description
N	Total number of participants.
K	Number of selected active training participants.
R	Number of communication round.
D_k	Local datasets.
LD_k	Test dataset for participant l_k in local scenarios.
GD	Test datasets in global scenarios.
$AvgAcc_t^l$	Average accuracy of local models in round t.
w_t^g	Global model parameters in round t.
$w_t^{l_k}$	Local model parameters for participant l_k in round t.
W_t^l	The set of total local model parameters in round t.
$v_t^{l_k}$	Proposed extracted local model gradients matrixes for participants l_k in round t.
V_t	The set of extracted local model gradients matrixes for total participants in round t.
$e_t^{l_i l_j}$	Proposed environment similarity of participant l_i with participant l_j in round t.
$E_t^{l_k}$	The set of proposed environment similarity for participants l_k with other participants in round t.

in model performance which is caused by data imbalance from other participants. And we define this type of participants as type I. The other type of participants which represents most participants in the FL scenario, should ensure direct benefits because of the lack of sample or computational power. And we define this type of participants as type II. EEFED regulates the local personalized models according to the requirements of various participants. Unlike the ordinary FL framework, both the central global and the local models for each participant are maintained in EEFED. The performance of the personalized local model is optimized to be superior to that of the first type participant local models. The central global model optimizes the goals of all participants and can be used directly by the second type of participants.

A. Design of the Dynamic Execution&Evaluation Dual Network Federated Framework

The proposed EEFED framework is categorized into two parts, namely execution and evaluation networks. In the execution network, both local and global models exchange model parameters. In the evaluation network, to reschedule the imbalance of the participant data and optimize system consumption, a dynamic changeable hyperparameter *Environment Similarity* calculator and model parameter replacement policy inspired by reinforcement learning are used. The dual network FL *Execution&Evaluation* framework that is used to collaboratively build personalized deep learning intrusion detection models is displayed in the Fig. 4.

The execution and evaluation networks are synched. During the operation of the execution network, the evaluation network calculates the optimization indicator and evaluates model parameter scores. The idle time of the distributed computing participants is considered to optimize model performance and system consumption. In the EEFED evaluation network, the difference between each participant's cyber scenarios are dynamically evaluated using *Environment Similarity* parameters. Local model rescheduling is performed with the least parameter transmission. The complete workflow of the EEFED

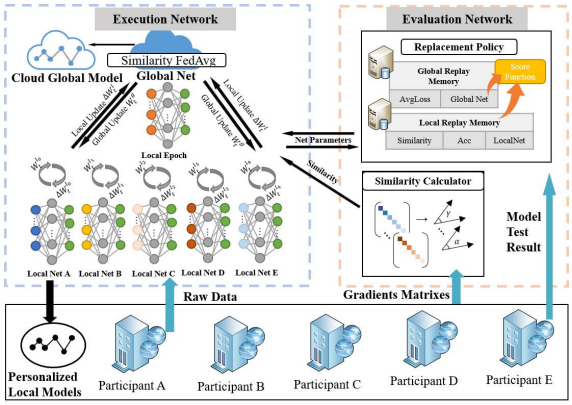


Fig. 4. EEFED framework overview.

method can be described in four phases, which is given below and in Algorithm 1.

Algorithm 1 EEFED Workflow

Input: Participants set L, data resources of all participants $D_k(k \in N)$, number of communication rounds R.
Output: The comprehensive global model M_g and N personalized local models m_{l_k}.

1) Initialization: A secure channel is established between the cloud server G and each $l_k(k \in N)$; G initializes model settings η, $\mathcal{L}$;
While $t \leq R$ **do:**
2) Initialization for Dual Network:
  For each $l_k(k \in N)$ **in Execution Network do:**
    Initial model parameters w_t^g, W_t^l;
    Update local model with local data D_k;
  For each $l_k(k \in N)$ **in Evaluation Network do:**
    Extract model gradients $V_t = \{v_t^{l_1}, \ldots, v_t^{l_N}\}$;
    Computes Environment Similarity $E_t^{l_k}$ via (5)
    and send $E_t^{l_k}$ to Execution Network;
3) Synchronous Communication Round:
  For each $l_k(k \in N)$ **do:**
    Upload the local model parameter $W_{t+1}^{l\,\prime}$;
  For cloud server G **do:**
    Aggregates the w_{t+1}^g with local weights $W_{t+1}^{l\,\prime}$
    via (6);
    Optimizes the w_{t+1}^g via Algorithm 3;
    Broadcast the final global model parameters w_{t+1}^g;
  For each $l_k(k \in N)$ **do:**
    Personalized update the local model parameter
    W_{t+1}^l via (7).
4) Local Optimation:
  For each $l_k(k \in N)$ **in Evaluation Network do:**
    Test the current local model and save results.
    Optimizes the W_{t+1}^l via Algorithm 3.
t ← t + 1;
return global model M_g and N personalized local models m_{l_k} with parameters w_R^g, W_R^l;

1) System Initialization: Secure channels are established between cloud sever G and each participant l_k in the FL execution and evaluation networks. Then, G and each L initialize

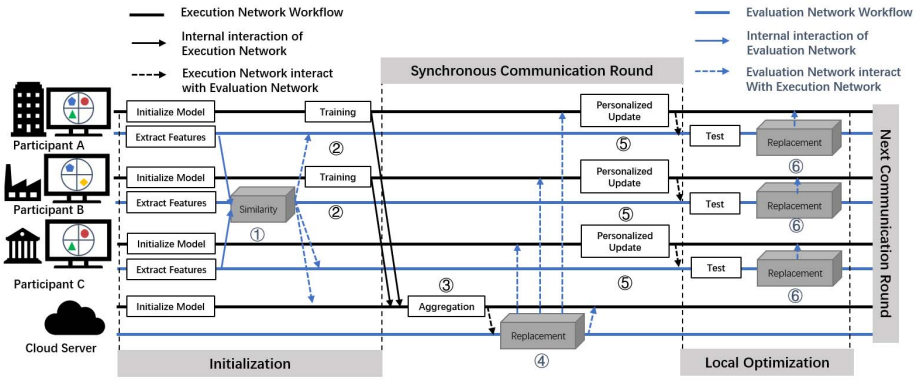


Fig. 5. Interaction of the dual network for EEFED. Execution network: training of the local model(②)Aggregation of the global model(③) Personalized update of the local model(⑤). Evaluation Network: similarity calculating between participants(①), replacement policy for the global model(④), replacement policy for the local model(⑥).

parameters related to model training such as learning rate η, loss function $\mathcal{L}$, and the number of active training participants K.

2) Initialization for the Dual Network: In the execution network, G and each l_k select the parameter set w_t^g, W_t^l of the last round t communication of the global model and the local model. After the execution network is initialized, the selected K participants train the local model with its own private data resource $D_k (k \in N)$.

In the evaluation network, each l_k extracts model gradient matrixes $V_t = \{v_t^{l_1}, \ldots, v_t^{l_N}\}$ from the initialized local models. Each l_k computes environment similarity parameters between other participants as a balanced factor, which is proposed in the next section, through $SimCal(V_t)$. Then participants send the environment similarity parameter sets $E_t^{l_k}$ to the execution network.

3) Synchronous Communication Round: In the synchronous communication phase, each l_k uploads the updated local model parameter $W_t^{l'}$. After G receives the parameters of round t corresponding to local parameters, the environmental similarity parameters are obtained from the evaluation network as the balanced factor. The *Globalupdate* is proposed in the next section to aggregate the local model parameters. Then, the generalized global model parameter w_t^g is obtained. At this stage, updated global model parameter w_t^g is optimized through interactions with the evaluation network and Algorithm 3, which is proposed in the next section. Next, the replaced final global model parameters are encrypted and broadcast to participants, and the global model is saved. After L receives the w_t^g of round t, participants update the local model parameters with *Localupdate*, which is proposed in the next section, to personalized update local model.

4) Local Optimization: In the local optimization phase, each l_k tests the local model and details the accuracy results and current state of the local model with the evaluation network. Next, the current local model parameter $w_t^{l_k}$ is optimized through Algorithm 3. Thus, the replaced final local model parameters are saved. The corresponding test results of the local model of this round are fed back to the next round of communication in the evaluation network.

The interaction process between the execution and evaluation networks is displayed in Fig. 5. In initialization phase, participants calculate Environment Similarity in Evaluation Network after initializing models(Fig.5 ①) and the model training process(Fig.5 ②) can be performed in Execution Network simultaneously. In synchronous communication round phase, the whole process can only be executed sequentially. Cloud server aggregate the local models (Fig.5 ③) and optimize the model by proposed replacement policy (Fig. 5 ④). Then the model broadcast to participants to personalized update their local models (Fig. 5 ⑤). In local optimization phase, participants optimize their local models through proposed replacement policy before next communication round training start (Fig. 5 ⑥).

The dual network structure can use the waiting time of local participants to calculate the balanced factor and the test feedback of the model. The asynchronous computing setup of the dual network ensures participant's idle time is utilized and improves model performance by using feedback mechanisms and the balanced factor without affecting the execution flow of the FL system.

Directly retraining the global model with the participant's local data may degrade model performance (such as slow convergence speed, decreased accuracy, and overfitting on a small number of data samples) [27]. To address this problem, the proposed framework incorporates personalized update algorithms in combination with the execution and evaluation network. We elaborate on the update algorithms used in our framework in the next section.

B. Personalized Algorithm for FL-Update-Based FEDAVG

By sharing model parameters, FL enables participants to contribute to a shared global model without sharing their private data. Deep learning, with its excellent generalization and model parameter inheritance abilities, is generally used for FL. Suppose N participants $L = \{l_1, \ldots l_N\}$ and the corresponding local dataset $D_k (k \in N)$, n_k is the number of samples available on participant l_k, $n = \sum_{k=1}^{N} n_k$ is the number of samples of all participants in one round of communication, where K is the active participant in the current round of communication.

Therefore, the FL problem is transformed into an empirical risk minimization [25] problem, as follows:

$$min_{w\in\mathbb{R}_d} f(w) = \sum_{k=1}^{N} \frac{n_k}{n} F_k(w),$$
$$where\ F_k(w) = \frac{1}{n_k} \sum_{i\in\mathcal{D}_k} f_i(w) \tag{2}$$

where the objective function $f(w)$ of the global model can be expressed as the aggregation of the objective function $l_k(w)$ of the local model linearly. In this study, the objective function is defined as the cross-entropy loss in the classification task. w, the model parameters that need to be iterative learned, are defined as the weights and biases in the deep learning network.

FEDAVG [17] is an improved version of the original FL algorithm FEDSGD [29] and increases the number of local training iteration. In the $t-th$ round communication, each participant performs a certain number of iterations of stochastic gradient descent (SGD), calculates $g_k = \nabla F_k(w_t)$, uploads the gradient of the current local model, aggregates the gradient submitted by all participants according to the size of data, and updates the global model through the following equation:

$$w_t \leftarrow w_{t-1} - \eta \sum_{k=1}^{\mathrm{K}} \frac{n_k}{n} g_k \tag{3}$$

FEDAVG has been proven to be accurate and robust in image classification tasks [25], and it is critical for the FL framework [24]. Adaptive moment estimate (ADAM) and SGD algorithms are typically used as gradient optimizers in FL. In the traffic classification task, we conducted a preliminary experiment to compare the ADAM with SGD algorithms. The results revealed that the accuracy and loss of the ADAM algorithm were higher and lower, respectively, than those of the SGD algorithm. Therefore, in our federated framework, the ADAM algorithm was used as an update optimization module, as follows:

$$w_t \leftarrow w_{t-1} - \eta \frac{\hat{s}}{\sqrt{\hat{r}+\delta}} \sum_{k=1}^{\mathrm{K}} \frac{n_k}{n} g_k \tag{4}$$

where $\hat{s}$ is the corrected deviation of the first moment, $\hat{r}$ is the corrected deviation of the second moment, and δ is the small constant used for numerical stability. For convenience, we denote $\frac{\hat{s}}{\sqrt{\hat{r}+\delta}}$ as β in the manuscript.

Since the intrusion detection implemented in this manuscript targeted at multi-classification tasks, the cross-entropy function is selected in the subsequent experiments and defined as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i} \sum_{c=1}^{M} y_{ic} log(p_{ic}) \tag{5}$$

where M is the number of classification categories, y_{ic} is the symbolic function and p_{ic} is the predicted probability of target sample i belongs to class c.

According to Eq. 4, the FEDAVG algorithm is a fine-tuned version of the FL model based local data. Averaging the model parameters can easily produce a high-precision global model, but it may damage the model's ability of subsequent personalization [24]. Because participant datasets are unevenly distributed in category and size, the local model trained with indiscriminate updates may not respond to the local cyber scenarios. In this case, we propose a hyperparameter called *Environment Similarity*, which is dynamically calculated in each FL round. Here, *Environment Similarity* is a balanced and personalization factor calculated in the evaluation network and is added to optimize the update algorithm.

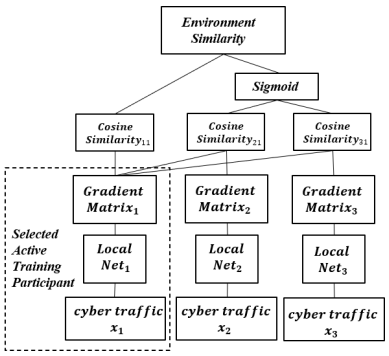


Fig. 6. Example of a concise process to calculate *Environment Similarity* among three participants.

We define the *Environment Similarity* parameter between each participant as $E_t^l = \{E_t^{l_1}, \ldots E_t^{l_N}\}$, which is used to measure the difference between cyber scenarios. While $E_t^{l_i} (i \in N)$ is the set of *Environment Similarity* between participant l_i and other participants, which is defined as $E_t^{l_i} = \{e_t^{l_i l_1}, \ldots e_t^{l_i l_N}\}$. Fig. 6 shows an example of a process to calculate *Environment Similarity*. The gradient matrixes are extracted from each round, and initial local models are flattened into vectors. Next, the product of vectors is used to obtain the angles between each matrix. The value of cosine similarity is obtained by angles in $[-1, 1]$, as follows:

$$e_t^{l_i l_j} = \left\langle \frac{v_t^{l_i}}{\left\| v_t^{l_i} \right\|}, \frac{v_t^{l_j}}{\left\| v_t^{l_j} \right\|} \right\rangle \tag{6}$$

where $e_t^{l_i l_j}$ is the cosine similarity of participant l_i to l_j, $v_t^{l_i}$ and $v_t^{l_j}$ are gradient matrixes, and l_i and l_j are the indices of each participant. In each communication round, participants waiting for an update should calculate *Environment Similarity* between the selected active training participants $l_k (k \in N)$. Then the preliminary computing results of *Environment Similarity* go through the $Sigmoid$ function into range [0, 1], as shown in Fig. 5. The proposed global and local update function can be summarized as Eqs. 7 and 8, respectively, and the local personalized update algorithm is described as Algorithm 2.

$$Globalupdate(w_{t-1}^g, W_t^{l_i}, \mathrm{E}_{t-1}^{l_i})$$
$$= w_t^g \leftarrow w_{t-1}^g - \eta\beta \sum_{k=1}^{K} \frac{n_k \sum \mathrm{E}_t^{l_k}}{nK} g_k \tag{7}$$

$$Localupdate(w_{t-1}^{l_i}, w_t^g, e_{t-1}^{l_i l_k}) = \begin{cases} w_t^{l_i} \leftarrow w_{t-1}^{l_i} - \eta\beta \triangle_\theta, & \text{if } i \in \mathrm{K} \\ w_t^{l_i} \leftarrow w_{t-1}^{l_i} - \eta\beta \sum_{k=1}^{\mathrm{K}} \frac{n_k e_t^{l_k l_i}}{n \sum \mathrm{E}_t^{l_k}} \triangle_\theta, & \text{if } i \notin \mathrm{K} \end{cases} \quad (8)$$

where $W_t^{l_i}$ is the linear sum of uploaded local model parameters, K is the number of selected active participants, and $\frac{\sum E_t^{l_k}}{K}$ is the average similarity of participants extracted for training and is used as the indicator of global model update. The similarity of participants in this round of training is 1, which is consistent with the use of the ADAM algorithm directly. The other participants were updated based on their cosine similarity to the extracted training participants.

Updating the global model alone to improve the average goal of all participants does not provide optimum results. To obtain superior performance, we should be specific with the local model. In the proposed personalized update algorithm, personal information was added without local data leakage by extracting model gradient matrices from participants in the evaluation network. Next, the *Environment Similarity* parameter is dynamically calculated as the balanced and personalized factor to optimize the models. Thus, each participant can obtain an accurate personalized local model through a secure method.

The FL interaction can be described as a process of lifelong learning. The continuous joining of new participants causes sample accumulation and facilitates the learning of unknown local knowledge. However, indeterminate number of participants increase the fluctuation of the FL system. Furthermore, the model cannot be converged for direct use. Although our similarity algorithm alleviated this problem to a certain extent, to increase system stability and improve sustainably, a model parameter replacement policy is proposed for lifelong learning, as detailed in the next section.

C. Optimizing Backtracking Parameters Replacement Policy

The model parameter update determines the overall direction of model iteration. If the parameter update of the model is performed unconditionally, the FL performance with diverse experience aggregation fluctuates considerably, which affects the stability of the system. In FL, experience sharing can be achieved through a certain number of communications, which can be regarded as a series of continuous actions.

In the proposed model parameter replacement policy algorithm, the model parameters updated in each round are regarded as actions $a_t \in A$, the model performance as the status $s_t \in S$, the score after performing the action as the reward $R(s_t, a_t)$, the evaluation of the rewards within a period of time as retribution $U_{a_t}(s_t)$, and the selection of action for best retribution as policy P. The objective of this study was to achieve system stability and development. The key mechanism of the proposed policy is described in the following subsections.

1) State of Action: As the decision-making part in FL, the cloud server considers the convergence of the central model and average performance of local models. The state of the global part S_t^g is defined as the set of the current model and T round history model's performance. We choose loss results of global model $Loss_t^g$ and average accuracy of all participants local models $AvgAcc_t^l$ to represent the t round of global model which can be describe as follows:

$$s_0^g = Loss_t^g, \quad s_1^g = (Loss_{t-1}^g, AvgAcc_{t-1}^l), \ldots$$
$$s_T^g = (Loss_{t-T}^g, AvgAcc_{t-T}^l), \quad S_t^g = \{s_0^g, s_1^g, \ldots, s_T^g\} \quad (9)$$

As the client in FL, the participants evaluate the direct performance of local models. Considering the computing power of the participants and waiting time limitation, we define the state of the local part $S_t^{l_i} (i \in N)$ as the set of the current model and last round model's accuracy performance $Acc_t^{l_i} (i \in N)$ as follows:

$$s_0^{l_i} = Acc_t^{l_i}, s_1^{l_i} = Acc_{t-1}^{l_i},$$
$$S_t^{l_i} = \{s_0^{l_i}, s_1^{l_i}\} \quad (10)$$

2) Score of Reward: Two types of value functions were defined for the global and local models to evaluate the reward of corresponding model parameters, which are defined as R_g and R_l respectively. To simplify the calculation of the evaluation metrics and easily distinguish the reward gained by corresponding model parameters, we define the value function for reward by using the scoring mode. The value function for the global model is described in Eqs. 11 and 12. The reward for global model R_g of current state s_0^g and corresponding action a_0^g is calculated as follows:

$$\begin{cases} R_g(s_0^g, a_0^g) = 1, & \text{if } Loss_t^g - Loss_{t-1}^g \le 0 \\ R_g(s_0^g, a_0^g) = -1, & \text{if } Loss_t^g - Loss_{t-1}^g > 0 \end{cases} \quad (11)$$

Algorithm 2 Local Personalized Update Algorithm

Input: $\eta, \mathcal{L}, B, \rho_1, \rho_2, \delta, w_t^g, w_{t-1}^{l_k}, \mathrm{D}_k, \mathrm{E}_t^{l_k}$;
Output: $w_t^{l_k}$
1: **Initialization:** Initialize the first and second moment variables by $s = 0, r = 0$; Split D_k into batches with size B; Initialize the environment similarity by $\alpha = \sum_{k=1}^{\mathrm{K}} \frac{n_k e_t^{l_k l_i}}{n \sum \mathrm{E}_t^{l_k}}$;
2: **repeat**
3: **For** *each batch of split data* **do**
4: Computes the gradient by $g \leftarrow \frac{1}{B} \triangle_\theta w_t^g \mathcal{L}$;
5: $t \leftarrow t + 1$;
6: Updates the biased first moment estimate by $s \leftarrow \rho_1 s + (1 - \rho_1) g$;
7: Updates the biased second moment estimate by $r \leftarrow \rho_2 r + (1 - \rho_1) g \odot g$;
8: Computes the bias-corrected first moment estimate by $\hat{s} \leftarrow \frac{s}{1-\rho_1{}^t}$;
9: Computes the bias-corrected second moment estimate by $\hat{r} \leftarrow \frac{r}{1-\rho_2{}^t}$;
10: Computes the update by $\triangle_\theta = -\eta\alpha \frac{\hat{s}}{\sqrt{\hat{r}+\delta}}$;
11: Updates the model parameters by $w_t^{l_k} \leftarrow w_t^{l_k} + \triangle_\theta$;
12: **until** *The loss function* $\mathcal{L}$ *converages.*
13: **return** $w_t^{l_k}$;

The reward for global model of history state $s_j^g (j \in (0, T])$ and corresponding action $a_j^g (j \in (0, T])$ is calculated as follows:

$$\begin{cases} R_g(s_j^g, a_j^g) = 1, & \text{if } Loss_{t-j}^g - Loss_{t-j-1}^g \leq 0 \text{ and} \\ & AvgAcc_{t-j}^l - AvgAcc_{t-j-1}^l \geq 0 \\ R_g(s_j^g, a_j^g) = 0, & \text{if } Loss_{t-j}^g - Loss_{t-j-1}^g \leq 0 \text{ or} \\ & AvgAcc_{t-j}^l - AvgAcc_{t-j-1}^l \geq 0 \\ R_g(s_j^g, a_j^g) = -1, & \text{if } Loss_{t-j}^g - Loss_{t-j-1}^g > 0 \text{ and} \\ & AvgAcc_{t-j}^l - AvgAcc_{t-j-1}^l < 0 \end{cases} \tag{12}$$

The value function for the local model is described in Eq. 13. The reward for local model R_l of current state $s_0^{l_i} (i \in N)$ and corresponding action $a_0^{l_i} (i \in N)$ is calculated as follows:

$$\begin{cases} R_l(s_0^{l_i}, a_0^{l_i}) = 1, & \text{if } Acc_t^{l_i} - Acc_{t-1}^{l_i} \geq 0 \\ R_l(s_0^{l_i}, a_0^{l_i}) = -1, & \text{if } Acc_t^{l_i} - Acc_{t-1}^{l_i} < 0 \end{cases} \tag{13}$$

3) Evaluation of Retribution: The accumulation of the reward is defined as the retribution after performing the action. Here, $\gamma \in (0, 1)$ is the discount factor, which decreases as index τ increases. Through the discount factor, we regard the history action, which requires less time to obtain better performance and is more valuable than a future action. The evaluation of accumulated global and local retributions is described as Eq. 14. For state $s_j (j \in [0, T])$, we have the following equation:

$$U_{a_j}(s_0, \ldots, s_j) = \sum_{\tau=0}^{j} \gamma^{\tau} R(s_{j-\tau}, a_{j-\tau}) \tag{14}$$

According to Equation (14), we assume that the retribution of the state s_t of iteration round t is U_{a_t} and $U_{a_{t+1}}$ for iteration round $t+1$. $U_{a_t} = U_{a_{t+1}}$ represents the minimum requirement of policy convergence. As long as the process of iterate each state in the same order to calculate the retribution, we can get there must be new retribution $\gamma^{t+1} R(s_{t+1}, a_{t+1}) = 0$. Therefore, the optimizing backtracking parameters replacement policy only execute when the retribution convergence. We set the round of history action for policy to optimize is limited and the discount factor would give discounts on past updates. If the $U_{a_t} = U_{a_{t+1}}$, the latest parameter is preferentially selected because of the principle of strategy improvement.

4) Overall Policy: In the proposed *Optimizing Back Tracking Parameters Replacement Policy*, we select the model parameter from replay memory and current update according to the best accumulation of retribution. The general process for both global and local models of *Optimizing BackTracking Parameters Replacement Policy* can be described as Algorithm 3.

The proposed optimizing backtracking replacement policy is deployed in the evaluation network of the EEFED framework. Through a continuous evaluation mechanism of the global and local models in each round, the model parameters with the best retribution are selected for replacement to promote the sustainable stability of the system. In the next section, we discuss the experiments conducted on the proposed method.

Algorithm 3 *Optimizing BackTracking Parameters Replacement Policy*

Input: Model parameter set of previous T round of models $\{w_t - 1, \ldots w_t - T\}$ and current model w_t'; Test result feedback from evaluation network of previous T round of models and the current model; Retribution of previous T round of model set $\{U_{a_1}, \ldots U_{a_T}\}$.

Output: Optimized w_t for $t - th$ round

```
Obtain the state of the current action and previous T round action by using (9) for the global model, and using (10) for the local model;
Calculate the reward of the current and previous states by using (11) and (12) for the global model, and using (13) for the local model;
Calculate the retribution of current state by using (14);
Obtain the retribution of the previous state;
if U_{a_0} = Max(U_{a_0}, ..., U_{a_T})
    w_t = w_t'
else
    x = index of(Max(U_{a_0}, ..., U_{a_T}))
    w_t = w_x
return w_t
```

VI. Experiment and Evaluation

We conducted experiments to evaluate EEFED performance. We mainly discuss the following problems: the effectiveness and robustness of EEFED with imbalanced data distribution, the advantages of FL methods over conventional local methods in detecting local unknown attacks, and the feasibility of balancing consumption and gain.

We used two CPS datasets and a benchmark cyber dataset for the experiments. First, we propose a non-i.i.d imbalanced data distribution mode with no overlapping of attack categories for cyber traffic scenarios. Based on this data distribution mode, we compared the performance of the proposed EEFED with some state-of-the-art studies including FEDAVG [17], FEDFUSION [23], and the model proposed by Jiang [27]. Furthermore, we propose a novel scenario with an additional participant to prove the robustness of proposed method. Next, we compared the EEFED method with the conventional local method to prove the ability of detecting unknown attacks. Finally, we evaluated the proposed EEFED with more system consumption metrics.

A. Experimental Settings

1) Environmental Setup: In this study, the deep learning convolutional neural network (CNN) was used for global and local models. The details of CNN model structures are shown in Table II. The structure of the models was adjusted with various datasets, but the same structure was maintained between comparison methods. The CNN model and FL framework

TABLE II
DETAILS OF CNN MODEL STRUCTURES

Dataset	CNN	Input	Operator		Pool		Output
			Size	Stride	Size	Stride	
SWAT	Conv1	7×7×1	3	2	2	1	3×3×32
	Conv2	3×3×32	3	1	2	1	2×2×64
WADI	Conv1	11×11×1	3	2	2	1	5×5×32
	Conv2	5×5×32	3	2	2	1	2×2×64
KDD	Conv1	6×6×1	2	1	2	1	4×4×32
	Conv2	4×4×32	2	1	2	1	2×2×64

TABLE III
COMPARISON ACCURACY RESULTS OF EEFED AND EEFED-P ON THREE DATASETS

Round	SWaT		WADI		NSL-KDD	
	EEFED	EEFED-P	EEFED	EEFED-P	EEFED	EEFED-P
10	**80.87%**	78.10%	**67.76%**	67.74%	**86.17%**	85.01%
20	**91.30%**	90.52%	**91.24%**	91.11%	**93.68%**	91.98%
30	**94.83%**	93.39%	**95.48%**	95.45%	**92.92%**	91.81%
40	**95.61%**	93.71%	**95.86%**	95.37%	**92.77%**	91.99%
50	**95.66%**	94.02%	**97.79%**	97.37%	**93.17%**	93.04%

were implemented with Pytorch. The experiments were conducted on a Windows 10 platform with an NVDIA GeForce RTX 3070TI GPU.

2) Datasets Description and Partitioning: We conducted experiments on two CPS datasets and a cyber dataset. In the first CPS dataset, SWaT (a water treatment testbed) [34] is collected in a six-segment safe water treatment test bed, which represents a scaled-down version of a real-world industrial water treatment plant. SWaT contains one category of cyber data exists under normal operations and 36 categories exist under various cyberattacks. In the second CPS dataset, WADI (a water supply testbed) [35] designed by the same designer of SWaT but with different attack simulation methods, different treatment activity segments and different sensor data collection methods. WADI contains one category of cyber data exists under normal operations and 13 categories under various cyberattacks. Therefore, conducting experiments on two CPS datasets proved the robustness and effectiveness of the proposed method in CPS scenarios. Because a real industrial scenario not only confronts ICS attacks but also conventional cyberattacks, the dataset including TCP/IP traffic should be considered. In the TCP/IP cyber dataset NSL-KDD [36] extracted from a real LAN, 22 attack types exist in the dataset except the normal traffic, which can be categorized into four broad categories, namely DoS, R2L, Probe, and U2R. Notably, the experiments on three datasets are independent to demonstrate the robustness of our method.

In our experiments, 60% of the dataset was for local training, 20% for local scenario testing, and 20% for global scenario testing. The local training and testing part of datasets were further distributed to participants without sample overlapping. Each participant exhibits several attack categories that differ from other participants to simulate the non-i.i.d distribution on data size and category. Notably, the global scenario was tested on the same testing data, but the local scenario was tested on testing data distributed to each participant.

3) Baseline and Comparison Methods: In this manuscript, we compare the performance of the proposed EEFED with some state-of-the-art studies incorporating FL. To obtain superior performance in non-i.i.d data distribution scenarios, a series of FL algorithms have been proposed. The most advanced algorithm is FEDAVG [17], which has been demonstrated as basic FL [15], [19], [20], [22]. Furthermore, Yao et al. [23] proposed a feature fusion method as FEDFUSION aggregating the features from both the local and global models to achieve a higher accuracy at a lesser communication cost. To achieve faster convergence and obtain personalized model, Jiang et al. [27] proposed a meta learning method as METAFL to personalize the global model for individual participants. Further, Huang et al. [37] proposed a FL method as FEDAMP employing attentive message passing to facilitate similar participants to collaborate more. We improved the basic FL framework, personalized the update algorithm, and added a parameter replacement policy in the proposed EEFED. Next, we compared the performance of the proposed EEFED with aforementioned state-of-the-art studies and other recent FL studies [38], [39], [40]. All the comparison methods and our method are implemented with same deep learning model structure.

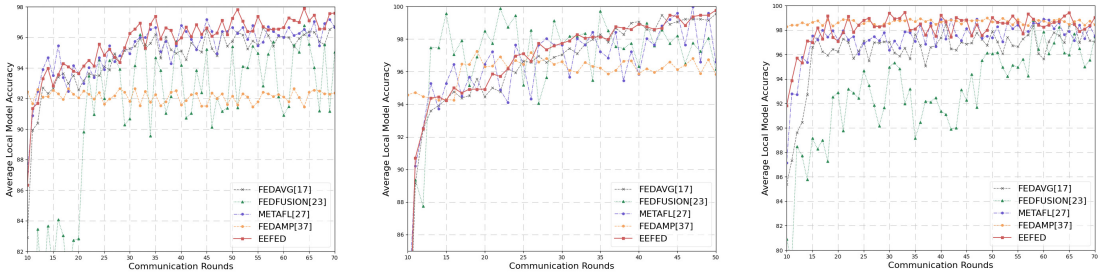
B. Performance Evaluation

In multiple experiments, we used average accuracy as the main metric for evaluating the effectiveness of methods because we assumed that all categories of misclassification cost the same. To this end, we plotted accuracy curves to indicate whether overfitting exists.

To simulate non-i.i.d imbalanced data distribution conveniently, the number of participants K is taken as five. Based on the previous works on FL [13], [14], [15], [16], local training rounds cause a certain influence on accuracy. We set the number of local training rounds as 5, 10 and 20 for the preliminary experiments. While adjusting the FL parameters, if the number of local training rounds is higher, a better performance can be achieved in fewer communication rounds. However, when local training rounds is 20 is almost the same as 10. Therefore, based on the consideration of model consumption, we set local training rounds as 10 in the following experiment. Thus, we have five local training sets $D_k(k \in [1, 5])$ for local training, and five local testing sets, defined as $LD_k(k \in [1, 5])$, for testing local scenarios, and one global testing set, defined as GD, for testing increasingly complex cyber scenarios.

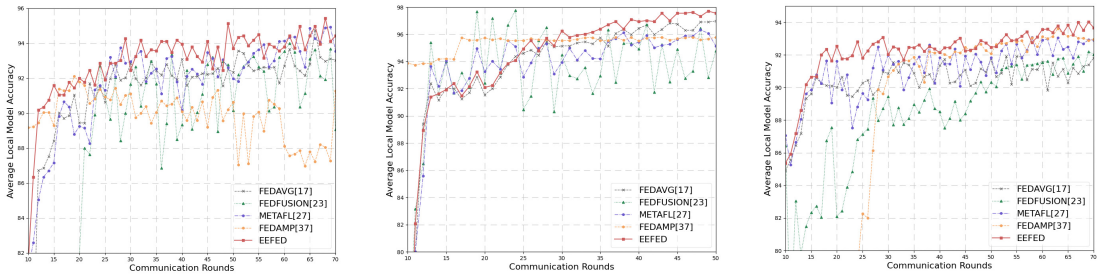
1) Effectiveness of Personalization: The experiments are set up to test whether the existence of personalized local models is effective to improve the performance. We conducted preliminary experiments to compare the performance of proposed method EEFED with EEFED removing personalized local models. EEFED removing personalized local models is abbreviated to EEFED-P. We tested the global model using global testing set GD. The accuracy results are shown in the Table III.

As we can see in the Table III, the performance of EEFED is always higher than EEFED-P in 50 rounds of communication. This shows the superiority of the existence of personalized



(a) Test on the SWaT dataset. (b) Test on the WADI dataset. (c) Test on the NSL-KDD dataset.

Fig. 7. Local scenario test results. Average accuracy of five participants personalized model tested on three datasets of local test data $LD_k (k \in [1, 5])$.



(a) Local model test on the SWaT dataset. (b) Local model test on the WADI dataset. (c) Local model test on the NSL-KDD dataset.

Fig. 8. Global scenario test results of local model: (a), (b), and (c) display the average accuracy of five participants personalized models tested on three datasets of global test data *GD*.

models and personalized algorithm. In subsequent experiments, we will compare EEFED with other methods in different scenarios.

2) Local Scenario Test: The experiments are set up to test whether local models personalized from EEFED can confront with common local cyber scenarios. We tested each participant's local models using the test set of the participants $LD_k (k \in [1, 5])$. The accuracy curves of three datasets are displayed in Figs. 7 (a), (b), and (c) respectively.

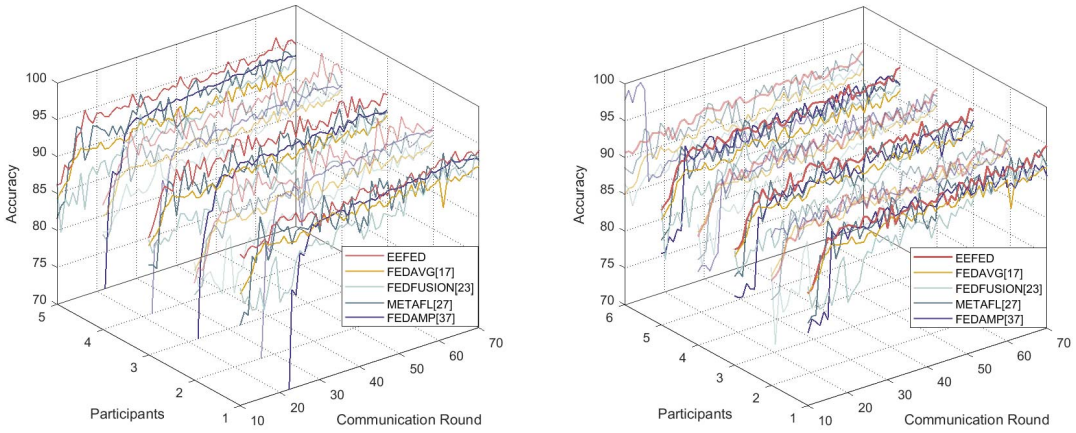
The accuracy curves denote the average local accuracy of the personalized models of five participants. The proposed EEFED achieves higher accuracy in most rounds of communication and more stable performance than other methods. The proposed EEFED exhibits robustness on various datasets, especially in the test on the NSL-KDD dataset. As displayed in Fig. 7 (b), the feature fusion method can be used to extract more information on data with more features in the preliminary round. However, with excessive information fusion, the ability of the personalized model is affected. As displayed in Fig. 7 (c), though the FEDAMP method has achieved stable, rapid and fair performance especially in NSL-KDD dataset, the effectiveness of the attention mechanism approach depends largely on model structure, which is not applicable to all participants and all datasets. Thus, EEFED generally outperformed other methods and was effective, robust, and stable in common local scenarios in different datasets.

3) Global Scenario Test: After proving that EEFED is stable and robust without influencing the performance in the local scenario, experiments were performed to test whether local models personalized from EEFED can confront with complex global scenarios. Furthermore, the performance of the global cloud model for the direct use of participant with limited samples or computation ability was investigated. We tested each participant local and global cloud models using the global scenario testing data *GD*. The accuracy curves of local models on three datasets are displayed in Figs. 8 (a), (b), and (c). The accuracy of the global cloud model on three datasets are displayed in Table IV.

TABLE IV
COMPARISON ACCURACY RESULTS OF GLOBAL MODEL ON THREE DATASETS

Methods	SWaT	WADI	NSL-KDD
FEDAVG[17]	92.97%	94.21%	91.70%
FEDFUSION[23]	93.50%	96.65%	93.94%
METAFL[27]	93.87%	95.99%	91.83%
FEDAMP[37]	89.53%	94.31%	93.82%
Baghersalimi's[38]	87.64%	96.87%	84.24%
PAIN-FL[39]	84.83%	94.29%	90.11%
PFedAtt[40]	95.16%	95.33%	89.88%
EEFED	**96.80%**	**97.79%**	**94.19%**
Ideal Model	98.18%	99.23%	97.02%

The results of the personalized local model reveal that EEFED exhibits higher accuracy, stability, and robustness than other comparison methods in the complex global scenario. As displayed in Figs. 8 (a), (b), and (c), EEFED exhibited a steady improvement state in approximately 10–15 rounds of communication and achieved the fastest model convergence compared with other comparison methods. Particularly distinct from the test on the SWaT dataset in Fig. 8 (a), the average local accuracy of EEFED was 90.19% in the 12th round of communication and became stable which is higher than other comparison methods. EEFED exhibited a superior average



(a) Former five participants global scenario.

(b) Additional special participant global scenario.

Fig. 9. Comparison results of individual local model accuracy between the addition special participant scenario with six participants and scenario with five participants tested on the NSL-KDD dataset.

local accuracy at the end with limited communication rounds: 97.55% in the 50th round on WADI, which is 0.77% higher than the baseline FEDAVG, 94.42% in the 70th round on SWaT, which is 1.4% higher than the baseline FEDAVG, and 93.32% in the 70th round on NSL-KDD, which is 1.5% higher than the baseline FEDAVG. The average accuracy curves of the personalized local model of EEFED exhibit more stability with less drastic fluctuation. Especially in the test on NSL-KDD displayed in Fig. 8 (c), the average local accuracy curve of EEFED revealed higher performance than other comparison methods.

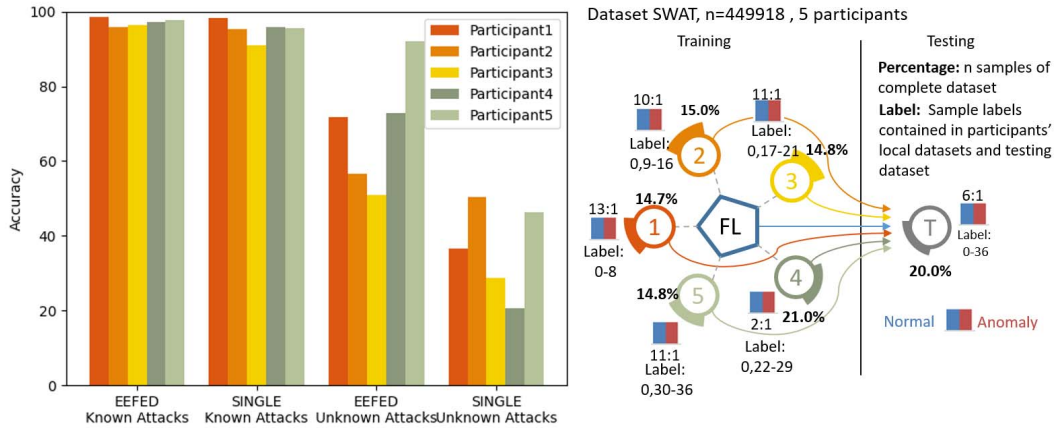
In addition to the aforementioned experiments, we conducted experiments to evaluate the performance of each participant-built global FL model using distributed data resources as well as the performance of an ideal central data model built by a central entity using all the data resources. The larger the performance close to the ideal model, the more effective the method is. In order to display the comparison results more efficiently, we added more latest studies with only one global model setting. Table IV shows the accuracy results of the global FL model for participants with limited samples and low computation power for direct use. And the accuracy results of ideal central data model are also included. The EEFED global model proved effective and stable and thus suitable for direct use by participants. The proposed model achieved satisfactory performance compared with the ideal central data model.

In summary, because of the proposed personalized update algorithm in EEFED, the personalized local model exhibited superior performance both in common local and complex global scenarios. The improvement satisfied the personalized requirements in a secure manner without deteriorating the ability of detecting common local attacks. Because of the proposed *Optimizing Backtracking Parameters Replacement Policy*, both the personalized local model and cloud global model exhibited superior stability and sustainability. The improvement directly benefits participants with limited samples or low computing power.

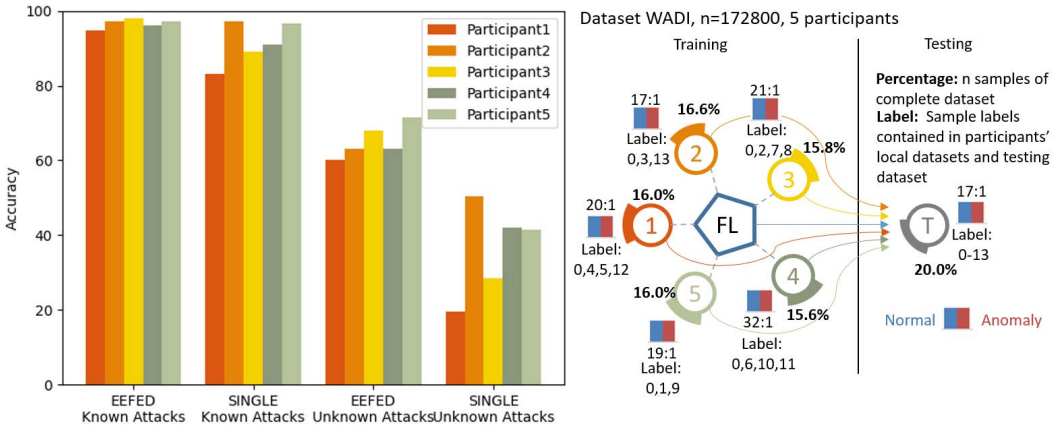
4) Additional Special Participant Scenario Test: To test whether EEFED can be used when indeterminate number of participants exists in the global scenario, an additional participant was added along with the existing five participants. For convenient data partition, we experimented on this scenario on the NSL-KDD dataset, which provides a test dataset with large distinction and completely unknown attacks. We distributed the test dataset to the additional participant to emphasize its specialty. The additional participant 6 with relatively sufficient data, can be regarded as type I of participants mentioned in section 3. The existing five participants can all be regarded as type II of participants in varying degrees. We compared the performance of individual personalized models between the former scenario with five participants and the additional scenario. The results are displayed in Fig. 9 (a) and Fig. 9 (b).

As displayed in Fig. 9 (a) and Fig. 9 (b), an additional participant considerably increased performance fluctuation. Some participants, such as participants 1 and 5, achieved improved performance, whereas some participants, such as participant 2, exhibited performance degradation. Thus, a tendency to continue to improve the performance appears. Notably, EEFED could still achieve satisfactory accuracy and stability compared with other methods.

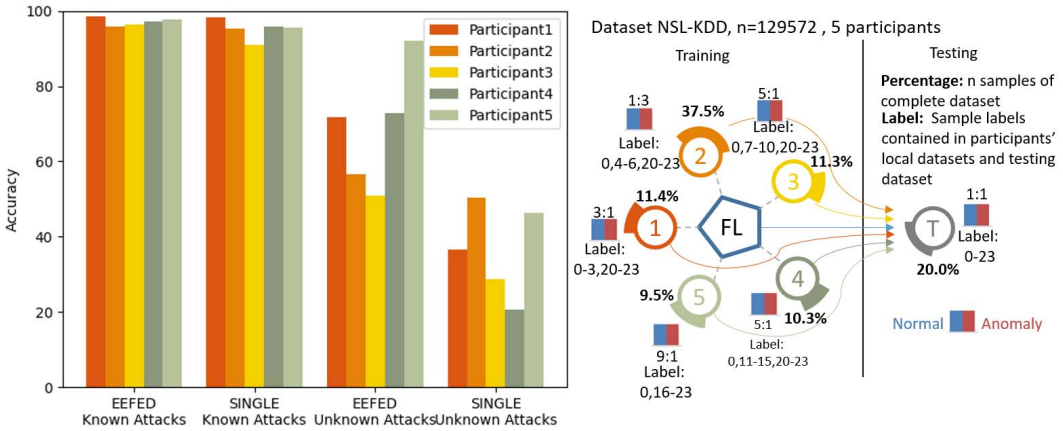
The addition of participants with large data difference in the FL has been a topic of research. Such local data are regarded to be low-quality data because of similarities with other participant data. This phenomenon is against the original intention of FL to share knowledge to secure local unknown knowledge. In this study, the proposed personalized update algorithm ensured each participant updated the local model but not completely abandoned it. FL is considered as a lifelong learning process. Therefore, a stable performance improvement provides possibility and time for a model to digest new knowledge. The proposed *Optimizing Backtracking Parameters Replacement Policy* strives for maintaining the stability of FL performance to a certain extent. The two proposed methods deployed in the dual network framework EEFED sync to form a virtuous circle. Next, we discuss



(a) Test on the SWaT dataset in the global scenario.



(b) Test on the WADI dataset in the global scenario.



(c) Test on the NSL-KDD dataset in the global scenario.

Fig. 10. Comparison results of EEFED and a single model of participant model accuracy of local known and local unknown attacks and the distribution of the datasets in each participant local dataset.

whether sharing knowledge through FL improves unknown attack detection performance.

5) Evaluation on Local Unknown Attacks: In complex cyber scenarios, the speed of collecting and labeling attack samples does not satisfy the requirement, making it difficult to detect local unknown attacks. We compared the performance of local models personalized from EEFED to a single local model built with distributed local data in term of detecting local unknown attacks. We extracted the fusion matrixes of the classification results of the global scenario test and single local model. The evaluation results are displayed in Figs. 10 (a), (b), and (c); the left part of each subfigure displays the accuracy of detecting known and unknown attacks for local participant, and the right part shows the details of the distribution of local participants' data. For example, Participant 1 in SWAT dataset test contains 7 attacks samples from segment 2, 3 and 4 with labels 1-7.

TABLE V
ACCURACY FOR THE EXPERIMENTS OF THE GLOBAL CLOUD MODEL IN THE GLOBAL SCENARIO

Methods	Accuracy of Final Round			Best Accuracy of All Round			Average Time of per Round(s)	Average parameters of per Round(B)
	SWaT	WADI	NSL-KDD	SWaT	WADI	NSL-KDD		
EEFED	**96.80%**	**97.79%**	**94.19%**	**96.80%**	98.17%	**95.42%**	83.24	1984
FEDAVG[17]	92.97%	96.21%	91.69%	91.70%	97.98%	91.95%	77.68	**1664**
FEDFUSION[23]	93.50%	96.65%	93.04%	94.50%	**98.73%**	93.94%	269.44	2496
METAFL[27]	93.87%	95.99%	91.83%	95.72%	98.17%	94.08%	81.63	2384
FEDAMP[38]	89.53%	94.31%	93.82%	92.12%	95.78%	93.67%	**71.17**	1856

And participant 2 in SWAT dataset test contains another 7 attacks samples from segment 3, 4, 5 and 6 with labels 9-15. Similarly, the attack samples of other participants are in different categories and occur in different segments of CPS industrial processes.

As displayed in Figs. 10 (a), (b), and (c), the unsatisfactory performance of detecting local unknown attacks adversely influences the performance of the single local model. The local personalized models obtained by EEFED to detect local unknown attacks outperformed the local single model. The EEFED improvement in unknown attack accuracy depends on the local data size, distribution, categories, and similarity with most participant's local data. For example, Participant 5 tested on the NSL-KDD dataset exhibited the largest improvement compared with the local single model, with an increase of 45.14% compared with other participants. It can be seen that participants with the smallest quantity of local data and the attack samples can greatly improve their detection ability of unknown attacks through EEFED. In contrast, participant 2 has the least effect. Probably because participant 2 exhibited the most similar quantity of local dataset attack categories to the global scenario and participant 2 had all Neptune samples under the DoS category, which is the main category of DoS attacks. A previous study [10] revealed that DoS attacks exhibited the highest success rate of transfer detection against Probe and R2L attacks. Therefore, participants having sufficiently high transferable attack classification experience and sufficiently more quantity of data are most likely to achieve an acceptable results of unknown attacks with a single local model.

Fig. 10 reveals that EEFED slightly improves the accuracy of known attacks in the global data test. Because of FL breaks resulting in $Isolated\ Data\ Island$ in complex cyber scenarios, EEFED also provides more experience with the varied types of local known attacks.

6) Evaluation of Performance and Consumption: In this section, we used the previously obtained result of the global scenario to evaluate the system performance of the four FL methods. The balance between performance and consumption was evaluated. The results are summarized in Table V. EEFED increases the calculation and transmission process of $Environment\ Similarity$, which revealed higher time consumption and storage space than that of baseline FEDAVG [17]. FEDFUSION [23] as a feature fusion method requires transferring the extracted features, training the fusion operator, and increasing the waiting time of the frozen model. Jiang's METAFL [27] method based on meta learning required more time to select the personalized mode but fine-tuned the global model more precisely than FEDAVG did. FEDAMP [37] method employs federated attentive message passing which increases a bit the calculation and transmission process as well.

TABLE VI
COMPUTATIONAL ANALYSIS ON THREE DATASETS OF EEFED IN PER COMMUNICATION ROUND

Dataset	General FL		EEFED	
	Operations	Parameters	Operations	Parameters
SWAT	11,065K	4,968K	11,229K	5,132K
WADI	58,311K	16,494K	58,478K	16,658K
KDD	5,116K	3,789K	5,260K	3,933K

As presented in Table V, EEFED achieved higher accuracy in the limited communication round than the other four methods. FEDFUSION achieved satisfactory performance on the WADI dataset, the time consumption was higher than those of other four methods. In EEFED, an evaluation network was added to evaluate the reward of update parameters and provide feedback to the execution network. FEDAMP achieved fast and acceptable performance on local personalized models. But its effect depends heavily on the participants' model and its global model perform less well in CPS detection scenarios. Although EEFED increased the transmission and calculation time of limited parameters, it reduced the communication rounds under the condition of achieving a certain accuracy. With a small increase in parameter transmission and elapsed time of $Execution\&Evalutaion$, EEFED achieved better performance with fewer rounds of communication. Therefore, for the total consumption of the system, EEFED exhibited the best comprehensive performance.

7) Analysis of Computational Complexity of Proposed Method: In this section, we evaluate the computational complexity of our proposed method. We use the number of computational operations and generated parameters to analyze the complexity. As shown in Table VI, the computational operations and generated parameters of CNN model are determined by the number of active training participants in each round and model structure. The number of operations and parameters for the execution network is determined by the model structure and the number of participants updating the model. Therefore, these two kinds of operations and parameters are the same as the basic FL method. However, in order to improve the performance of FL, we added the evaluation network. The computational operation and parameters increased by the evaluation network mainly depends on the calculation of Environmental Similarity and the calculation and comparison in the parameter backtracking replacement policy. As shown

in Table VI, EEFED increased 1.48%,0.28% and 2.81% of the computation operations compared to basis FL in the three datasets respectively and increased 3.30%,0.99% and 3.80% of the parameters in the three datasets respectively. It can be shown that EEFED achieves more efficient results at the cost of a small increase in the amount of computational operations.

VII. Conclusion

A novel FL framework, EEFED, was proposed for developing secure intrusion detection models collaboratively. To secure local private data, EEFED constructs the global model for participants who have limited samples or low computing ability. The proposed personalized update algorithm was deployed in EEFED to personalize the local model for each participant. Participants could detect local unknown cyberattacks. Furthermore, the personalized local model from EEFED exhibited superior performance when confronted with a common local scenario. The proposed update algorithm alleviated the non-i.i.d. statistical unbalanced data distribution challenges inherent in FL. Furthermore, personalized models can be adapted to individual participant scenarios. To achieve the sustainable stability of the system, the proposed optimizing backtracking parameters replacement policy cooperated with the dual $Execution\&Evaluation$ network framework. The proposed replacement policy could be used to adjust model parameters asynchronously based on the evaluation results obtained in the idle time of the participants. Thus, the model performance improved steadily and sustainably. Furthermore, our experiments proved that EEFED exhibited higher speed, stability, and effectiveness than baseline FEDAVG. We determined that FL exhibited effective performance for local unknown knowledge. With the advantages of solving $Isolated\ Data\ Island$ problem and obtaining mutual benefit, FL can be applied in more key areas in the future. In a future study, we aim to obtain privacy protection based on the attacks from transmission of FL to cloud server and optimization of privacy preserving methods.

References

[1] X. Xu, "From cloud computing to cloud manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 1, pp. 75–86, Feb. 2012.

[2] X. Vincent Wang and X. W. Xu, "An interoperable solution for cloud manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 29, no. 4, pp. 232–247, Aug. 2013.

[3] C. Chen, J. Y. An, N. Lu, Y. Wang, X. Yang, and X. Guan, "Typical characteristics, technologies and applications of cloud manufacturing," *Comput. Integr. Manuf. Syst.*, vol. 18, no. 7, pp. 1345–1357, 2012.

[4] J. Wan, M. Chen, F. Xia, L. Di, and K. Zhou, "From machine-to-machine communications towards cyber-physical systems," *Comput. Sci. Inf. Syst.*, vol. 10, no. 3, pp. 1105–1128, 2013.

[5] (Jun. 8, 2021). *The Colonial Pipeline Ransomeware Cyberattack: How a Major Oil Pipeline Got Held for Ransom [EB/OL]* [Online]. Available: https://www.vox.com/platform/amp/recode/22428774/ransomeware-pipeline-colonial-darkside gas-prices

[6] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, *Guide to Industrial Control Systems (ICS) Security*, document NIST-800-82 (R2), U.S. Department of Commerce, May 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

[7] W. Liang, K.-C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2063–2071, Mar. 2020.

[8] P. F. de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. G. Santos, D. Macedo, and C. Zanchettin, "Intrusion detection for cyber–physical systems using generative adversarial networks in fog environment," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6247–6256, Apr. 2021.

[9] Y. Lai, J. Zhang, and Z. Liu, "Industrial anomaly detection and attack classification method based on convolutional neural network," *Secur. Commun. Netw.*, vol. 2019, pp. 1–11, Sep. 2019.

[10] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," *EURASIP J. Inf. Secur.*, vol. 2019, no. 1, pp. 1–13, Dec. 2019.

[11] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Mar. 2019.

[12] T. T. Huong et al., "Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach," *Comput. Ind.*, vol. 132, Nov. 2021, Art. no. 103509.

[13] 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.

[14] S. Chatterjee and M. K. Hanawal, "Federated learning for intrusion detection in IoT security: A hybrid ensemble approach," 2021, *arXiv:2106.15349*.

[15] T. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*. Dallas, TX, USA, Jul. 2019, pp. 756–767.

[16] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," *Appl. Sci.*, vol. 8, no. 12, p. 2663, Dec. 2018.

[17] H. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2016, pp. 1273–1282.

[18] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

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

[20] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "FedHome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2818–2832, Aug. 2022.

[21] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," 2021, *arXiv:2102.02079*.

[22] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.I.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[23] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun, "Towards faster and better federated learning: A feature fusion approach," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 175–179.

[24] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," 2020, *arXiv:2002.04758*.

[25] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," 2020, *arXiv:2002.05516*.

[26] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," 2020, *arXiv:2002.10619*.

[27] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," 2019, *arXiv:1909.12488*.

[28] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, "Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3144–3159, Oct. 2021.

[29] J. Chen, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," in *Proc. ICLR Workshop Track*, 2016, pp. 1–10.

[30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[31] Z. H. Mahmood and M. K. Ibrahem, "New fully homomorphic encryption scheme based on multistage partial homomorphic encryption applied in cloud computing," in *Proc. 1st Annu. Int. Conf. Inf. Sci. (AiCIS)*, Nov. 2018, pp. 182–186.

[32] S. Lin, G. Yang, and J. Zhang, "Real-time edge intelligence in the making: A collaborative learning framework via federated meta-learning," 2020, *arXiv:2001.03229*.
[33] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 374–380.
[34] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *Proc. Int. Workshop Cyber-Physical Syst. Smart Water Netw. (CySWater)*, Apr. 2016, pp. 31–36, doi: 10.1109/CySWater.2016.7469060.
[35] C. Feng, V. R. Palleti, A. Mathur, and D. Chana, "A systematic framework to generate invariants for anomaly detection in industrial control systems," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–22, doi: 10.14722/ndss.2019.23265.
[36] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[37] Y. Huang et al., "Personalized cross-silo federated learning on non-IID data," in *Proc. Assoc. Advancement Artif. Intell.*, 2021, pp. 7865–7873.
[38] S. Baghersalimi, T. Teijeiro, D. Atienza, and A. Aminifar, "Personalized real-time federated learning for epileptic seizure detection," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 2, pp. 898–909, Feb. 2022.
[39] P. Sun et al., "Pain-FL: Personalized privacy-preserving incentive for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3805–3820, Dec. 2021.
[40] Z. Ma, Y. Lu, W. Li, J, Yi, and S. Cui, "PFedAtt: Attention-based personalized federated learning on heterogeneous clients," in *Proc. Asian Conf. Mach. Learn.*, 2021, pp. 1253–1268.

Xianting Huang received the B.E. degree in information security from the Beijing University of Technology, Beijing, China, in 2020, where she is currently pursuing the M.S. degree in information security. Her research interests include intrusion detection and federated learning.

Jing Liu (Member, IEEE) received the Ph.D. degree from the Beijing University of Technology in 2017. She is currently with the Faculty of Information Technology, Beijing University of Technology, as a Lecturer. Her research interests cover network security, ICS security, edge computing, and trusted computing. She participated in the development of a number of high-level prototype systems, such as the software-defined network terminal access prototype system based on trusted computing and the industrial internet platform security reference model. She has/had over 30 papers published in various international journals and conferences. She is a reviewer of several international journals and conferences. She is a member of the China Computer Federation.

Yingxu Lai (Member, IEEE) received the Ph.D. degree from the Chinese Academy of Sciences in 2003. She joined the College of Computer Science, Beijing University of Technology, in 2003. She was a Visiting Scholar at Arizona State University from 2013 to 2014. She is currently a Full Professor. Her research interests cover cloud computing, network security, edge computing, and trusted computing. She has/had over 70 papers published in various international journals and conferences. She is currently an Editorial Board Member of *Simulation Modelling Practice and Theory* and an Associate Editor of the *Journal of Artificial Intelligence and Technology*.

Beifeng Mao received the B.E. degree in information security from the Beijing University of Technology, Beijing, China, in 2019, where he is currently pursuing the M.S. degree in information security. His research interests include intrusion detection and deep learning.

Hongshuo Lyu is currently pursuing the M.S. degree in information security with the Beijing University of Technology. His research interests include intrusion detection and deep learning.