# An Intrusion Detection Method for Advanced Metering Infrastructure System Based on Federated Learning

Haolan Liang, Dongqi Liu, Xiangjun Zeng, and Chunxiao Ye

***Abstract*—An advanced metering infrastructure (AMI) system plays a key role in the smart grid (SG), but it is vulnerable to cyberattacks. Current detection methods for AMI cyberattacks mainly focus on the data center or a distributed independent node. On one hand, it is difficult to train an excellent detection intrusion model on a self-learning independent node. On the other hand, large amounts of data are shared over the network and uploaded to a central node for training. These processes may compromise data privacy, cause communication delay, and incur high communication costs. With these limitations, we propose an intrusion detection method for AMI system based on federated learning (FL). The intrusion detection system is deployed in the data concentrators for training, and only its model parameters are communicated to the data center. Furthermore, the data center distributes the learning to each data concentrator through aggregation and weight assignments for collaborative learning. An optimized deep neural network (DNN) is exploited for this proposed method, and extensive experiments based on the NSL-KDD dataset are carried out. From the results, this proposed method improves detection performance and reduces computation costs, communication delays, and communication overheads while guaranteeing data privacy.**

***Index Terms*—Federated learning (FL), advanced metering infrastructure (AMI) system, intrusion detection, data concentrator.**

## I. Introduction

AN advanced metering infrastructure (AMI) system is a key system in the field of smart grid (SG) technology. In recent years, AMI systems have been vigorously developed with the development of the SG [1]-[2]. These systems are based on a two-way communication network that connects power companies and customers. They collect user consumption data and other information and implement necessary control measures [3]. An AMI system provides information platforms and technical support for advanced applications such as real-time two-way interaction, demand response management, and distributed energy generation and storage. These actions heavily rely on information infrastructure and communication networks, and they are most vulnerable to various threats [4], [5]. Large amounts of data are transmitted over the two-way interaction in real time at the network layer. Once attacked, private data may be exposed. The attack surface and vulnerabilities of security protection increase owing to the application of multiple communication technologies in AMI systems and the access of a large number of smart terminals. Meanwhile, the security of enterprise information, the advanced metering system, and grid operation are threatened.

At present, an intrusion detection system (IDS) is widely adopted in the AMI cybersecurity, which can monitor cyberattacks and malicious intrusion behaviors in the AMI system in a timely manner as well as activities from access points. Moreover, the IDS records and prevents suspicious activities marked as intrusions [6]. An IDS is considered as the second layer of protection after the failure of security mechanisms such as encryption and security protocols [7], [8]. IDSs are mainly categorized according to two detection methods based on misuse and abnormality. The misuse-based method needs to create a knowledge base of malicious activities to match and identify intrusions with the known behavior patterns of intruders. However, this method cannot detect unknown attacks, and the attack detection database needs to be constantly updated. The anomaly-based method builds a model by training normal behavior through network features and then detects unknown attacks by checking whether an actual behavior deviates from normal behavior [9] - [10]. For AMI intrusion detection, the anomaly-based method is effectively used.

In recent years, various studies have applied these methods to anomaly-based IDSs with in-depth developments in data mining, machine learning, and deep learning [11] - [13]. Some similar methods are extensively adopted when developing IDSs for AMI systems. Reference [14] presents a comprehensive compilation of several intrusion detection and pre-

Manuscript received: May 8, 2021; revised: August 28, 2021; accepted: January 2, 2022. Date of CrossCheck: January 2, 2022. Date of online publication: June 27, 2022.

This work was supported in part by the National Natural Science Foundation of China (No. 51807013), the Foundation of Hunan Educational Committee (No. 18B137), the Research Project in Hunan Province Education Department (No. 21C0577), and Postgraduate Research and Innovation Project of Hunan Province, China (No. CX20210791).



H. Liang, D. Liu (corresponding author), and X. Zeng are with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410114, China, and H. Liang is also with the Hunan Institute of Engineering, Xiangtan 411104, China (e-mail: lianghaolan@hnie.edu.cn; liu.dongqi@hotmail.com; eexjzeng@qq.com).

C. Ye is with the Hunan Institute of Engineering, Xiangtan 411104, China (e-mail: a747873575@163.com).

DOI: 10.35833/MPCE.2021.000279

vention systems (IDPSs) that are devoted to protecting the SG. Thirty seven IDPSs are analyzed and evaluated by studying their architectures, intrusion detection methods, and programming characteristics, and then the appropriate IDPSs are specified for the SG. Moreover, 13 IDPSs are evaluated and analyzed that focus on AMI systems and it is concluded that the majority of IDPSs for AMI systems employ anomaly detection methods based on artificial intelligence (AI). These methods have been demonstrated to be able to detect zero-day attacks. Reference [15] proposes an intrusion detection method for AMI systems by using an online sequence extreme learning machine, which learns data samples in batches and deletes old data when new data arrives. Thus, it reduces the training time and occupation of storage resources in the AMI system. It is concluded that the solution overtakes other algorithms and obtains good detection performance. However, the specific dataset does not include network records for identifying cyberattacks, nor does it include abnormal behavior patterns. Reference [16] proposes an intrusion detection method for AMI based on an improved online sequence simplified extreme core learning machine, which shortens the detection time and improves the generalization ability and intrusion detection accuracy of the proposed algorithm. Reference [17] proposes an intrusion detection method for AMI based on an artificial immune system (AIS) and establishes an intrusion detection framework in a wide area network (WAN). This method achieves good intrusion detection effects, but the application of this method is only for WANs. Reference [18] proposes an optimal frequency for on-site investigation to investigate potential anomalies of malware footprinting by applying a Markovian decision process. The method is demonstrated that it can investigate and detect worm propagation in AMI systems caused by the potentially infected IP-based smart meters. However, it can not detect various attacks that may occur in AMI systems. Reference [19] proposes an IDS with two-stage collaborative detection processes for smart meters to identify malicious behaviors, which collaboratively use support vector machine (SVM) classifier and the temporal failure propagation graph (TFPG) technique to identify intrusion events. However, this paper only focuses on the neighborhood area network (NAN) domain.

As the weaknesses and vulnerabilities of AMI systems tend to be distributed, the level of intrusion increases. The intrusion behavior is no longer a single behavior, which is gradually becoming more distributed. It is difficult for centralized intrusion detection methods to effectively prevent intrusions. Therefore, distributed intrusion detection methods have aroused the interests of many researchers. Reference [20] proposes a distributed IDS architecture that consists of smart meters, data concentrators, and a central system for AMI systems, using a datastream mining algorithm to analyze the requirements of the three components in the AMI system for detecting anomalies. It is concluded that the datastream mining technique shows promising potential for solving security issues in AMI systems. Reference [21] continues to explore the feasibility of the datastream mining technique used in the IDS architecture [20] and utilizes the NSL-KDD dataset and multiple evaluation measures to analyze the performance of seven existing state-of-the-art datastream mining algorithms. The results demonstrate that these algorithms show promising potential for solving security issues in AMI. Reference [22] proposes a distributed IDS (DIDS) for SG by deploying an intelligent analysis module (AM) in multiple layers of the SG: home area networks (HANs), NANs, and WANs. They use SVM and AIS algorithms to detect and classify malicious data and possible cyberattacks. The effectiveness of the method for improving security is demonstrated through multiple simulations. Reference [23] proposes a real-time distributed intrusion detection system for AMI, which places data concentrators and the headend server in a distributed manner to construct two detection layers. The online clustering "mini-batch *K*-means" is adopted to the DIDS, and the experiments are demonstrated that it suits the requirements of DIDS. However, *K*-means needs to use more memory and time. Reference [24] proposes a smart collaborative advanced IDPS with full distributed architecture, which supports the network and the host-based detection and prevention of attacks, incorporating machine-learning techniques and a rich ontological knowledge base with fuzzy logic analysis to the smart components of IDPS. Experiments show that it can detect and prevent intrusions more efficiently than that of traditional IDPS.

Although the intrusion detection methods proposed for AMI systems in the above studies perform well, these methods need to share large amounts of data over the network and then send it to a data center or distributed independent node for training and intrusion detection. First, these methods have the hidden danger of data privacy leakage. Second, they impose communication delays that make real-time detection difficult, a high communication overhead, and high computation costs.

To address the aforementioned limitations, we propose an intrusion detection method for AMI systems based on federated learning (FL). In the proposed method, the data concentrators do not need to upload all of their data to the data center. Instead, they utilize their computing resources to train the IDS model locally using their own data, and only the model parameters are uploaded to the data center instead of sending an extensive amount of data. The data center aggregates the uploaded model parameters and then disseminates the global improvement model to all data concentrators for collaborative training. An optimized deep neural network (DNN) is designed to train the intrusion detection model, and the NSL-KDD dataset is utilized in experiments. This method aims to achieve better detection performance with privacy protection and to reduce communication delay, computation costs, and communication overhead.

The main contributions of this paper are summarized as follows.

1) We create an intrusion detection model for AMI systems based on FL, and an optimized DNN is designed for the model. Extensive experiments based on the NSL-KDD dataset [25] are carried out.

2) We develop an FL model suitable for SG AMI systems. On one hand, this model helps build a comprehensive intru-

sion detection model by federating multiple data concentrators, and the model helps the data concentrators collaborativey learn the results in an improved model for better detection. On the other hand, this model can deploy an IDS to the data concentrators to process their own data. It can realize real-time detection and maintain data privacy.

This paper is organized as follows. Section II introduces the AMI system architecture and the cybersecurity issues in AMI systems. Section III describes the proposed intrusion detection method for AMI systems based on FL. Section IV verifies the proposed method using the NSL-KDD dataset. Section V concludes this paper.

## II. AMI System Architecture and Cybersecurity Issues in AMI Systems

### A. AMI System Architecture

An AMI system consists of smart meters, data concentrators, data center, and communication network, which is interconnected with the communication network to achieve two-way communication of power data. The AMI system architecture with three layers is shown in Fig. 1.

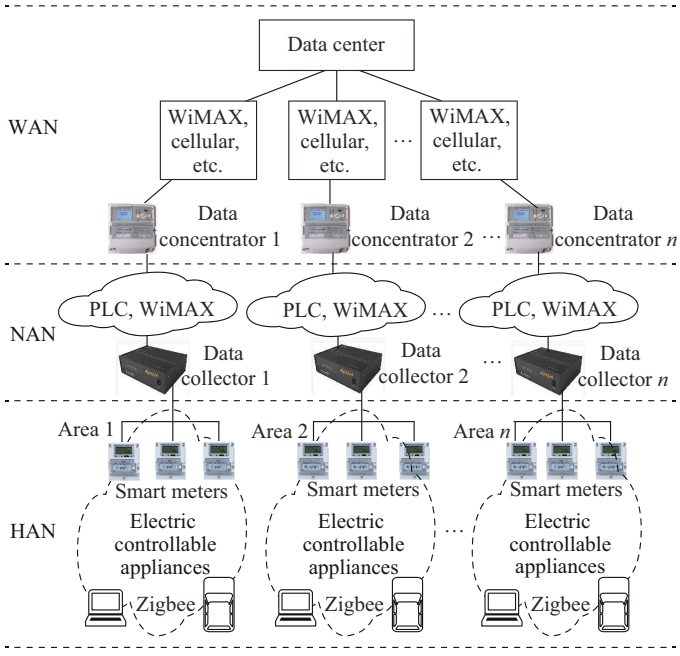


Fig. 1. AMI system architecture.

The first layer is the HAN, which is composed of smart meters and the electric controllable appliances connected by the consumers. The smart meters collect power consumption information of the users and the consumption through the HAN. Zigbee is a low-cost and low-power wireless mesh networking protocol, which can be widely deployed and utilized in a more reliable environment. The HAN communicates among these electric controllable appliances over a short distance. In rare cases, power line communication (PLC) can be used in HAN.

The second layer is the NAN, which is composed of data collectors that communicate with the smart meters of the HAN and aggregate the information from the smart meters. PLC, WiMAX, etc. can possibly be applied at this layer.

The third layer is the WAN, which is composed of the network between the data concentrators and the data center. The data center actively requests power data from the data concentrators through the WAN, or the data concentrators pass through the WAN at a preset time interval. They centrally upload regional data to the data center; then, the data center distributes the electricity price information to users and implements related measures such as load management, demand response, and meter control commands to improve customer service [26]. The WAN is mainly for long-distance communication. It contains various components in order to ensure reliable and stable operation of the SG mixed with wireless and wired networks in the WAN, which is an effective way to ensure the stable transmission of large amounts of data. The regional WAN adopts an optical migration communication method with a high capacity and low delay, mainly using WiMAX and cellular. The WAN can communicate with other WANs with various communication types such as WiMAX, satellite, cellular, and fiber-optic communications [27], [28].

### B. Cybersecurity Issues in AMI Systems

An AMI system realizes a two-way communication of power consumption data, electricity price billing, load transfer, and remote control instructions through interconnections with computer networks. Although the regulation efficiency of the SG has been improved, the application of various wireless and wired heterogeneous communication technologies in AMI systems and the access of a large number of smart terminal devices greatly increase the possibility of cyberattacks owing to the expanded use of software and the introduction of wireless interfaces [29]. The drawbacks of implementing AMI systems are vulnerabilities to potential cyber threats [22], [30], [31].

The cybersecurity standards of AMI systems include confidentiality, integrity, availability, and accountability [8], and cyberthreats can be categorized into two situations.

The first situation is to attack the access node of an AMI system such as a smart meter, a data concentrator, and the data center. Device-based attacks are manipulated through the security flaws of the device terminal to perform malicious activities such as tampering with metering storage, man-in-the-middle (MIM) attacks, denial of service (DoS) attacks, or unauthorized use of services. Digital technologies enhance the extensibility and scalability of metering functions. However, they also introduce new attack vectors to AMI systems. An attacker can insert malicious code in the memory of a smart meter to tamper with stored data [32]. Penetration testing identifies a number of possible attacks against smart meters, including meter spoofing, DoS, and power disconnection [33]. For instance, the device memory in the AMI system could be modified by inserting malicious software, or a disconnect command may be sent to the smart meters, which would block the transmission of metering information.

The second situation is to attack the three-layer communication network for AMI systems. Unlike the traditional transmission control protocol (TCP)/IP in the network, multiple wired and wireless communication media and numerous public and private protocols may be applied in the WAN, NAN, and HAN because of their convenience and low cost, making the three-layer communication networks vulnerable to malicious attacks. For example, Zigbee is easy to compromise with a DoS attack [34]. More examples of wireless network technologies are as follows: the configuration integrity and routing, e.g., distributed DoS (DDoS) and communication traffic, illegitimate network operations, inconsistent traffic direction, data alteration, unintentional dissemination of consumer data, unresponsive destination nodes, the overusage of grid bandwidth, and the overconsumption of power signal [31].

In addition, each two-way communication path supports control and measurement in the AMI system, which has the potential to become an entry point for both physical attacks and cyberattacks that may be used by anyone with malicious intent. Wireless networks can easily be probed by attackers and are susceptible to MIM attacks. Moreover, there exists the possibility of logging into these nodes, reprogramming the measurements, and controlling the commands. Therefore, data privacy would be compromised, leading to significant errors in power metering, which will lead to power outages.

## III. Intrusion Detection Mothod for AMI Systems Based on FL

### A. Intrusion Detection Model Based on FL

FL is a new distributed learning paradigm to decentralize training data. A machine-learning model is built using a database distributed on the edge, and only the model parameters are uploaded to the data center instead of the original data, which prevents data leakage and protects user privacy. At the same time, FL can also solve the problems related to limited network bandwidth, communication delays, and high communication costs. FL is an AI model that meets the requirements of data privacy, information security, and communication performance [35]-[37]. In the proposed method, FL is used to utilize machine learning for intrusion detection to train an AMI model instead of utilizing the data center to perform a detection task after the data concentrators send their captured data to the data center. We decentralize the machine learning tasks by moving the training of IDSs to the data concentrators. The data concentrators are the edge training nodes, and the data center serves as the aggregator for global model training.

The intrusion detection model for AMI systems based on FL, as shown in Fig. 2, mainly comprises the data center and $n$ data concentrators.

1) Data center: the data center acts as a coordinator for the data concentrators and is responsible for constructing a global intrusion detection model by federating and aggregating the model parameters of locally trained models at each data concentrator. In order to obtain an optimal intrusion detection model, multiple rounds of communication of the interactions between the data center and the data concentrators are required. The optimized model is then communicated back to the distributed data concentrators. Hence, the knowledge is shared among them. This sharing scheme results in better learning for each data concentrator, as it enables an edge device, i. e., a data concentrator, to detect intrusions based on the comparable behavior generated from different participating devices, which allows a data concentrator to benefit from the peer-to-peer model. In short, the data center in the proposed method aims to distribute the training results among multiple data concentrator defenders to generate appropriate defense strategies.

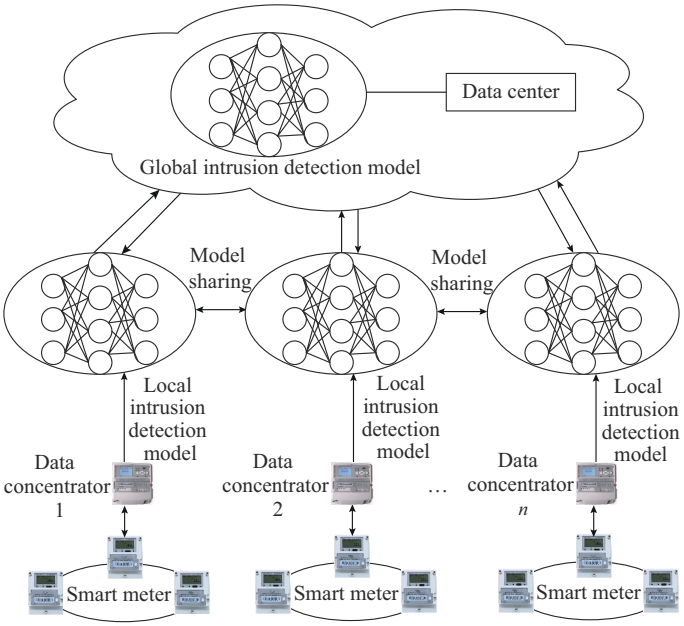


Fig. 2. Intrusion detection model for AMI systems based on FL.

2) Data concentrator: each data concentrator represents an edge-node owner in the AMI system. The IDSs are distributed in the data concentrators, which are responsible for monitoring the network traffic from the NAN and collecting the network traffic from smart meters. The data concentrators are in charge of building a local intrusion detection model based on their own monitored network traffic data. Once a local intrusion detection model for each data concentrator has been trained, only the parameters of local intrusion detection model are uploaded to the data center by repeated interaction with the data center until the model converges. Data concentrators have the autonomy of local intrusion detection through the local execution of training, parameter optimization, and inference. Therefore, it preserves the database, maintains data privacy, and accelerates the detection time since an analysis is performed locally, where the network traffic data are generated.

We assume that there are two types of data concentrators that do not participate in cooperation. One is a malicious device, and the other is a selfish device. First, we assume that the data concentrators in the the proposed method are normal before being trained, which can be ensured by an initial security inspection offline. Then, the data concentrators under external attacks can be identified and provide effective resistance using the intrusion detection model.

### *B. Steps for Intrusion Detection for AMI System Based on FL*

We transplant the FL into an AMI system for intrusion detection and explore the similarities between the SG AMI system and FL. The steps for intrusion detection for the AMI system based on the FL are as follows.

*Step 1*: the data center generates a general intrusion detection model, builds a DNN structure in the model, and determines the hidden layer, neurons, number of iterations, and other parameters. We use a DNN as the neural network architecture for the IDS.

*Step 2*: the data concentrators that participate in FL training download the general model uniformly.

*Step 3*: the data concentrators monitor their network data using a profiler and use their anomaly data to train a local intrusion detection model. There is no information interaction between data concentrators, and each concentrator only uploads its model information.

*Step 4*: only the model parameters of the updated intrusion detection models are shared with the data center instead of uploading a large amount of sensitive and private data from the data concentrators.

*Step 5*: once the updated local intrusion detection models of all the data concentrators participating in FL training are received, the data center uses the federated averaging algorithm to aggregate the parameters from different intrusion detection models of the data concentrators and creates a new updated intrusion detection model. The complete process and steps of the algorithm are presented in Algorithm 1.

**Algorithm 1**: intrusion detection for AMI systems based on FL

**Input**: dataset $D_k$, initial global model parameters $w_G^t(t \geq 0)$, local mini-batch size $B$, number of data concentrators $K$, learning rate $\eta$, number of local epochs $E$, and fraction of data concentrators $C$

**Output**: the parameters of global intrusion detection model for next communication round $w_G^{t+1}$

**Procedure that the data center executes**:
  Initialize $w_G^t$
  **for** each round $t = 1, 2, \ldots, n$ **do**
    $m$ max $(CK,1)$
    $S_t$ is the random set of $m$ data concentrators
    **for** each data concentrator $k \in S_t$ **in parallel do**
      $w_k^{t+1} = w$, where $w$ is the output of the data concentrator update $(k,w)$
    **end for**
    $w_G^{t+1} = \sum_{k=1}^{K} \frac{n_k}{N} w_k^{t+1}$
  **end for**
**end procedure that the data center executes**
**Procedure that the data concentrators execute**:
  Data concentrator update $(k,w)$
  Run on the selected concentrator $k$
  $B_k$ is the split local intrusion detection data into batches of size $B$
  **for** each local epoch $i$ from 1 to $E$ **do**
    **for** batch $b \in B_k$ do
      $w = w - \eta \nabla l(w; b)$
    **end for**
  **end for**
Return $w$ to the data center
**end procedure that the data concentrators execute**

*Step 6*: the data center sends the updated model parameters to each data concentrator.

*Step 7*: each data concentrator uses the updated model parameters and improves them using its newly generated data. *Steps 4*, *5*, *6*, and *7* are repeated to update iteratively and to improve the federated intrusion detection model until convergence.

The FL utilized in the AMI system provides the following benefits. First, the security and privacy of the data in the data concentrators are preserved. Second, anomaly detection is realized in real time, and response messages are provided in a timely manner. In addition, the communication delay due to the large amounts of transmitted data is reduced, and the risk of data leakage is avoided. Moreover, even if there is no connection between the data concentrators and the data center, the local intrusion detection model of a data concentrator can still detect anomalies in the AMI system. A data concentrator benefits from its peer intrusion detection models and improves the detection performance of the model.

### *C. General Model of FL*

We use a DNN as the general model of FL for intrusion detection for AMI system. A DNN is an artificial neural network (ANN) method, and a multilayer perceptron (MLP) model is a forward structure of an ANN, which contains multiple hidden layers, also called a DNN. An MLP maps a set of input vectors to a set of output vectors, and it can be observed as a directed graph consisting of multiple node layers in which each layer is fully connected to the next layer. Except for the input node, each node is a neuron with a nonlinear activation function. We use the supervised learning method of the backpropagation (BP) algorithm to train the MLP. An MLP is the improvement of a perceptron and overcomes a weakness of the perceptron: it cannot recognize linearly inseparable data.

An MLP is defined mathematically as $\boldsymbol{O}: \mathbb{R}^m \times \mathbb{R}^n$, where $m$ is the size of the input vector $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{m-1}, \boldsymbol{x}_m]$ and $n$ is the size of the output vector $\boldsymbol{O}(\boldsymbol{x})$. Each layer $h_i$ is computed by a nonlinear activation function, mathematically expressed as:

$$h_i(\boldsymbol{x}) = f(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x} + \boldsymbol{b}_i) \tag{1}$$

where $h_i: \mathbb{R}^{d_i - 1} \to \mathbb{R}^{d_i}$, and $d_i$ is defined as the size of the input; $f(\cdot)$ is any activation function; $\boldsymbol{w}_i \in \mathbb{R}^{d_i}$; and $\boldsymbol{b}_i \in \mathbb{R}^{d_i}$.

With dropout [38], (1) becomes (2), which is used to avoid overfitting problems during the training:

$$h_i(\boldsymbol{x}) = \boldsymbol{r}_i^{(l)} f(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x} + \boldsymbol{b}_i) \tag{2}$$

$$\boldsymbol{r}_i^{(l)}: Bernoulli(p) \tag{3}$$

where $\boldsymbol{r}_i^{(l)}$ is a vector of Bernoulli random variables, each of which has probability $p$ of being 1; and $l \in \{1, 2, \ldots, L\}$ is the index of the hidden layers of the DNN.

For the output-layer activation function, the MLP model uses a softmax function as the nonlinear activation function in the multiclass problem since the attack types are multiclass. The softmax activation function outputs the probabilities of each class and selects the largest value among the probability values to give a more accurate value. The mathematical formula for the softmax activation function is:

$$softmax(\boldsymbol{x}_i) = \frac{\mathrm{e}^{\boldsymbol{x}_i}}{\sum_{j=1}^{n} \mathrm{e}^{\boldsymbol{x}_j}} \tag{4}$$

where $0 < softmax(\boldsymbol{x}_i) \leq 1$; $\boldsymbol{x}_i$ is the input for each layer; and $e^{x_i}$ is the output for each node at the hidden layer.

We select a rectified linear unit (ReLU) as the hidden-layer activation function, as it helps mitigate the vanishing error gradient issue [39]. The advantages of the ReLU are that it is faster than other nonlinear activation functions and facilitates the training of the MLP model with a large number of hidden layers. Moreover, its complicated classification boundaries are more expressive and the performance of the IDS is enhanced [40].

The ReLU is mathematically defined as:

$$f(\boldsymbol{x}) = \max(0, \boldsymbol{x}) \tag{5}$$

It is essential to determine the optimal parameters to achieve good performance for modeling an MLP, in which constructing the loss function is the first step. A loss function is used to calculate the magnitude of the difference between the predicted and target values. In this paper, we use the categorical cross-entropy to calculate the error or loss function. The final layer is also passed from the softmax activation function so that the output value must have a probability between 0 and 1. The cross-entropy function is expressed as [41]:

$$J(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{y}', \boldsymbol{y}) = -\boldsymbol{y} \ln \boldsymbol{y}' - (1 - \boldsymbol{y}) \ln(1 - \boldsymbol{y}') \tag{6}$$

where $J(\cdot)$ is the loss function; $\boldsymbol{W}$ is the weight matrix; $\boldsymbol{b}$ is the bias vector; $\boldsymbol{y}$ is the output vector formed from actual probability values; and $\boldsymbol{y}'$ is the output vector formed from the expected probability values.

The training parameters are usually learned with gradient descent, which is a nonlinear optimization problem. Gradient descent is randomly initiated by setting a set of deep network parameters, but it is updated at each step to decrease the gradient by computing the gradient descent of the nonlinear function being optimized.

The technique for minimizing $J(\cdot)$ is stochastic gradient descent, which is a standard gradient computed via BP using a constant $\alpha$ as a learning rate. The final parameters $\boldsymbol{W}$ and $\boldsymbol{b}$ are obtained by averaging. Equations (7) and (8) show the iteration of standard gradient descent upon the updates of $\boldsymbol{W}$ and $\boldsymbol{b}$ using sample $i$ until the convergence is obtained. Usually, sample $i$ selects a minibatch to train during each iteration since it can simplify the learning process and avoid local optima [42].

$$\boldsymbol{W}_{ji} = \boldsymbol{W}_{ji} - \alpha \frac{\partial J(\boldsymbol{W}, \boldsymbol{b}/j)}{\partial \boldsymbol{W}_{ji}} \tag{7}$$

$$\boldsymbol{b}_{ji} = \boldsymbol{b}_{ji} - \alpha \frac{\partial J(\boldsymbol{W}, \boldsymbol{b}/j)}{\partial \boldsymbol{b}_{ji}} \tag{8}$$

where $\boldsymbol{W}_{ji}$ and $\boldsymbol{b}_{ji}$ are the model parameters of the weight matrix and bias vector, respectively.

After analyzing the hyperparameters, the hyperparameters of the DNN have to be appropriately and experimentally selected in order to achieve optimal performance in FL applications. Therefore, we carry out many experiments to analyze and compare the performance of the DNN using different hyperparameters in order to obtain a better model for the FL. Therefore, we build the model using three hidden layers. The first, second, and third layers have 256, 512, and 256 neural units, respectively. We use an ReLU as the activation function and a dropout rate of 0.4 to ensure the regularization after the first hidden layer. The dropout layer helps control the overfitting by removing an individual unit with a random probability while training the model. The softmax activation function is used for the output layer of the classifier [43]-[45]. The model architecture of DNN is shown in Fig. 3 and the hyperparameters of the DNN model are listed in Table I.

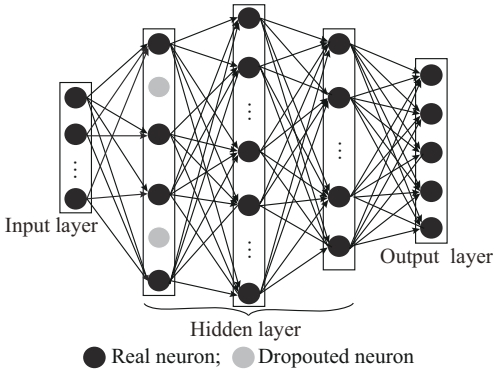


Fig. 3. Model architecture of DNN.

TABLE I
HYPERPARAMETERS OF DNN

| Hyperparameter | Value |
|---|---|
| DNN units of input layer | 122 |
| DNN units of hidden layer 1 | 256 |
| Dropout rate of hidden layer 1 | 0.4 |
| DNN units of hidden layer 2 | 512 |
| DNN units of hidden layer 3 | 256 |
| DNN units of output layer | 5 |

## IV. EXPERIMENT AND ANALYSIS

Since an AMI system can be treated as a combination of a computer network and a power system with some additional new characteristics, some communication scenarios observed for computer networks can resemble those of an AMI system. For example, some attacks such as blocking data and theft of information are common in the NSL-KDD dataset [25] for intrusion detection in computer networks. Therefore, we select the NSL-KDD dataset for the experiments.

### A. Data Description

The NSL-KDD dataset contains 125973 records in the training dataset and 22544 records in the testing dataset. Each record has 41 features and a label. The features include time-based network traffic data and host-based network traffic data, and the label types include normal and abnormal types. Among the abnormal types, there are four main types of attacks, i.e., DoS, Probe, user-to-root (U2R), and root-to-local (R2L). The training dataset has one normal and 22 different types. In order to verify the effectiveness of the model in detecting unknown attacks, 17 new attacks have been added to the testing set. We use "KDDTrain+" and "KDDTest+" for training and testing, respectively. In this paper, we combine the two datasets, of which 80% is used for train-

ing and 20% is used for testing. In order to simulate real scenarios in an AMI system, the dataset is not independently and identically distributed in each concentrator for training and testing. The samples used in the experiment are randomly distributed among data concentrators [46]. The attack categories for training and testing datasets are shown in Table II.

TABLE II
ATTACK CATEGORIES FOR TRAINING AND TESTING DATASETS

| Attack category | 22 attacks in training dataset | 17 new attacks in testing dataset |
|---|---|---|
| Dos | Back, land, Neptune, pod, smurf, teardrop | Apache2, Processtable, mailbomb, udpstorm |
| Probe | Ipsweep, nmap, portsweep, satan | Saint, mscan |
| U2R | Buffe_roverflow, loadmodule, perl, rootkit | Ps, snmpguess, sqlattack, worm, xterm |
| R2L | Spy, Warezclient, ftp_write, guesspasswd, imap, multihop, phf, warezmaster | Httptunnel, named, sendmail, snmpgetattack, xlock, xsnoop |

### B. Data Preprocessing

1) The dataset contains three discrete features including the protocol type, service, and flag. In order to enable the dataset to be recognized and trained by the DNN, one-hot encoding is used to separate the three discrete data features. We convert the three discrete data features into continuous data and then express these data as numerical values.

2) In order to ensure that the training and testing results are effective and reliable, the features of the dataset must be normalized, and all feature data must be normalized in the range of [0,1]. In this paper, we use min-max normalization during data processing.

3) After processing the data labels of the training and testing datasets, the labels are categorized into five categories, i. e., normal, DoS, Probe, U2R, and R2L. Label coding is used to convert these five categories of labels into continuous numerical variables expressed as 0, 1, 2, 3, and 4. Finally, we obtain 122-dimensional features and a one-dimensional label.

### C. Evaluation Metrics

To analyze the performance of the central intrusion detection model and the proposed method, we adopt the accuracy $A_{cc}$, recall $R$, precision $P$, F1-measure $F_1$, and computation cost for comparison [47]. Moreover, a confusion matrix is created to calculate all performance measures, which is a table that represents the performance of a classification model on a set of test data for which the true values are identified. The specific formulas for calculating the performance measures are as follows, and the confusion matrix is shown in Table III.

$$A_{cc}=\frac{T_P+T_N}{F_P+F_N+T_P+T_N} \tag{9}$$

$$R=\frac{T_P}{T_P+F_N} \tag{10}$$

$$P=\frac{T_P}{T_P+F_P} \tag{11}$$

$$F_1=2\frac{P_r R_e}{P_r+R_e} \tag{12}$$

where $T_P$ is the number of samples correctly classified as the attack type; $T_N$ is the number of samples correctly classified as the normal type; $F_P$ is the number of normal samples that are incorrectly classified as the attack type; and $F_N$ is the number of attack samples that are incorrectly classified as the normal type.

TABLE III
CONFUSION MATRIX

| True class | Predicted class | |
|---|---|---|
| | Negative class | Positive class |
| Negative class | $T_N$ | $F_P$ |
| Positive class | $F_N$ | $T_P$ |

### D. Result Analysis

All experiments are performed on Windows desktop computers equipped with a central processing unit (CPU) running at 4.4 GHz, 64 GB of random access memory (RAM), and an NVIDIA GeForce MX250 graphics processing unit (GPU). The proposed method and centralized model are implemented in PyTorch.

The parameters for the simulations are listed in Table IV.

TABLE IV
SYSTEM PARAMETERS FOR SIMULATIONS

| Parameter | Value |
|---|---|
| Batch size | 128 |
| Learning rate | 0.001 |
| Number of local epoch | 15 |
| Communication round | 10, 20, 50, 100 |
| Loss function | Cross entropy |
| Optimizer | Adam |
| Fraction | 0.1 |
| Number of users | 50 |

1) Result 1: the influence of the participation ratio of data concentrators for the proposed method. The number of data concentrators affects its iterative convergence performance. In the experiments, we select 50 data concentrators to participate in the experiment and explore the influence of the number of data concentrators on the proposed method. The training and testing datasets are randomly and unevenly assigned to 50 data concentrators. We vary the parameter $C$ for each training round. For instance, $C=1$ means that all of the data concentrators in the AMI system are used for training. $C=0.5$ implies that half of data concentrators are used, and $C$ equaling between 0.1 and 0.5 means a few data concentrators in the AMI system are used. The results are shown in Fig. 4. As can be observed from the results, better convergence performance is achieved when more data concentrators participate in the proposed method training. Moreover, the proposed method helps improve the accuracy of the models of the edge data concentrators because the limited data stored by any concentrator can easily fall into a local opti-

mum. Furthermore, the model trained by other concentrators can effectively help participants to discard local optima to obtain a more accurate model. Further, it can solve the problems of data barriers that commonly exist among data concentrators.

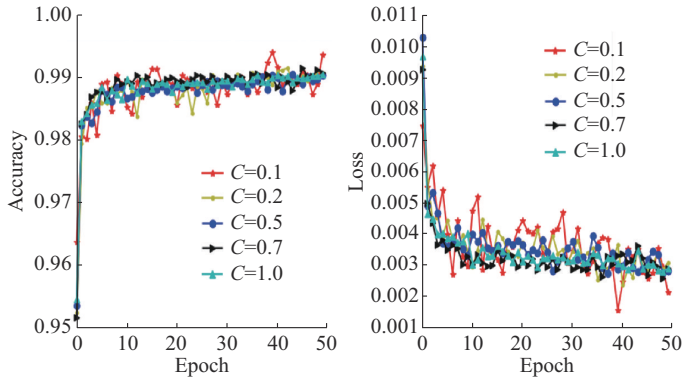


Fig. 4. Performance of different numbers of data concentrators participate in training proposed method. (a) Accuracy. (b) Loss.

2) Result 2: the computation costs of the centralized and the proposed method. We qualitatively analyze the DNN under the centralized machine-learning model and the proposed method using the same specifications for the training samples to compare the computation costs of the two methods. The time required for the traditional centralized machine-learning model $t_{centralized}$ is the time required for all samples to be trained once through the DNN, while the time required for the proposed method $t_{FL}$ is the maximum time required for multiple small local samples $D_i$ to be trained once through the local model. The results are summarized in Table V.

TABLE V
CONFUSION MATRIX COMPARISON OF COMPUTATION COST

| Method | Participant | Training data size | Time required for one iteration (s) |
|---|---|---|---|
| Centralized | 1 | Data center training set size $D=D_1 \cup D_2 \cup \ldots \cup D_N$ | 39.2904 |
| Proposed | ≥2 | Local training dataset of each participant $D_i$ | 23.5514 |

The computation time required for the proposed method is expressed as:

$$t_{FL}=\max\{t_i\} \tag{13}$$

where $\max\{t_i\}$ is the training time of the data concentrator with the longest training time in a communication round.

We conclude from Table V that the proposed method approximately reduces the computation time by a factor of two compared with that of the centralized model.

3) Result 3: the detection performance of the centralized and the proposed methods. We compare the intrusion detection performance of the centralized and the proposed methods. In this set of experiments, we assume a case where five data concentrators are distributed in the AMI system, and each data concentrator represents an IDS. We consider five IDSs that monitor network traffic generated by five subnetworks. Table VI presents the detection accuracy of the centralized and the proposed methods. Table VII presents a comparison of the detection performances of the proposed and centralized models for all attack types. Figures 5 and 6 show the confusion matrices of the proposed and the centralized models. The experimental results in Table VI show that the detection rate for the proposed method is higher than that of the centralized model. As shown by the results in Table VII, we see that the proposed method outperforms the centralized model in terms of $A_{cc}$, $F_1$, and $R$ for all attack types. Moreover, it is observed from the results in Table VII and the confusion matrices in Figs. 5 and 6 that the proposed method increases the detection rate of R2L attacks when the system is faced with a small sample intrusion due to R2L and U2R attacks. Furthermore, the centralized model can not detect U2R attacks, whereas the proposed method is able to detect U2R attacks at an approximate detection rate of 70%. We conclude that the detection performance of the proposed method is better than that of the centralized model.

TABLE VI
DETECTION ACCURACIY OF PROPOSED AND CENTRALIZED METHODS

| Model | Accuracy (%) |
|---|---|
| Proposed | 99.32 |
| Centralized | 98.94 |

TABLE VII
COMPARISON OF DETECTION PERFORMANCE OF PROPOSED AND CENTRALIZED METHODS FOR ALL ATTACK TYPES

| Attack type | Proposed model | | | Centralized method | | |
|---|---|---|---|---|---|---|
| | $A_{cc}$ (%) | $F_1$ (%) | $R$ (%) | $A_{cc}$ (%) | $F_1$ (%) | $R$ (%) |
| Normal | 99.34 | 99.38 | 99.45 | 98.91 | 99.07 | 99.23 |
| DoS | 99.96 | 99.96 | 99.38 | 99.76 | 99.82 | 99.88 |
| Probe | 98.84 | 99.12 | 99.38 | 97.56 | 98.26 | 98.97 |
| R2L | 93.73 | 89.33 | 86.33 | 93.17 | 88.39 | 80.55 |
| U2R | 80.00 | 73.41 | 67.23 | | | 0 |

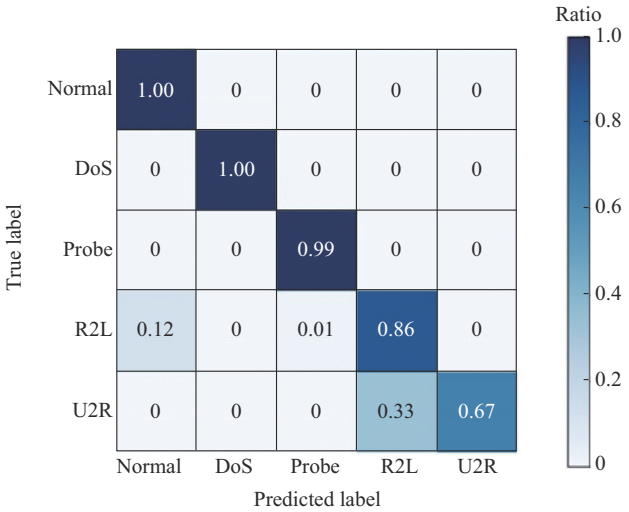


Fig. 5. Confusion matrix for proposed method.

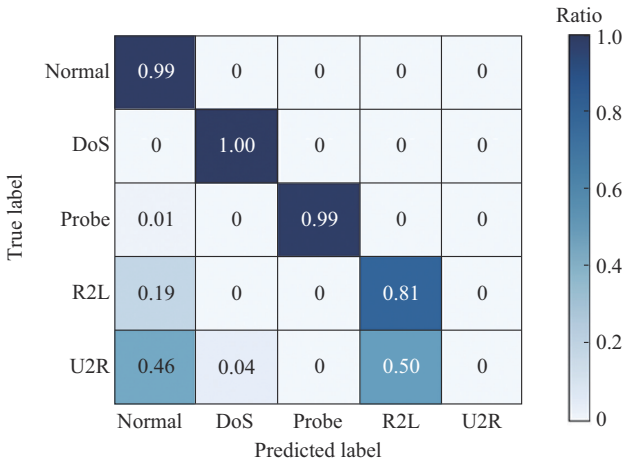


Fig. 6. Confusion matrix for centralized model.

4) Result 4: the convergence speeds and communication overheads of the centralized and the proposed methods.

In real-time anomaly detection, a response message is crucial. Quick and accurate detections of abnormal data at the data concentrator are very important for security protection of AMI system. As shown in Figs. 7-10, the proposed method significantly lowers the loss function and improves detection accuracy. The convergence speed of the proposed method is more than twice faster compared with the centralized model during the training on the same dataset. In addition, it reaches a stable detection accuracy of 99% in only 10 communication rounds, while the centralized model needs more than twice as many communication rounds. We conclude that it reduces the communication delay and communication overhead.

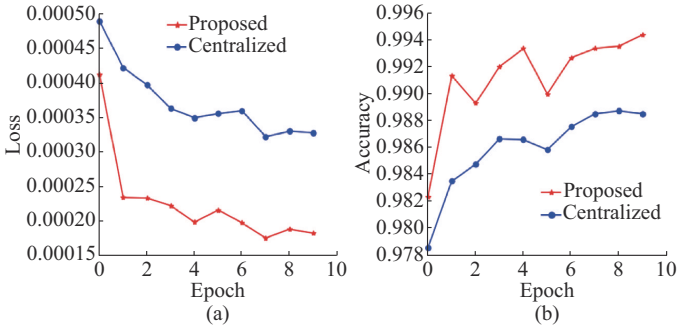


Fig. 7. Comparison of proposed and centralized models with 10 communication rounds. (a) Loss. (b) Accuracy.

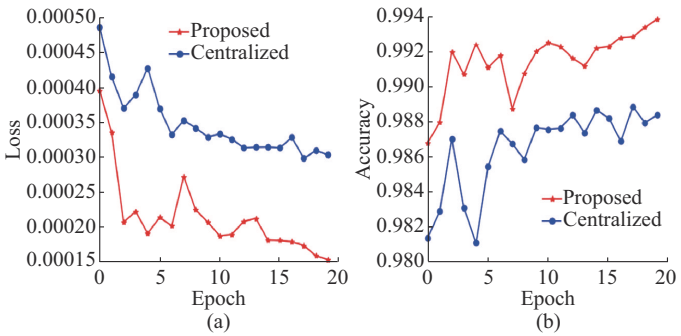


Fig. 8. Comparison of proposed and centralized models with 20 communication rounds. (a) Loss. (b) Accuracy.

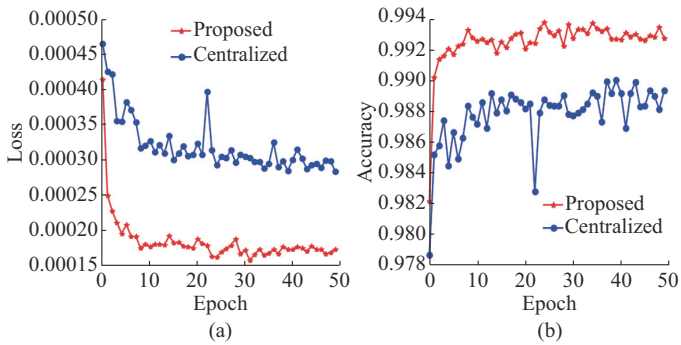


Fig. 9. Comparison of proposed and centralized models with 50 communication rounds. (a) Loss. (b) Accuracy.

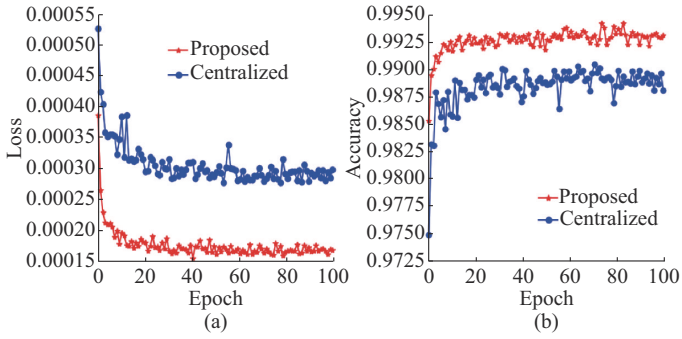


Fig. 10. Comparison of proposed and centralized models with 100 communication rounds. (a) Loss. (b) Accuracy.

## V. Conclusion

We propose an intrusion detection method for AMI systems based on FL. We transplant the FL into an AMI system for distributed collaborative intrusion detection while maintaining data privacy. Moreover, we design an optimized DNN for the proposed method in order to train and detect the intrusions.

We utilize the NSL-KDD dataset to carry out extensive experiments. The results show that the proposed method for AMI systems achieves better detection performance according to four evaluation metrics calculated from the confusion matrices.

Further, it reduces communication overhead and computation costs. When faced with small intrusion of R2L and U2R attacks, the proposed method has an increased detection rate of approximately 7% for R2L attacks and a 60%-70% improved detection accuracy for U2R attacks. In addition, better detection performance can be achieved as more data concentrators participate in the proposed method for training.

In the future, we will investigate encryption strategies for the parameters of the proposed method to further improve the reliability and security. Furthermore, we will collect real SG intrusion data to improve the proposed method.

## References

[1] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power & Energy Magazine*, vol. 7, no. 2, pp. 52-62, Feb. 2009.

[2] J. Zhang and Z. Chen, "The impact of AMI on the future power system," *Automation of Electric Power Systems*, vol. 43, no. 2, pp. 20-23, Jan. 2010.

[3] M. Arian, V. Soleimani, B. Abasgholi *et al.*, "Advanced metering infra-

structure system architecture," in *Proceedings of 2011 Asia-Pacific Power and Energy Engineering Conference*, Wuhan, China, Mar. 2011, pp. 1-6.

[4] R. Shein, "Security measures for advanced metering infrastructure components," in *Proceedings of 2010 Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Chengdu, China, Mar. 2010, pp. 1-3.

[5] A. Anzalchi and A. Sarwat, "A survey on security assessment of metering infrastructure in smart grid systems," in *Proceedings of the IEEE Southeast conference in Fort Lauderdale*, Fort Lauderdale, USA, Apr. 2015, pp.1-4.

[6] X. Miao, X. Chen, X. Ma *et al*., "Comparing smart grid technology standards roadmap of the IEC, NIST and SGCC," in *Proceedings of 2012 China International Conference on Electricity Distribution (CICED 2012)*, Shanghai, China, Sept. 2012, pp.1-4.

[7] D. Li, Z. Aung, J. R. Williams *et al*., "Efficient authentication scheme for data aggregation in smart grid with fault tolerance and fault diagnosis," in *Proceedings of 2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, Washington DC, USA, Jan. 2012, pp.1-8.

[8] F. M. Cleveland, "Cyber security issues for advanced metering infrastructure (AMI)," in *Proceedings of 2008 IEEE PES General Meeting*, Pittsburgh, USA, Jul. 2008, pp. 1-5.

[9] R. Berthier, W. H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: requirements and architectural direction," in *Proceedings of 2010 First IEEE International Conference on Smart Grid Communications*, Gaithersburg, USA, Oct. 2010, pp. 350-355.

[10] M. Ni, M. Li, J. Li *et al*., "Concept and research framework for coordinated situation awareness and active defense of cyber-physical power systems against cyber-attacks," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 3, pp. 477-484, May 2021.

[11] Y. Dong, R. Wang, and J. He, "Real-time network intrusion detection system based on deep learning," in *Proceedings of 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, Oct. 2019, pp. 1-4.

[12] S. Yu, J. Cha, T. Lee *et al*. (2019, Nov.). Features recognition from piping and instrumentation diagrams in imageformat using a deep learning network. [Online]. Available: https://doi. org/10.3390/en12234425.

[13] M. Ishaque and L. Hudec, "Feature extraction using deep learning for intrusion detection system," in *Proceedings of 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabla, May 2019, pp. 1-5.

[14] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the smart grid: a comprehensive compilation of intrusion detection and prevention systems," *IEEE Access*, vol. 7, pp. 46595-46620, Apr. 2019.

[15] Y. Li, R. Qiu, and S. Jing, "Intrusion detection system using online sequence extreme learning machine (OS-ELM) in advanced metering infrastructure of smart grid," *PloS One*, vol. 13, no. 2, p. e0192216, Sept. 2018.

[16] F. Liu, Z. Wu, and L. Ding, "AMI intrusion detection algorithm based on improved online sequence learning machine," *Computer Engineering*, vol. 46, no. 9, pp. 136-142, Sept. 2019.

[17] K. Song, P. Kim, V. Tyagi *et al*. (2018, May). Artificial immune system(AIS) based intrusion detection system (IDS) for smart grid advanced metering infrastructure (AMI) networks. [Online]. Available: https://vtechworks.lib.vt.edu/handle/10919/83203

[18] Y. Guo, C. Ten, S. Hu *et al*., "Preventive maintenance for advanced metering infrastructure against malware propagation," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1314-1328, May 2016.

[19] C. Sun, D. S. Cardenas, A. Hahn *et al*., "Intrusion detection for cybersecurity of smart meters," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 612-622, Jan. 2021.

[20] M. A. Faisal, Z. Aung, J. R.Williams *et al*., "Securing advanced metering infrastructure using intrusion detection system with data stream mining," in *Proceedings of Pacific Asia Conference on Intelligence & Security Informatics*, Berlin, German, May 2012, pp. 96-111.

[21] M. A. Faisal, Z. Aung, J. R. Williams *et al*., "Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: a feasibility study," *IEEE Systems Journal*, vol. 9, no. 1, pp. 31-44, Jan. 2015.

[22] Y. Zhang, L. Wang, W. Sun *et al*., "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Transaction on Smart Grid*, vol. 2, no. 4, pp. 796-808, Dec. 2011.

[23] F. A. A. Alseiari and Z. Aung, "Real-time anomaly-based distributed intrusion detection systems for advanced metering infrastructure utilizing stream data mining," in *Proceedings of 2015 International Conference on Smart Grid & Clean Energy Technologies*, Offenburg, Germany, Oct. 2016, pp. 148-153.

[24] A. Patel, H. Alhussian, J. M. Pedersen *et al*., "A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems," *Computers & Security*, vol. 64, pp. 92-109, Jul. 2016.

[25] M. Tavallaee, E. Bagheri, W. Lu *et al*., "A detailed analysis of the KDD cup 99 data set," in *Proceedings of IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, Canada, Jul. 2009, pp. 1-6.

[26] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in smart grid and smart home security: issues, challenges and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1933-1954, Apr. 2014.

[27] Z. E. Mrabet, N. Kaabouch, H. E. Ghazi *et al*., "Cyber security in smart grid: survey and challenges," *Computers & Electrical Engineering*, vol. 67, pp. 469-482, Apr. 2018.

[28] V. C. Gungor, D. Sahin, T. Kocak *et al*., "A survey on smart grid potential applications and communication requirements," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 28-42, Feb. 2013.

[29] C. Chen, K. Zhang, M. Ni *et al*., "Cyber-attack-tolerant frequency control of power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 2, pp. 307-315, Mar. 2021.

[30] E. Bou-Harb, C. Fachkha, M. Pourzandi *et al*., "Communication security for smart grid distribution networks," *IEEE Communication Magazine*, vol. 51, no. 1, pp. 42-49, Jan. 2013.

[31] R. Berthier, W. H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: requirements and architectural directions," in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications*, Gaithersburg, Maryland, Oct. 2010, pp. 350-355.

[32] S. McLaughlin, D. Podkuiko, and P. McDaniel, "Energy theft in the advanced metering infrastructure," in *Proceedings of International Workshop on Critical Information Infrastructures Security, CRITIS 2009*, Bonn, Germany, Sept. 2009, pp.176-187.

[33] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka *et al*., "Multi-vendor penetration testing in the advanced metering infrastructure," in *Proceedings of Annual Computer Security Applications Conference*, Honolulu, USA, Dec. 2010, pp. 107-116.

[34] C. Matthew. (2009, Feb.). Hacking AMI, SANS process control and SCADA security summit. [Online]. Available: https://www.inguardians.com/sans-scada-summit-presentations/.

[35] H. B. McMahan, E. Moore, D. Ramage *et al*., "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, USA, Feb. 2017, pp. 1273-1282.

[36] J. Konecny, H. B. McMahan, and D. Ramage. (2016, Oct.). Federated optimization: distributed machine learning for on-device intelligence. [Online]. Available: https://arxiv.org/abs/1610.02527

[37] Q. Yang, Y. Liu, T. Chen *et al*., "Federated machine learning: concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1-19, Jan. 2019.

[38] N. Srivastava, G. Hinton, A.Krizhevsky *et al*., "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, Jun. 2014.

[39] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, Jun. 2011, pp. 315-323.

[40] S. Ahmad, F. Arif, Z. Zabeehullah *et al*., "Novel approach using deep learning for intrusion detection and classification of the network traffic," in *Proceedings of International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Tunis, Tunisia, Jun. 2020, pp. 1-6.

[41] J. Ran, Y. Ji, and B. Tang, "A semi-supervised learning approach to IEEE 802.11 network anomaly detection," in *Proceedings of IEEE 2019 89th Vehicular Technology Conference*, Kuala Lumpur, Malaysia, Apr. 2019, pp. 1-5.

[42] Q. Zheng, J. Fang, Z. Hu *et al*., "Aero-engine on-board model based on batch normalize deep neural network," *IEEE Access*, vol. 7, pp. 54855-54862, May 2019.

[43] N. A. A. A. Al-Marri, B. S. Ciftler, and M. M. Abdallah, "Federated mimic learning for privacy preserving intrusion detection," in *Proceedings of 2020 IEEE International Black Sea Conference on Communications and Networking*, Odessa, Ukraine, May 2020, pp. 1-6.

[44] A. Maamarl and K. Benahmed, "A hybrid model for anomalies detection in AMI system combining *K*-means clustering and deep neural network," *Cmc-Tech Science Press*, vol. 60, no. 1, pp. 15-39, Jan.

2019.
[45] S. Choudhary and N. Kesswani, "Analysis of KDD-cup 99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT," in *Proceedings of International Conference on Computational Intelligence and Data Science*, Hyderabad, India, Mar. 2020, pp. 1561-1573.
[46] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Proceedings of 2015 International Conference on Signal Processing and Communication Engineering Systems*, Guntur, India, Jan. 2015, pp. 92-96.
[47] A. Y. Javaid, Q. Niyaz, W. Sun *et al*., "A deep learning approach for network intrusion detection system," in *Proceedings of 9th EAI International Conference on Bio-Inspired Information and Communications Technologies*, New York, USA, Dec. 2015, pp. 1-6.

**Haolan Liang** received the B.S. degree and M.S. degree in electrical engineering from Changsha University of Science and Technology, Changsha, China, in 2014 and 2017, respectively. Currently, she is a Lecturer in Hunan Institute of Engineering, Xiangtan, China, and she is pursuing the Ph.D. degree with Changsha University of Science and Technology. Her research interests include smart grid information security, and power system control and protection.

**Dongqi Liu** received the B.S. degree in electronic information engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2008, and the Ph.D. degree in control science and engineering from Hunan University, Changsha, China, in 2017. He worked in the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA, as a Visiting Researcher from 2015 to 2016. Since 2017, he has been with the Department of Electric Power Engineering, Changsha University of Science and Technology, Changsha, China. His research interests include electric vehicles, distributed energy system, and smart grid information engineering.

**Xiangjun Zeng** received the B.S. degree from Hunan University, Changsha, China, in 1993, the M.S. degree from Wuhan University, Wuhan, China, in 1996, and the Ph.D. degree from Huazhong University of Science and Technology, Wuhan, China, in 2001, all in electrical engineering. He worked as Post-doctoral Fellow in Xuji Relay Company, Xuchang, China, and the Hong Kong Polytechnic University, Hong Kong, China, and a Visiting Professor at Nanyang Technological University, Singapore, Singapore. He is now a Professor and Vice-Chancellor of Changsha University of Science and Technology, Changsha, China. His research interests include real-time computer application in power system control and protection.

**Chunxiao Ye** received the B.S. degree in Hunan Institute of Engineering, Xiangtan, China, in 2012, and the M.S. degree in software engineering from Central South University, Changsha, China, in 2016. Since 2019, he has been with the college of electric power engineering, Hunan Institute of Engineering, Xiangtan, China. His research interests include communication networks and machine learning.