IEEE Open Journal of the
Industrial Electronics Society

Received 22 January 2024; revised 16 February 2024; accepted 23 February 2024. Date of publication 27 February 2024; date of current version 7 March 2024. The review of this article was arranged by Associate Editor Yang Shi.

Digital Object Identifier 10.1109/OJIES.2024.3370496

End-Edge Collaborative Lightweight Secure Federated Learning for Anomaly Detection of Wireless Industrial Control Systems

CHI XU [1,2,3] (Senior Member, IEEE), XINYI DU [1,2,3,4], LIN LI [1,3], XINCHUN LI [4], AND HAIBIN YU [1,2,3] (Senior Member, IEEE)

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China
[3]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
[4]School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China

CORRESPONDING AUTHOR: CHI XU (e-mail: xuchi@sia.cn)

This work was supported in part by National Natural Science Foundation of China under Grant 92267108 and Grant 62173322, and in part by Science and Technology Program of Liaoning Province under Grant 2023JH3/10200004 and Grant 2022JH25/10100005.

ABSTRACT With the wide applications of industrial wireless network technologies, the industrial control system (ICS) is evolving from wired and centralized to wireless and distributed, during which eavesdropping and attacking become serious problems. To guarantee the security of wireless and distributed ICS, this article establishes an end-edge collaborative lightweight secure federated learning (LSFL) architecture and proposes an LSFL anomaly detection strategy. Specifically, we first design a residual multihead self-attention convolutional neural network for local feature learning, where the variability and dependence of spatial-temporal features can be sufficiently evaluated. Then, to reduce the wireless communication cost for parameter exchange and edge federal learning, we propose a dynamic parameter pruning algorithm by evaluating the contribution of each parameter based on the information entropy gain. Furthermore, to ensure the parameter security during wireless transmission in the open radio environment, we propose an adaptive key generation algorithm for parameter encryption. Finally, the proposed strategy is experimentally validated on representative datasets, including Smart Meter, NSL-KDD, and UNSW-NB15. Experimental results demonstrate that the proposed strategy achieves 99% accuracy on different datasets, where at least 89.6% wireless communication cost is reduced and tampering/injecting attacks are defended.

INDEX TERMS Industrial control system (ICS), anomaly detection, federated learning, convolutional neural network (CNN).

I. INTRODUCTION

Industrial control system (ICS) plays an irreplaceable role in key fundamental infrastructures, such as manufacturing factories, power systems, and nuclear power plants [1]. However, ICSs are facing more and more serious security and privacy problems when they are interconnected by the open Internet. Thus, anomaly detection becomes critical in protecting the security of ICSs. By detecting abnormal behaviors, potential attacks can be detected in advance, further guaranteeing the reliability of ICSs.

Existing anomaly detection strategies are mainly based on machine learning algorithms including support vector machine, random forest, and decision tree [2]. Zhou et al. [3] proposed a variational long short-term memory (LSTM) learning network for intelligent anomaly detection based on reconstructed feature representation. Kaur et al. [4] proposed the Bayesian method for the convolutional neural network (CNN) integration, where Bayesian component was used to distinguish network physical intrusion from normal events in binary and multiclass events, while the CNN was used

to process high-dimensional feature space before intrusion classification task. Ahakonye et al. [5] proposed an agnostic Chi-square feature selection and prepruned decision tree for intrusion detection in SCADA systems. Chen et al. [6] proposed an adaptive method of information-enhanced countermeasure domain, and constructed a feature extractor through CNN and bidirectional LSTM architecture. These strategies are mainly oriented toward centralized ICSs and can achieve high-accuracy detection when there is enough training data. However, when a single node in the centralized ICS is attacked and fails, there will be serious privacy leakage and data security problems, which could significantly reduce the performance of anomaly detection and impact the security and reliability of ICS.

As a consequence, a distributed anomaly detection strategy is gaining popularity in fault tolerance, scalability, and privacy protection for distributed ICSs [7]. Rather than relying on a single centralized algorithm, the distributed strategy employs multiple interconnected nodes that cooperatively perform anomaly detection. Liu et al. [8] designed a CNN based on attention strategy and a LSTM network for distributed anomaly detection, where the gradient is compressed. Li et al. [9] proposed a distributed anomaly detection strategy based on the CNN and gated cyclic unit network, and designed a secure communication protocol based on the paillier cryptosystem to protect the security and privacy of network parameters. Huong et al. [10] proposed a hybrid network based on variational autoencoder and LSTM to deal with distributed anomaly detection of time-series data. Zhai et al. [11] proposed a distributed intrusion detection method based on the CNN and gated recurrent unit (GRU) under the federated learning architecture. Khan et al. [12] proposed a distributed intrusion detection system combined with simple cycle unit.

In particular, federated learning is emerging as a promising distributed strategy that has gained wide attention and applications in academia and industry [13], [14]. With federated learning, nodes only share parameters rather than raw data or intermediate results. Therefore, the risk of data leakage can be effectively reduced. However, most federated learning-based anomaly detection strategies simply transmit the raw parameters, and do not fully consider the features and importance of parameters. In this way, the communication cost for parameter exchange is still ignorable, especially for the wireless ICS whose communication resources are always limited. More importantly, when the parameters are wirelessly exchanged in an open radio environment, eavesdroppers and attackers are more easy to intrude and destruct the wireless ICS. If a trusted third party is employed, more communication costs will be introduced to the wireless ICS and further reduce the performance of anomaly detection. Thus, how to design an effective federal learning strategy with low communication cost and high security remains a hot topic. Motivated by this, this article proposes an end-edge collaborative lightweight secure federal learning (LSFL) anomaly detection strategy for the wireless ICS.

The major contributions are summarized as follows.

1) *A novel lightweight secure collaborative federal learning architecture:* We establish an end-edge collaborative LSFL architecture for the wireless ICS, where multiple end devices perform local feature learning and share parameters with an edge server for federal learning.
2) *Enhanced CNN for feature learning:* We design a residual multihead self-attention convolutional neural network (RMS-CNN) for spatial-temporal features learning. Herein, we employ multihead self-attention network to learn the dependence among features and use a residual connection to retain and integrate variability features at different layers. In this way, we can obtain more comprehensive information with fewer layers, and thus, enhance the detection capability.
3) *Dynamic parameter pruning for lightweight wireless communication:* We develop a dynamic parameters pruning algorithm based on information entropy gain. Herein, we evaluate the contribution of each parameter by calculating the information entropy gain, and dynamically set the pruning threshold. In this way, we can prune the parameters and reduce the wireless communication cost for parameter exchange.
4) *Adaptive key generation for secure parameter exchange:* We propose an adaptive key generation algorithm to encrypt the pruned parameters. Herein, end devices together with edge server dynamically generate different key pairs, adaptively adjust keys by multiple Hash operations and encrypt the pruned parameters by the advanced encryption standard (AES) algorithm. In this way, we ensure the parameter security during wireless transmission in open radio environment.
5) *Extensive experiments:* We perform extensive experiments on representative datasets including Smart Meter, NSL-KDD, and UNSW-NB15, and compare the proposed strategy with two benchmark strategies. The experimental results demonstrate that the proposed strategy achieves 99% accuracy on different datasets, while reducing at least 89.6% communication cost and ensuring the security.

The rest of this article is organized as follows. Section II presents the end-edge collaborative LSFL architecture. Section III specifies the LSFL anomaly detection strategy in detail. Section IV evaluates the performance of the proposed strategy by extensive experiments, and finally, Section V concludes this article.

II. END-EDGE COLLABORATIVE LSFL ARCHITECTURE

The ICS contains an edge server and I end devices denoted as ED_i ($i = 1, 2, \ldots, I$). The edge server, which has sufficient computation resources, is responsible for end-edge collaborative task scheduling and data training. End devices, which are task customized with limited computation resources, can be sensors, controllers, and actuators distributed along the production process. Each end device continuously collects local data, e.g., monitoring information and control command.

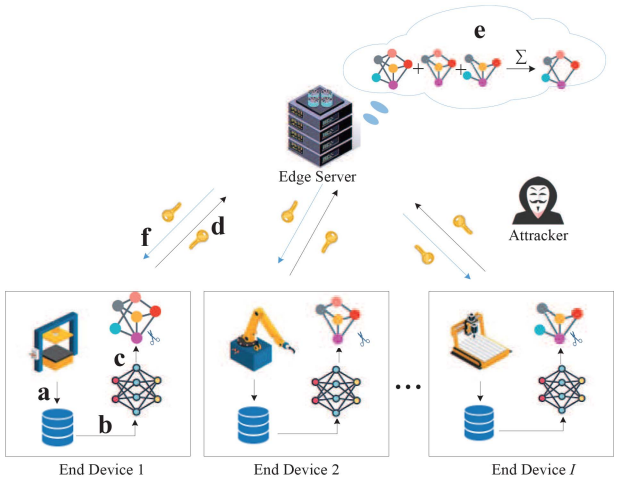


FIGURE 1. End-edge collaborative LSFL architecture for anomaly detection.

The edge server and end devices are wireless connected by a high-reliable and strong-realtime industrial wireless network such as industrial 5G [15]. Due to the limited spectrum resources, the data volume transmitted by the wireless ICS is limited. Moreover, the transmitted data maybe eavesdropped since the radio environment is open to everyone. In this way, attackers can send invalid requests and malicious data to the wireless ICS, and further interpolate the data to compromise end devices. This may cause serious problems, such as production interruptions and device failures preventing industrial production lines from running normally.

To ensure the security of the wireless ICS, we propose the end-edge collaborative LSFL architecture as Fig. 1. End devices first perform feature learning with the raw data collected locally, then prune the parameters to reduce the wireless communication cost, encrypt the pruned parameters for secure wireless communication, and finally, transmit the encrypted pruned parameters to edge server for federal learning. Specifically, the process of end-edge collaborative LSFL is given as follows.

A. RAW DATA COLLECTION

Each end device first collects raw data and form local dataset X_i $(i = 1, 2, \ldots, I)$ for training. Due to the limited computation resource, each end device collects the same length of sequence. Thus, the local dataset X_i with M sequences, where each sequence is with the same length and N features, is given as

$$X_i = \begin{pmatrix} x^i_{1,1} & x^i_{1,2} & \cdots & x^i_{1,N} \\ x^i_{2,1} & x^i_{2,2} & \cdots & x^i_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ x^i_{M,1} & x^i_{M,2} & \cdots & x^i_{M,N} \end{pmatrix} = \left[x^i_{m,n}\right]_{M\times N} \quad (1)$$

where $x^i_{m,n}$ denotes the nth feature of the mth sequence of the local data by ED_i.

B. PARALLEL FEATURE LEARNING

With the collected raw data, all end devices perform parallel training for feature learning by the following proposed RMS-CNN, where the input training data of ED_i is X_i. The initial training models of all end devices are the same, and the initial training parameter set of ED_i is denoted as Y_i $(i = 1, 2, \ldots, I)$. In detail, Y_i, which contains J parameters, is given as

$$Y_i = \begin{pmatrix} y_{i,1} & y_{i,2} & \cdots & y_{i,J} \end{pmatrix} \quad (2)$$

where $y_{i,j}$ denotes the jth initial training parameter of ED_i.

Then, according to the local training results, each end device updates Y_i into a new parameter set $\tilde{Y}_i$ $(i = 1, 2, \ldots, I)$, which also contains J parameters and is given as

$$\tilde{Y}_i = \begin{pmatrix} \tilde{y}_{i,1} & \tilde{y}_{i,2} & \cdots & \tilde{y}_{i,J} \end{pmatrix}. \quad (3)$$

By training, the parameter set of ED_i is updated as

$$Y_i \leftarrow \tilde{Y}_i. \quad (4)$$

C. PARAMETER PRUNING

To reduce the wireless communication cost for parameter exchange and realize lightweight federal learning, each end device further prunes the parameter by the following proposed dynamic parameter pruning algorithm. The parameter set Y_i is pruned as Z_i $(i = 1, 2, \ldots, I)$ with J parameters, which is given as

$$Z_i = \begin{pmatrix} z_{i,1} & z_{i,2} & \cdots & z_{i,J} \end{pmatrix} \quad (5)$$

where $z_{i,j}$ denotes the jth parameter.

D. PARAMETER ENCRYPTION AND SECURE TRANSMISSION

To avoid being eavesdropped and attacked during wireless communication in the open radio environment, all pruned parameters are encrypted for secure transmission. The encryption key for Z_i is denoted as K_i, which is generated by the following proposed adaptive key generation algorithm. Then, each end device encrypts the pruned parameters and transmits the encrypted parameter to edge server for federal learning.

E. PARAMETER AGGREGATION

With the encrypted pruned parameter sets from I end devices, the edge server decrypts each pruned parameter set and aggregates the pruned parameters to get a new parameter set with J parameters. Herein, the new parameter $\bar{z}_j$ is calculated as the weighted mean of the parameters uploaded by I end devices, namely

$$\bar{z}_j = \frac{\sum_{i=1}^{I} \alpha_{i,j} z_{i,j}}{I}, \; j = 1, \ldots, J \quad (6)$$

where $\alpha_{i,j} \in [0, 1]$ indicates the importance of the pruned parameter $z_{i,j}$.

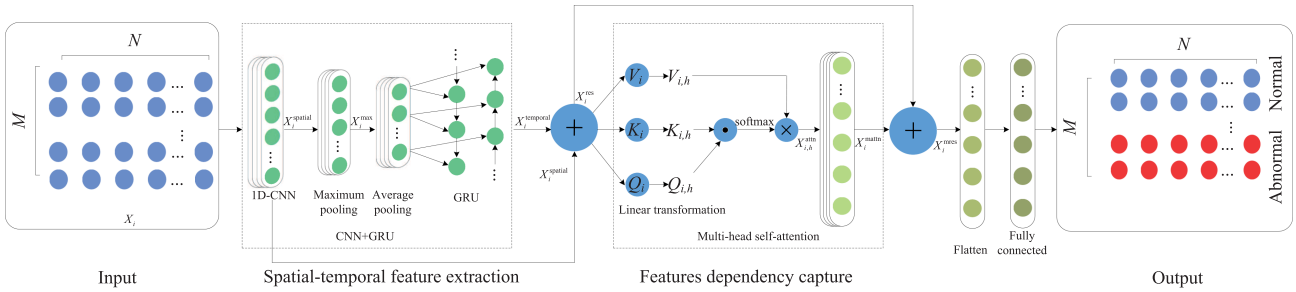


FIGURE 2. Proposed RMS-CNN structure.

In this way, we formulate a new parameter set

$$\bar{Z} = \begin{pmatrix} \bar{z}_1 & \bar{z}_2 & \cdots & \bar{z}_J \end{pmatrix}. \tag{7}$$

F. PARAMETER UPDATE

With the new parameter set $\bar{Z}$, the edge server sends it to all end devices, and each end device updates its parameter set for the next-round training, namely

$$Y_i \leftarrow \bar{Z}, i = 1, \ldots, I. \tag{8}$$

With multiround interactions for parameter exchange and training, we can obtain an accurate model for anomaly detection. After completing the offline anomaly detection training, the end devices are ready for online anomaly detection.

III. LSFL ANOMALY DETECTION STRATEGY

With the established end-edge collaborative LSFL architecture, we further specify the LSFL anomaly detection strategy in this section. In detail, we first design RMS-CNN for feature learning to enhance the capability of anomaly detection. Then, we propose the dynamic parameter pruning algorithm to reduce the wireless communication cost for parameter exchange. Finally, we propose the adaptive key generation algorithm for parameter encryption and secure transmission in the open radio environment.

A. RMS-CNN FOR LOCAL FEATURES LEARNING

To enhance feature learning capability for anomaly detection, we design RMS-CNN for adequately extracting, integrating, and capturing the variability and dependence of spatial-temporal features. The RMS-CNN structure is illustrated in Fig. 2.

1) SPATIAL-TEMPORAL FEATURE EXTRACTION

In the wireless ICS, both normal and abnormal protocol data are collected cyclically for training, most of which are spatial-temporal sequences. To fully extract the spatial features of sequences, we employ 1-D CNN (1D-CNN) to extract the local information according to the length of input data. Thus, we have

$$X_i^{\text{spatial}} = f\left(X_i * W_{i,l} + V_{i,l}\right) = \left[x_{m,n}^{i,\text{spatial}}\right]_{M\times N} \tag{9}$$

Herein, X_i and X_i^{spatial} are the input sequence and the output feature of the convolution layer, respectively; $f(\cdot)$ is the activation function; $*$ denotes the convolution operation; $W_{i,l}$ and $V_{i,l}$ are the weight factor and bias factor at the lth $(1 \le l \le L)$ convolution kernel in the convolution layer with a total of L convolution kernels.

After spatial feature extraction, the dimension of X_i^{spatial} should be very high. Thus, it is necessary to reduce the redundant information. We apply two pooling layers after 1D-CNN to reduce the feature dimensions while retaining important spatial feature information. Specifically, we divide each line of X_i^{spatial} into $\hat{N} = \lceil N/S \rceil$ parts, where $\lceil\rceil$ is the ceiling function. In this way, each part is with S features and denoted as $\left[x_{m,s}^{i,\text{spatial}}\right]_{1\times S}$. It is first input to the maximum pooling to obtain the maximum value of every S features, i.e.,

$$\begin{aligned} X_i^{\max} &= \left(\left(\max_{s=1,\ldots,S}\left[x_{m,s}^{i,\text{spatial}}\right]\right)_1, \ldots, \left(\max_{s=1,\ldots,S}\left[x_{m,s}^{i,\text{spatial}}\right]\right)_{\hat{N}}\right) \\ &= \left[x_{m,\hat{n}}^{i,\max}\right]_{M\times\hat{N}} \end{aligned} \tag{10}$$

where $x_{m,\hat{n}}^{i,\max}$ denotes the $\hat{n}$th feature of the mth data. In this way, a sequence with N dimensions is shortened to a new sequence with $\hat{N}$ dimension.

Then, $X_i^{\max}$ is input to the global average pooling to obtain the average value of features, i.e.,

$$X_i^{\text{avg}} = \frac{\sum_{\hat{n}=1}^{\hat{N}} x_{m,\hat{n}}^{i,\max}}{\hat{N}} = \left[x_{m,\hat{n}}^{i,\text{avg}}\right]_{M\times 1} \tag{11}$$

where $x_{m,\hat{n}}^{i,\text{avg}}$ denotes the $\hat{n}$th feature of the mth data. In this way, a sequence with $\hat{N}$ dimensions is shortened to that with only one dimension.

Furthermore, to learn the temporal features of X_i^{avg}, we apply GRU to extract long-memory dependencies with unique memory property, i.e.,

$$X_i^{\text{temporal}} = g\left(X_i^{\text{avg}}\right) \tag{12}$$

where $g(\cdot)$ denotes the activation function used in GRU.

2) FEATURE DEPENDENCE CAPTURE

In order to reduce information loss during spatial-temporal feature extraction, we further capture the dependence of spatial-temporal features by multihead self-attention. To maintain the relative consistency of the original spatial-temporal features, we first add a residual connection after the spatial-temporal feature extraction, wherein the residual connection integrates the features extracted by 1D-CNN and GRU. In this way, both the enhanced features and unmodified original input features provided by 1D-CNN and GRU are fully considered. Mathematically, the extracted features X_i^{spatial} and X_i^{temporal} from 1D-CNN and GRU are added together, i.e.,

$$X_i^{\text{res}} = X_i^{\text{spatial}} + X_i^{\text{temporal}}. \tag{13}$$

Then, we capture the dependence in X_i^{res} by calculating the importance of each location with respect to other locations. Specifically, we map the input features to different subspaces by multiple times linear transformations to obtain more different information. Furthermore, by calculating multiple attention heads and paying attention to different location information and dependencies in parallel, we can get more comprehensive and accurate global information.

Mathematically, X_i^{res} is linearly transformed by a learnable parameter matrix to obtain query vector Q_i, key vector K_i and value vector V_i. Each vector is further divided into H numbers of attention heads. Then, the self-attention of hth ($1 \le h \le H$) head is calculated as

$$X_{i,h}^{\text{attn}} = \text{softmax}\left(\frac{Q_{i,h} \bullet K_{i,h}^{\mathbf{T}}}{\sqrt{d}}\right) V_{i,h}, \quad h = 1, \ldots, H \tag{14}$$

where $Q_{i,h} = X_i^{\text{res}} U_{i,h}^{Q}$, $K_{i,h} = X_i^{\text{res}} U_{i,h}^{K}$, and $V_{i,h} = X_i^{\text{res}} U_{i,h}^{V}$ are the hth query vector, key vector, and value vector, respectively; $U_{i,h}^{Q}$, $U_{i,h}^{K}$, and $U_{i,h}^{V}$ are the learnable parameter matrix for linear transformation; softmax$(\cdot)$ is the normalization function; $\sqrt{d}$ is the dimension of $Q_{i,h}$ and $K_{i,h}$; and $\bullet$ is the dot product.

To get all features captured through multihead self-attention, we connect the output of each attention head to form a large matrix, which is then multiplied by the weight matrix for final linear transformation. That is

$$X_i^{\text{mattn}} = c\left(X_{i,1}^{\text{attn}}, X_{i,2}^{\text{attn}}, \ldots, X_{i,H}^{\text{attn}}\right) \cdot U^{O} \tag{15}$$

where $c(\cdot)$ is the connection function; and U^O is the learnable parameter matrix.

After calculating the multihead self-attention matrix, the output feature X_i^{mattn} together with the original X_i^{res} is again used to learn the enhanced features denoted as X_i^{mres}, i.e.,

$$X_i^{\text{mres}} = X_i^{\text{res}} + X_i^{\text{mattn}}. \tag{16}$$

Finally, we use the general flatten layer and fully connected layer to obtain the parameters and perform anomaly detection. Note that the initial or updated parameters are utilized throughout the aforementioned process for training.

B. DYNAMIC PARAMETER PRUNING FOR LIGHTWEIGHT FEDERAL LEARNING

As a large number of parameters are generated after RMS-CNN training, the wireless communication cost for parameter exchange of federal learning increases dramatically. However, the communication resources (e.g., bandwidth and transmit power) are very limited in the industrial wireless network, which certainly cannot support the massive parameter exchange frequently. Thus, we propose the dynamic parameter pruning algorithm to reduce the wireless ICS. Specifically, at the end of each training round, each end device dynamically prunes its parameters by fully considering the features and contributions of parameters.

1) PARAMETER CONTRIBUTION EVALUATION

To evaluate the contribution of the parameter after training, we propose to calculate the information entropy gain. First, we make the parameter $y_{i,j}$ discrete as

$$y_{i,j} = \begin{cases} 1, & y_{i,j} > 0 \\ 0, & y_{i,j} \le 0. \end{cases} \tag{17}$$

Then, we calculate the information entropy of each parameter as

$$e_{i,j} = -p(y_{i,j}) \log p(y_{i,j}) \tag{18}$$

where $p(y_{i,j})$ is the ratio of parameter $y_{i,j}$ to the total parameter set Y_i. Similarly, the parameter after training $\tilde{y}_{i,j}$ can also be discrete and the information entropy is calculated as $\tilde{e}_{i,j} = -p(\tilde{y}_{i,j}) \log p(\tilde{y}_{i,j})$.

Then, we calculate the information entropy gain of each parameter as

$$\Delta e_{i,j} = e_{i,j} - \tilde{e}_{i,j}. \tag{19}$$

With the information entropy gain, we can evaluate the contribution of each parameter, where the greater the information entropy gain, the higher the contribution of the parameter. In this way, we can enhance the training performance.

2) DYNAMIC PARAMETER THRESHOLD SETTING

To further measure the volatility of the parameters and ensure the reliability of the contribution, we further calculate the standard deviation of the information entropy gain as

$$\Phi_i = \sqrt{\frac{\sum_{j=1}^{J} (\Delta e_{i,j} - \bar{e}_i)^2}{J}} \tag{20}$$

where $\bar{e}_i = \frac{\sum_{j=1}^{J} \Delta e_{i,j}}{J}$ is the average value of information entropy gain.

As different parameters have different contributions, we set the parameter pruning threshold based on Φ_i. In this article, we mainly consider the parameters with respect to weight, bias, and gradient since they significantly impact the training performance. Specifically, the parameter pruning thresholds

are calculated as

$$\Psi_i^{\text{weight}} = \Phi_i \varphi \tag{21}$$

$$\Psi_i^{\text{bias}} = \Phi_i \zeta \tag{22}$$

$$\Psi_i^{\text{grad}} = \Phi_i \psi \tag{23}$$

where Ψ_i^{weight}, Ψ_i^{bias}, and Ψ_i^{grad} are the pruning thresholds with respect to weight, bias, and gradient. φ, ζ, and ψ are weight factor, bias factor, and gradient factor, respectively. In this way, we can select different thresholds for parameter pruning.

3) PARAMETER PRUNING AND RECONSTRUCTION

With the parameter pruning threshold calculated based on the information entropy gain, we prune the parameters dynamically. Specifically, according to the parameter characteristics with respect to weight, bias, and gradient, we prune the parameters according to (21)–(23), respectively. If a parameter is larger than the calculated threshold, the parameter is reserved; Otherwise, the parameter is pruned to be 0, namely it is inactive. In this way, we reduce the data volume of redundant parameters with low contributions.

However, after parameter pruning, the value range of parameters is compressed, which may influence the performance of federal learning. Thus, we further enlarge the value range of parameter. The scaling factor is defined as the ratio of the sum of the parameters' absolute values before and after pruning, namely

$$z_{i,j} \leftarrow \frac{\sum_{j=1}^{J} |y_{i,j}|}{\sum_{j=1}^{J} |z_{i,j}|} z_{i,j}. \tag{24}$$

Then, we multiply the parameters by (24) to enhance the sparsity of parameter's value range. In this way, we reconstruct the pruned parameters for federal learning.

C. ADAPTIVE KEY GENERATION FOR PARAMETER ENCRYPTION

When the pruned parameters are exchanging between end devices and edge server for federal learning, the risk of attack increases substantially in the open radio environment. Thus, to avoid being eavesdropped and attacked, we propose the adaptive key generation algorithm to encrypt parameters for secure wireless communication.

1) KEY NEGOTIATION

The key is generated by each communication pair, namely end device and edge server, and we do not use third party for the distributed ICS. Hence, in order to establish the secure transmission channel for each communication pair, we randomly generate key pairs for all communication pairs. Each key pair includes a private key K_i^{r} and a public key K_i^{u}. K_i^{r} is a random number, while K_i^{u} is generated based on K_i^{r} where K_i^{u} is a point on the elliptic curve $y^2 = x^3 + ax + b$ subjecting to $4a^3 + 27b^2 \neq 0$.

TABLE 1. Encryption Algorithm Comparison

Performance	Basic AES algorithm	Proposed algorithm
Key generation mode	Static	Dynamic
Key pair	Same	Different
Predictability	Deterministic	Random
Complexity	Low	Medium

Then, the communication pair retains the private key K_i^{r} and exchanges the public key K_i^{u} to each other to ensure that they are the only participants who can decrypt them. With the received K_i^{u}, the communication pair performs key negotiation. Specifically, ED_i and the edge server negotiate on their own K_i^{r} to obtain a shared key K_i^{s}, where $K_i^{\text{s}} = K_i^{\text{r}} K_i^{\text{u}}$ is calculated by the communication pair and should be the same. By key negotiation, the two participants over the same communication channel have the same key K_i^{s}, while those over different communication channels have different keys.

2) KEY CONVERSION

To ensure the security of parameter exchange, we employ the widely used AES algorithm to encrypt the pruned parameters. We first prepare the input key material K_i^{m} by adding a random number R_i to K_i^{s}, namely $K_i^{\text{m}} = c(K_i^{\text{s}}, R_i)$. However, K_i^{m} cannot be directly applied to encrypt the pruned parameters since the length of K_i^{m} is much longer than the length supported by the AES algorithm. Thus, we need to covert the format of K_i^{m} to make its length is supported by AES algorithm.

As Hash function maps the key with any length to that with a fixed length, we employ Hash function to convert the format of the key. Meanwhile, we can also enhance the security of the key as Hash function is one way, namely the output is unique and irreversible, and the attacker cannot obtain the original key by calculating the Hash value reversely. Furthermore, to increase the complexity of the key, we execute multiple times of Hash operations as

$$K_{i,T_i}^{\text{hash}} = \text{HASH}(K_i^{\text{m}}, T_i) \tag{25}$$

where $\text{HASH}(K_i^{\text{m}}, T_i)$ is the Hash function indicating T_i times Hash operation for K_i^{m}. T_i is calculated as

$$T_i = \left\lceil \frac{L_i}{L_i^{\text{hash}}} \right\rceil \tag{26}$$

where L_i is the length supported by the AES algorithm (i.e., 128 bits, 192 bits, or 256 bits), and L_i^{hash} is the output length by the Hash function.

Then, we connect the output of each Hash operation to obtain the key until the length is supported by the AES algorithm, i.e.,

$$K_i = c\left(K_{i,1}^{\text{hash}}, K_{i,2}^{\text{hash}}, \ldots, K_{i,T_i}^{\text{hash}}\right). \tag{27}$$

TABLE 2. Dataset Characteristics

Symbol	Length	Normal sample	Abnormal sample	Attack rate	Attack type
Smart Meter	383	22128	22175	50.05%	Spoofing, Random Collision, Horizontal Scan, Vertical Scan, Evil Twin
NSL-KDD	34	90167	62444	40.91%	DOS, R2L, U2R, Probe
UNSW-NB15	42	40968	26648	39.41%	Fuzze, Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shellcod, Worms

TABLE 3. Confusion Matrix Classification Results

Type	Normal (Detected)	Abnormal (Detected)
Normal (True)	TP	FP
Abnormal (True)	FN	TN

In this way, we obtain the key for parameter encryption.

3) PARAMETER ENCRYPTION

As the volume of industrial data is generally very large, we employ CounTeR (CTR) in AES to split the large data into small blocks quickly and encrypt parameters. Specifically, CTR generates the key stream based on a counter and K_i, and performs the exclusive OR operation with the plaintext parameters to obtain the encrypted parameters. The decryption process is on the contrary. It is worth noting that with CTR mode, multiple parameter blocks can be encrypted and decrypted simultaneously, which speeds up the encryption process.

Table 1 makes a comparison on the proposed algorithm with the basic AES algorithm. Obviously, by dynamically generating different key pairs for parameter exchange, the proposed algorithm is more secure than the basic AES algorithm even with some complexity enhancement.

D. SUMMARY OF THE PROPOSED STRATEGY

With the aforementioned proposed RMS-CNN, dynamic parameter pruning, and adaptive key generation algorithms, we summarize the LSFL anomaly detection strategy as Algorithm 1 corresponding to the process depicted in Fig. 1.

Algorithm 1: LSFL anomaly detection strategy.

Input: X_i, $W_{i,l}$, $V_{i,l}$, φ_i, ζ_i, ψ_i ($i = 1, 2, ..., I$, $l = 1, 2, ..., L$);

Initialize $y_{i,j}$ as (2) for $i = 1, 2, ..., I$ and $j = 1, 2, ..., J$;
for *communication round* $= 1, 2, ...$ **do**
  for *training round* $= 1, 2, ...$ **do**
    Input X_i to 1D-CNN, and calculate X_i^{spatial} by (9);
    Reduce dimension of X_i^{spatial} and calculate $X_i^{\max}$ and X_i^{avg} by (10) and (11), respectively;
    Input X_i^{avg} to GRU, and calculate X_i^{temporal} by (12);
    Calculate X_i^{res} by (13), and self-attention matrix $X_{i,h}^{\text{attn}}$ by (14);
    Connect $X_{i,h}^{\text{attn}}$ to obtain multi-head self-attention matrix X_i^{mattn} by (15);
    Calculate X_i^{mres} by (16) for anomaly detection;
    Obtain parameter $\tilde{y}_{i,j}$ as (3), and discrete $y_{i,j}$ and $\tilde{y}_{i,j}$ by (17);
    Calculate $e_{i,j}$ and $\tilde{e}_{i,j}$ by (18), and obtain information entropy gain $\Delta e_{i,j}$ by (19);
    Calculate standard deviation Φ_i by (20), and thresholds Ψ_i^{weight}, Ψ_i^{bias}, Ψ_i^{grad} by (21), (22), and (23);
    Prune parameter $y_{i,j}$ to obtain $z_{i,j}$, and reconstruct parameter $z_{i,j}$ with $y_{i,j}$ by (24);
  Generate private key K_i^{r} and public key K_i^{u}, and calculate share key K_i^{s};
  Prepare key material K_i^{m} with $K_i^{\text{m}} = c(K_i^{\text{s}}, R_i)$, and convert K_i^{m} to K_{i,T_i}^{hash} by (25);
  Calculate convert times T_i with L_i and L_i^{hash} by (26);
  Connect K_{i,T_i}^{hash} to obtain K_i by (27);
  ED_i employs CTR in AES to encrypt $z_{i,j}$ with K_i and transmits the parameter to edge server;
  Edge server decrypts parameter $z_{i,j}$ and calculates $\bar{z}_j$ by (6) to update the parameter for ED_i as (8);

Output: Anomaly data;

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENT SETTINGS

All experiments are conducted on TensorFlow-GPU-2.7.0 with Python 3.9 running on Intel i7-11700 CPU and NVIDIA RTX4060-16 G GPU. To fully evaluate the proposed strategy, we select three typical datasets for experimental validation, namely Smart Meters [16], NSL-KDD [17], and UNSW-NB15 [18]. The fundamental characteristics of these datasets are described in Table 2. The data are divided into two parts: 80% of one dataset is used for training, while the remaining 20% is used for testing.

Furthermore, the proposed LSFL anomaly detection strategy with RMS-CNN, denoted as RMS-CNN-LSFL, is compared with two benchmark strategies denoted as CNN-FL and MLP-FL. Herein, CNN-FL is a distributed federal learning anomaly detection strategy based on 1D-CNN with GRU [11], while MLP-FL is a similar strategy based on the MLP network [16].

To evaluate and compare the performances of different strategies, we calculate four performance metrics, namely *Accuracy*, *Precision*, *Recall*, and harmonic mean *F-score*, which are calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{28}$$

$$Precision = \frac{TN}{TN + FP} \tag{29}$$

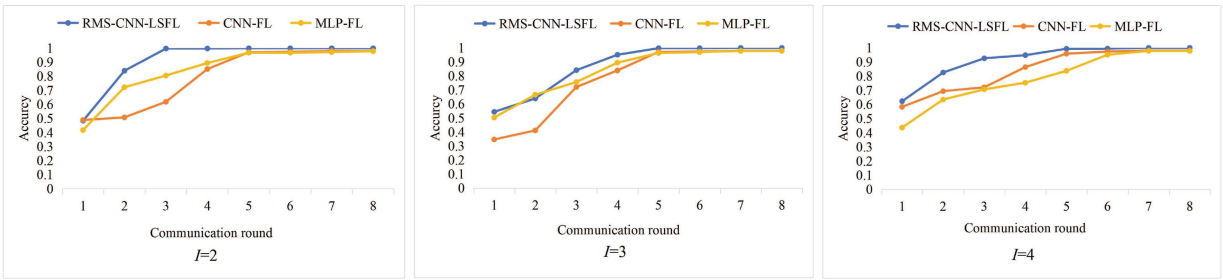


FIGURE 3. Accuracy versus communication round for different strategies with different numbers of devices.

TABLE 4. Performance Evaluations of Different Strategies At Different Datasets

Devices	Datasets	RMS-CNN-LSFL Accuracy Precision Recall F1-score	CNN-FL Accuracy Precision Recall F1-score	MLP-FL Accuracy Precision Recall F1-score
2	Smart Meter	0.99912 0.99914 0.99918 0.99016	0.97928 0.97826 0.97785 0.97805	0.97862 0.97858 0.97858 0.97858
	NSL-KDD	0.99062 0.99125 0.98942 0.99031	0.98838 0.98848 0.98822 0.98835	0.98856 0.98915 0.98769 0.98838
	UNSW-NB15	0.99956 0.99954 0.99957 0.99956	0.98894 0.98773 0.99155 0.98953	0.99019 0.99031 0.98980 0.99005
3	Smart Meter	0.99956 0.99954 0.99957 0.99956	0.97902 0.97884 0.97898 0.97891	0.97885 0.97890 0.97873 0.97882
	NSL-KDD	0.99188 0.99086 0.99246 0.99164	0.98874 0.98880 0.98860 0.98870	0.98692 0.98727 0.98623 0.98673
	UNSW-NB15	0.99956 0.99954 0.99957 0.99956	0.98713 0.98890 0.98248 0.98558	0.99824 0.98757 0.98853 0.98804
4	Smart Meter	0.99982 0.99981 0.99983 0.99982	0.98031 0.97972 0.97634 0.97800	0.97801 0.97815 0.97778 0.97796
	NSL-KDD	0.99210 0.99168 0.99204 0.99186	0.98885 0.98873 0.98890 0.98882	0.98850 0.99021 0.98421 0.98711
	UNSW-NB15	0.99485 0.99486 0.99489 0.99485	0.98805 0.98742 0.98824 0.98783	0.98844 0.98771 0.98881 0.98826

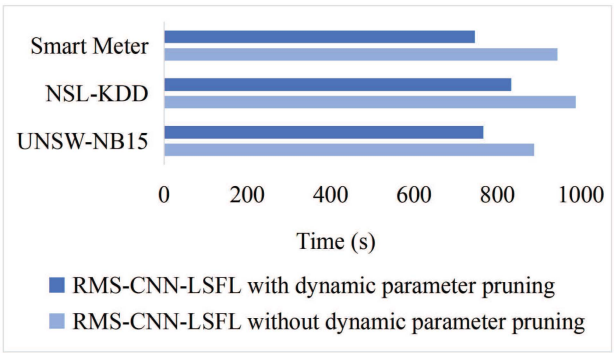


FIGURE 4. Running time of RMS-CNN-LSFL with and without dynamic parameter pruning on different datasets.

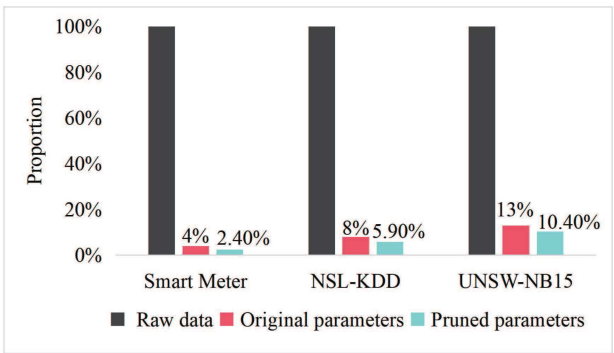


FIGURE 5. Comparison of communication costs on different datasets.

$$Recall = \frac{TN}{TN + FN} \tag{30}$$

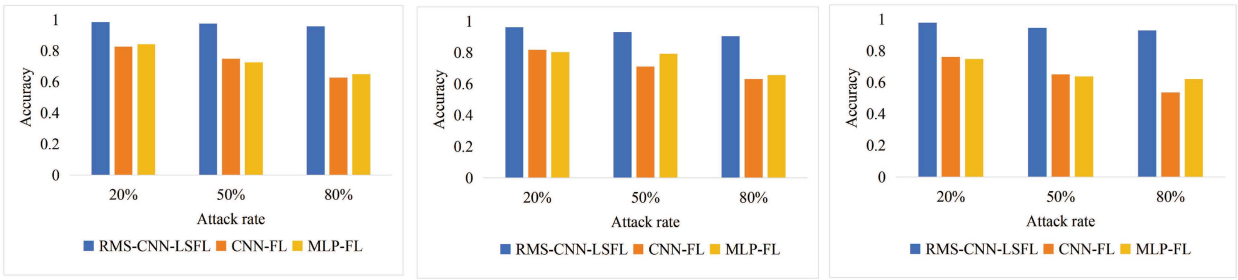
$$F - Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{31}$$

where true positive (TP), false positive (FP), false negative (FN), and true negative (TN) are defined in Table 3.

Obviously, *Accuracy* indicates the proportion of all correctly detected samples to the total samples as given by (28). The higher of the accuracy, the more effectiveness of the anomaly detection strategy. *Precision* indicates the proportion of true abnormal samples among the predicted abnormal samples as given by (29). *Recall* indicates the proportion of abnormal samples correctly detected in true abnormal samples as given by (30). Harmonic mean *F-Score* comprehensively measures precision and recall as given by (31). The higher of the precision, recall, and harmonic mean, the lower probability of false alarm by the anomaly detection strategy.

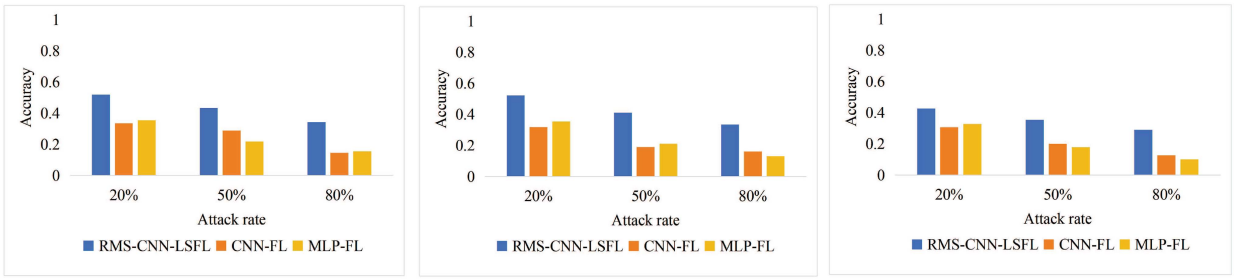
B. PERFORMANCE COMPARISON

Fig. 3 first verifies the effectiveness of the three federal learning strategies on the Smart Meter dataset by evaluating the accuracy versus communication round (i.e., the times for parameter exchange). We can observe that the accuracy increases with the increase of communication rounds, and finally, remains invariability. That is to say, all strategies can converge, indicating that the proposed strategies are effective. Herein, the accuracy of RMS-CNN-LSFL remains higher than those of CNN-FL and MLP-FL, indicating the advantage of the proposed RMS-CNN-LSFL.



(a) Smart Meter (b) NSL-KDD (c) UNSW-NB15

FIGURE 6. **Accuracy for tampering attack on different datasets.**



(a) Smart Meter (b) NSL-KDD (c) UNSW-NB15

FIGURE 7. **Accuracy for injecting attack on different datasets.**

Moreover, the convergence speed of RMS-CNN-LSFL is more quickly than those of CNN-FL and MLP-FL for different numbers of end devices. This is because the combination of multihead self-attention and residual connection in RMS-CNN-LSFL speeds up the process of feature learning. Meanwhile, the residual connection can make the model easier to optimize by mitigating the gradient vanishing problem. Therefore, RMS-CNN-LSFL can converge more quickly and stably.

More specifically, Table 4 comprehensively compares the performance with respect to accuracy, precision, recall, and F-score for different strategies with different numbers of end devices. Obviously, when $I = 4$, the accuracy, precision, recall, and F-score of the proposed RMS-CNN-LSFL strategy on the Smart Meter dataset are 99.982%, 99.981%, 99.983%, and 99.982%, respectively. These performance values are much better than those of CNN-FL with 98.031%, 97.972%, 97.634%, and 97.800%, and those of MLP-FL with 97.801%, 97.815%, 97.778%, and 97.796%. Similarly, the performance evaluations on NSL-KDD and UNSW-NB15 also indicate that RMS-CNN-LSFL achieves much better accuracy, precision, recall, and F-score than CNN-FL and MLP-FL. In detail, the accuracy of RMS-CNN-LSFL is above 99%, while those of CNN-FL and MLP-FL are generally below 99%. The main reason is that the proposed RMS-CNN-LSFL with multihead self-attention network can capture more spatial-temporal features for the data with long-term dependencies.

Fig. 4 compares the runtime of the proposed RMS-CNN-LSFL strategy with and without dynamic parameter pruning on different datasets. Note that the running time include all the time for feature learning, parameter pruning, encryption, and exchange as described in Section III. We can observe that, with dynamic parameters pruning, the runtime is reduced by 21.2%, 15.6%, and 13.7% on the three datasets. This is because less parameters are exchanged after pruning, while the accuracy is not loss.

Fig. 5 depicts the processed data volume at different stages by RMS-CNN-LSFL, CNN-FL, and MLP-FL on different datasets. It is observed that the processed parameters by federal learning is only 4%, 8%, and 13% of the raw data on the three datasets. Furthermore, with dynamic parameter pruning, the parameters are reduced to only 2.4%, 5.9%, and 10.4% of the raw data. That is to say, our proposed strategy saves at least 89.6% wireless communication cost for parameters exchange.

Fig. 6 evaluates how tampering attacks impact the performance of different strategies on the three datasets. With the increase of tampering attack, namely more and more parameters are ineffective, the accuracy of CNN-FL and MLP-FL is gradually decreasing since they do not perform parameter encryption. For this case, once the parameters are tampered, the features and distribution of parameters cannot be accurately captured, thus decreasing the accuracy of anomaly detection by federal learning. In contrast, the accuracy of RMS-CNN-LSFL does not decrease and remains the highest, since RMS-CNN-LSFL performs adaptive key generation to encrypt the parameters and certainly prevent the tampering attacks.

Furthermore, Fig. 7 studies the influence of injecting attacks on the accuracy of different strategies. By evaluating on different datasets, we can observe that the accuracy of

all strategies decreases with the increase of malicious data continuously injected. However, our proposed strategy still remains the highest accuracy than those of CNN-FL and MLP-FL.

Comparing Fig. 7 with Fig. 6, we can also observe that the accuracy of all strategies significantly decreases when there is injecting attack. This is because tampering attack and injecting attack are different kinds of attacks, which make different influence on the valid parameters for federal learning. Tampering attack directly modifies the content of parameters, which can make the unencrypted parameters invalid or even destructive. In this way, the proposed strategy with parameter encryption can protect the pruned parameters from tampering attack. In contrast, injecting attack does not destroy the existing parameters, but add more invalid or even destructive parameters. In this way, the ratio of valid parameters is decreased, which decreases the accuracy of all strategies.

V. CONCLUSION

In this article, we established an end-edge collaborative LSFL architecture and proposed the LSFL anomaly detection strategy for the wireless ICS. First, the RMS-CNN structure was designed for local spatial-temporal feature learning at end devices. Then, the dynamic pruning algorithm based on information entropy gain was proposed to reduce the wireless communication cost for parameter exchange. Furthermore, the adaptive key generation algorithm was presented to encrypt the pruned parameters for edge federal learning. Extensive experiments were performed on three representative datasets, namely Smart Meter, NSL-KDD, and UNSW-NB15, during which two benchmark strategies were compared. The results showed that the proposed LSFL anomaly detection strategy achieves above 99% accuracy on different datasets, where at least 89.6% communication cost is reduced and tampering and injecting attacks are defended.

To summarize, the proposed anomaly detection strategy simultaneously considered the powerful computation resource requirement of federal learning, the low communication cost requirement of end-edge collaborative computing and the high security requirement of parameter exchange in the open radio environment of the wireless ICS. This is different from existing federal learning-based anomaly detection strategies, where only communication cost or security issue is considered. In the future, we will further consider the joint computation and communication allocation for LSFL in the end-edge collaborative architecture.

REFERENCES

[1] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A survey on SCADA systems: Secure protocols, incidents, threats and tactics," *IEEE Commun. Surv. Tuts.*, vol. 22, no. 3, pp. 1942–1976, Thirdquarter 2020.

[2] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine learning for anomaly detection: A systematic review," *IEEE Access*, vol. 9, pp. 78658–78700, 2021.

[3] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial Big Data," *IEEE Trans. Ind. Inform.*, vol. 17, no. 5, pp. 3469–3477, May 2021.

[4] D. Kaur, A. Anwar, I. Kamwa, S. Islam, S. M. Muyeen, and N. Hosseinzadeh, "A Bayesian deep learning approach with convolutional feature engineering to discriminate cyber-physical intrusions in smart grid systems," *IEEE Access*, vol. 11, pp. 18910–18920, 2023.

[5] L. A. C. Ahakonye, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Agnostic CH-DT technique for SCADA network high-dimensional data-aware intrusion detection system," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10344–10356, Jun. 2023.

[6] Y. Chen et al., "Cross-domain industrial intrusion detection deep model trained with imbalanced data," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 584–596, Jan. 2023.

[7] C. Ma et al., "Trusted AI in multiagent systems: An overview of privacy and security for distributed learning," *Proc. IEEE*, vol. 111, no. 9, pp. 1097–1132, Sep. 2023.

[8] Y. Liu et al., "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.

[9] 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. Inform.*, vol. 17, no. 8, pp. 5615–5624, Aug. 2021.

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

[11] F. Zhai, T. Yang, H. Chen, B. He, and S. Li, "Intrusion detection method based on CNN-GRU-FL in a smart grid environment," *Electronics*, vol. 12, no. 5, 2023, Art. no. 1164.

[12] I. A. Khan, D. Pi, M. Z. Abbas, U. Zia, Y. Hussain, and H. Soliman, "Federated-SRUs: A federated-simple-recurrent-units-based IDS for accurate detection of cyber attacks against IoT-augmented industrial control systems," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8467–8476, May 2023.

[13] P. Ruzafa-Alczar et al., "Intrusion detection based on privacy-preserving federated learning for the industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 19, no. 2, pp. 1145–1154, Feb. 023.

[14] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, "Federated semisupervised learning for attack detection in industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 19, no. 1, pp. 286–295, Jan. 2023.

[15] C. Xu, P. Zeng, H. Yu, X. Jin, and C. Xia, "WIA-NR: Ultra-reliable low-latency communication for industrial wireless control networks over unlicensed bands," *IEEE Netw.*, vol. 35, no. 1, pp. 258–265, Jan./Feb. 2021.

[16] M. D. Hossain, H. Ochiai, L. Khan, and Y. Kadobayashi, "Smart meter modbus RS-485 intrusion detection by federated learning approach," in *Proc. 15th Int. Conf. Comput. Automat. Eng.*, 2023, pp. 559–564.

[17] 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.*, 2009, pp. 1–6.

[18] M. Zeeshan et al., "Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSW-NB15 and Bot-IoT data-sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022.

CHI XU (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently a Professor with State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. His research interests include industrial control networks, 5G ultrareliable low-latency communications, edge computing, and tactile Internet.

Dr. Xu is a voting member of IEEE 1918.1 working group for tactile Internet as well as a member of IEEE 1932.1 working group for licensed/unlicensed spectrum interoperability in wireless mobile networks. He also serves as a standardization delegate for 3GPP Technical Specification Group Radio Access Network.

XINYI DU received the B.S. degree in communications engineering, in 2021, from Liaoning Technical University, Huludao, China, where she is currently working toward the master degree in communications and information systems.

Since 2021, she has been with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. Her research interests include industrial control networks and 5G ultrareliable low-latency communications.

LIN LI received the M.S. degree in control theory and control engineering from Shenyang Ligong University, Shenyang, China, in 2016.

She is currently an Engineer with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang. Her research interests include industrial control system and robotics.

XINCHUN LI received the B.S. degree in electrical industrial automation, in 1992, from Liaoning Technical University, Huludao, China. He is a Senior Engineer with Liaoning Technical University, Huludao, China. His research interests include embedded systems and wireless networks.

HAIBIN YU (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 1997.

He is an Academician of Chinese Academy of Engineering, Beijing. Since 1997, he has been a Professor with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang. He has authored and coauthored three books, more than 200 papers, and held more than 50 patents.

Dr. Yu and his research team have proposed the WIA-PA and WIA-FA standards, which are specified as IEC 62601 and IEC 62948, respectively. He was elected as an ISA Fellow for his contributions in fieldbus technologies in 2011. He serves as the Chair of IEC ACART, the Vice-Chair of Chinese Association of Automation, the Chair of China National Technical Committee for Industrial Process Measurement Control and Automation Standardization. He is the Editor-in-Chief of the Chinese journal *Robot* as well as the Associate Editor-in-Chief of the Chinese journal *Information and Control*. His research interests include wireless sensor networks, industrial communication and networked control, industrial automation, and intelligent manufacturing.