Distributed IIoT Anomaly Detection Scheme Based on Blockchain and Federated Learning

Xiaojun Jin, Chao Ma, Song Luo, Pengyi Zeng, and Yifei Wei

***Abstract*—Anomaly detection in the industrial internet of things (IIoT) devices is significant due to its fundamental role in protecting modern critical infrastructure. In the IIoT, anomaly detection can be carried out by training machine learning models. Data sharing between factories can expand the data from which the model is trained, thus improving the performance of the model. However, due to the sensitivity and privacy of IIoT data, it is also difficult to build a high-performance anomaly detection model between factories. To address this problem, we design an anomaly detection method for IIoT devices combined blockchain of main-side structure and federated learning. We store the global model on the main-chain while the side-chain records the hash value of the global models and local models, which updated by participating nodes, controlling nodes access to the global model through the main-side blockchain and the smart contracts. Only the nodes participating in the current federated learning training can get the latest global model, so as to encourage the nodes to take part in the training of the global model. We designed a proof of accuracy consensus algorithm, and select the nodes to participate in training according to the accuracy of the local model on the test dataset to resist the poisoning attack of the models. We also use the local differential privacy (LDP) algorithm to protect user data privacy from model inference attacks by adding noise to the local model. Finally, we propose a new algorithm named Fed_Acc to keep the accuracy of the global model stable when the users add a lot of noise to their local models.**

***Index Terms*—Anomaly detection, blockchain, federated learning, IIoT, privacy protection.**

I. Introduction

INDUSTRY 4.0, also known as the Fourth Industrial Revolution, represents a new industrial era where the embedded devices used in intelligent production systems and processes are ultimately able to achieve the collaboration of the Internet, the Internet of Things (IoT), and the logistics network [1]. However, due to the existence of malicious nodes, the application of the Industrial Internet of Things (IIoT) faces serious security risks, which greatly impedes the development of the IIoT. For example, in the intelligent manufacturing scenario, if abnormal behavior (such as abnormal flow and irregular reporting frequency) occurs to industrial devices, such as engines with sensors, industrial production may be interrupted, thus causing huge economic losses to the factory [2]. In recent years, intelligent applications based on the IoT have been widely used in production and manufacturing, such as intelligent devices, intelligent transportation, etc., in which various data information collected by the sensor of the IoT devices will be uploaded to the server for centralized processing. The data can then be periodically uploaded to a cloud-based system via an intermediate server or gateway in a specific region [3]. However, this communication infrastructure is more vulnerable to security breaches, as untrusted cloud servers may want access to large amounts of sensitive data. Therefore, it is particularly important to provide security for these systems [4] to ensure the privacy and confidentiality of industrial data.

Meanwhile, these compromised sensors can send inaccurate data, and anomalies in IIoT devices may also expose users' sensitive data, which brings security and privacy threats to IIoT applications [5]. In order to distinguish between unreliable and reliable sensors (which sensors are faulty and which sensors are reliable), there needs to be some trusted infrastructure in which these problems can be seen [6].

If the manufacturer does not provide a complete security system for the devices, once it encounters external malicious attacks, among them, sensitive information and data privacy are likely to be leaked maliciously, so the related security protection issue has gradually become a challenge hot spot. In the face of increasing network security threats and expanding attack surface, the security of IIoT network and environment becomes more and more complex and challenging [7]. In order to solve the IIoT devices anomaly problem, the typical approach is to conduct anomaly detection on the affected IIoT devices. At present, the most popular technology is artificial intelligence (AI). AI constantly improves the anomaly detection model and realizes automatic detection through machine learning and deep learning. However, any anomaly detection technology based on machine learning and deep learning is faced with challenges, such methods have training data scarcity and privacy issues [8]. Anomaly detection methods based on machine learning or deep learning require a great deal of data for model training, which may expose privacy. The anomaly detection algorithm based on federated learning can coordinate model training among multiple parties without exposing data privacy.The federated learning framework was originally proposed by [9] to train the model without the user uploading their own local data samples, thus protecting the user's privacy. Federated learning provides not only privacy protection for the client, but also ultra-high learning performance, all due to the model parameters of the local

Manuscript received October 15, 2023; approved for publication February 17, 2024; approved for publication by Chen, Chien-Ming Division 3 Editor, March 26, 2024.

X. Jin, P. Zeng, and Y. Wei are with Beijing University of Posts and Telecommunications Ringgold standard institution - School of Electronic Engineering, Beijing China, emails: {jinxiaojun, zengpy, weiyifei}@bupt.edu.cn.

C. Ma and S. Luo are with China Academy of Information and Communications Technology Ringgold standard institution, Beijing, China, emails: {machao, luosong}@caict.ac.cn.

C. Ma is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2024.000016

1229-2370/24/$10.00 © 2024 KICS

computation used to build the final model after summarizing all of them without the client having to transmit the local training data [10]. However, there is no doubt that federated learning still has serious challenges. For example, federated learning features model gradient aggregation by a central node. Therefore, there are great security risks in the process of polymerization. One is that the central node cannot be fully trusted, and the client may refuse to participate in cooperation because of distrust of the central node, and the other is the inability to obtain transparent operational information about the central node. Blockchain can be used to solve federated learning's challenges. Blockchain perfectly complements federated learning's security issues, and the two complement each other, providing better security, privacy, interoperability, scalability and reliability by enabling data traceability, consensus mechanisms to ensure that data cannot be tampered with, avoidance of single points of failure, extension to public networks where there may be untrusted nodes, and provide incentive mechanism [11].

Due to various problems in anomaly detection of smart factory devices in IIoT, a federated learning architecture combined with blockchain is designed in this paper which can also protect the data privacy of the factory. Since the training of anomaly detection model requires local data from each smart factory, there is a risk of privacy disclosure in the process of federated learning and training. Therefore, security can be provided to the system through the blockchain, and the anomaly detection model can be trained through differential privacy (DP) in a way that protects user privacy. Contributions from this paper are as follows:

1) We propose the IIoT anomaly detection architecture combined federated learning and the main-side blockchain structure, store the global model on the main-chain, record the hash of the global model and node local model on the side-chain, and design an access control mechanism to control nodes' access to the global model on the main-chain. As for encouraging clients to actively participate in the federated learning training , we set that only the client participating in the training can get the latest global model from the main-chain.
2) We propose the proof of accuracy (PoA) consensus algorithm, which test the accuracy of the local model uploaded by nodes on the side-chain and nodes whose accuracy are higher than the threshold were selected to take part in the federated learning training, while local models with lower accuracy were prohibited from participating in the training of this round.
3) We propose the Fed_Acc algorithm which uses a sliding window to judge whether the loss value of the global model decreases in the overall trend during the federated learning and training process, so as to resist the poisoning attack of the model. To ensure the data privacy of smart factories, clients can use local differential privacy (LDP) to add noise when uploading model parameters. In this paper, simulation verifies that the global model has better anti-noise ability than Fed_Avg when the local model adds more noise.

II. Related Work

At present, the IIoT has a large number of sensor devices, and the security system of many devices has a certain delay, it is likely to be unable to deal with novel network attacks. Therefore it is necessary to apply anomaly detection for unknown attacks. Each factory may not have the same type of attack, and the data generated is not the same, so multiple factories can use federated learning to expand the dataset and cooperate to complete anomaly detection. Blockchain and federated learning have been widely used in neural network training. Federated learning has successfully decentralized data, enabling clients to independently train local datasets and upload trained model parameters without uploading local data to the central server. Finally, all uploaded model parameters are aggregated, and the final model is obtained after multiple rounds of iterative convergence. This ensures the data privacy security of each client. The characteristics of blockchain, such as decentralization, verifiability and traceability, help to reduce attacks on federated learning, thus improving the accuracy of the model. Meanwhile, in order to make the model more perfect, the incentive mechanism makes each factory node actively take part in the training. There are many federated learning architectures based on blockchain. Li *et al.* [10] proposed blockchain-based federated learning (BCFL) architecture, where blockchain, completely decentralized and protected by privacy, is mainly used as the central database of federated learning system. The primary goal of the BCFL is to protect the local privacy datasets against malicious nodes by fairly rewarding nodes according to the quality of their contributions. Abbas [12] et al. used federated learning to build a cluster-based threat search system for anomaly detection, named Block-Hunter, which can automatically search for external attacks in blockchain-based IIoT networks. In order to solve the problems of device asynchrony and resource consumption in federated learning, Cao cite9500737 et al. designed a new consensus mechanism and built a framework of federated learning weighted by blockchain (DAG-FL) based on direct acyclic graph (DAG). For member inference attacks such as model theft attacks, model reconstruction attacks and other privacy attacks in federated learning, Jia *et al*. [13] proposed three solutions: K-means clustering, random forest, and AdaBoost which are all distributed and based on DP and homomorphic encryption, and combine these methods with federated learning and blockchain.

In the study of federated learning and anomaly detection, Taheri *et al*. [14] proposed an architecture Fed-IIoT, introduced Android malware detection algorithm, including a variety of equally independent learning model distribution, proposed two kinds of potential random variable based poison attacks, using generative adantagonistic network (GAN) and federated learning to float malware on benign data samples, viz. GAN and Federated GAN (FedGAN). A GAN network (A3GAN) defense algorithm combined aggregation federated learning and GAN algorithms is proposed to avoid aggregation anomalies and detect adversaries in server-side components. Compared with the centralized ML method, [15] uses federated learning training on the device to protect the data privacy of the

terminal device not only has achieved higher accuracy, but also minimized false alarms. The advantages of integrating federated learning with an integrator for optimal results are demonstrated. As for solving the problem of network attacks and sample dissimilarity, Wu *et al*. [16] proposed a model combining hybrid Gaussian variational self-coding network and federated learning for anomaly detection classification. In the MGVN network model, the variational autoencoder is constructed using the mixed Gaussian structure, and then features are extracted from the input data in order to construct the depth support vector network.

Privacy in federated learning is also important. In [17], a new federated learning based on converter architecture, which is called FedAnomaly, is proposed to timely implement privacy protection exception detection in cloud manufacturing. As to details, for the edge devices, a transformer based weakly supervised anomaly feature extraction model is designed, and the devices upload the feature with DP noise. Meanwhile, the MLPS model is used to detect exception for the cloud side. Because of the differences between transformer training and traditional device training, Ma textitet al. [17] customized a new federated-learning training protocol for the anomaly detection model, which minimizes communication overhead by uploading code features and losses instead of model parameters. In [18], an unsupervised model called TCN-ACNN was constructed to detect anomalies deeply in system logs and learn time representations. To ensure collaboration and privacy between IoT devices, a new federated learning algorithm has been developed to improve the TCN-ACNN. A masking strategy based on the lottery hypothesis is designed to enhance customization of federated learning and reduction in communication overhead when dealing with non-IID datasets.

Although the above researchers have made the latest development and research results, the FedAvg federated average algorithm, which is directly used in their aggregation algorithm, does not consider the problem of data heterogeneity among multiple factories. The data generated between multiple factories is likely to be heterogeneous data. How to deal with heterogeneous datasets in the federated learning training process is also a hot research direction. In terms of heterogeneous data processing, Wu *et al*. [19] selects nodes by calculating the inner product of local model and global model gradients, and then removes unfavorable local updates that will affect the overall data heterogeneity, thus changing the relationship between local gradients and global gradients. However, Zhang *et al*. [20] proposed a new federated learning framework named centroid distance-weighted joint average (CDW_FedAvg). One of the key points is the centroid distance. Specifically, each node's dataset has two classes, positive and negative. The centroid distance takes into account the distance between the two classes. However, it is not considered that the centroid distance of the dataset will reveal the data privacy of the client.

Incentive in federated learning is also a hot research trend. Multitudes of literature revolves around designing joint learning incentives through customer contributions, which has two attributes that are most valued: data quantity and data quality [21]. To incentivize honest clients to upload high-quality models, while promptly detecting and punishing malicious clients, Bao *et al*. [22] proposed a federated learning ecosystem to equitably distribute profits, called FLChain. This ecosystem has the benefits of a typical blockchain: decentralized, trusted, auditable, and incentivized. To keep incentives safe, some federated learning works [22], [23], encourage clients to place the parameters of their training models in the blockchain and reward them with cryptocurrency. However, the current incentive mechanism rarely considers the problem of security. Data providers may provide low quality data or false data. In this case, how to allocate incentives is a problem worth studying.

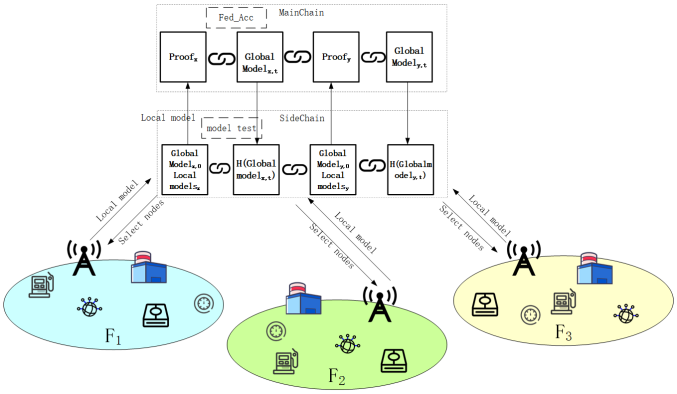


Fig. 1. System architecture.

The role of blockchain in IIoT is mainly to improve system security. Matthew *et al*. [24] implemented a collaborative IDS framework to enable secure data transactions in IIoT systems. Rathee *et al*. [25] utilized the Viterbi algorithm to implement a blockchain-based IDS framework to ensure security in the IIoT. Hewa *et al*. [26] designed a new security architecture based on blockchain and fog computing to improve the security of IIoT cloud networks.

At present, there are few studies on the application of BCFL architecture for anomaly detection in IIoT. In this paper, a federated learning anomaly detection system based on blockchain architecture is designed, which uses LDP to ensure the protection of the node privacy, and considers the problem of model accuracy decline caused by differential privacy, and designs a new federated learning model aggregation algorithm. The effect of noise is reduced and our algorithm is verified by simulation.

III. System Model

In this paper, we propose a detection model for IIoT device traffic anomaly combined federated learning and blockchain, which includes modules such as main-side blockchain structure, PoA consensus algorithm, Fed_Acc federated learning algorithm, and main-side blockchain access control smart contract. Let's take a look at the overall architecture of this paper's model.

A. Anomaly Detection Architecture of IIoT based on Double-blockchain

The system structure diagram is shown in Fig. 1 contains three smart factories F_1, F_2, and F_3, each of which

contains multiple data-generating devices. The three smart factories conduct federated learning through blockchain. Blockchain adopts the double-chain architecture of the main-side blockchain. The main-chain records the smart factory global model and the access permission proof of nodes while recording local model parameters, the initial global model and the hash of global model on the side-chain. The role of the side-chain is to select the consensus node of the main-chain. Only the nodes that upload the local model and pass the model test can become the consensus node, obtain access to the data of the main-chain, and then participate in federated learning training to obtain the trained global model. The nodes that do not participate in the global model training can only obtain the initial global model. The subscripts x and y of $GlobalModel_{x,0}$ and $GlobalModel_{y,t}$ in the figure are used to distinguish different global models, 0 and t represent different iterations, 0 represents the initial globalmodel that has not been trained, and t represents the globalmodel after t iterations trained by federation learning. We use Fed_Acc algorithm to train and update the model, so as to finally get a global model to detect anomaly, which can be used by each smart factory to detect anomaly. After the global model is obtained, the side-chain records the hash value of the global node. Because the attacks on each factory and the anomaly data on the local device may be different, the global model through federated learning across multiple factories helps them detect anomaly that are not present locally.

B. Federated Learning

In this section, we propose an operational framework for federated learning. The smart factory uses local difference privacy (LDP) to disturb the local model and resist the model inference attacks when training data locally. In order to solve the model poisoning attack that may occur during the training iteration process, we proposed a new algorithm named Fed_Acc to redistribute the local model according to the accuracy of the aggregation model. We also designed the access control smart contract of the main-side blockchain to realize the node's access control to the main-chain.

1) initialization: In the initialization phase, we consider the set of factories as $K_i \in \{K_1, \cdots, K_n\}$, dataset is $D_i \in \{D_1, \cdots, D_n\}$, all local models are $w_i \in \{w_1, \cdots, w_n\}$, the global initial model to be trained is W_0 (the subscript represents the number of iteration rounds, and 0 means no training has been conducted).

2) Local model training: K_i, a smart factory, obtains the global model W_t (t represents the number of iterations) from the blockchain and uses the local privacy dataset D_i to train the model on the local device to get the local model. The learning objective of local client K_i is finally to get it's own best model parameter w_t^j that minimizes the objective function $l\left(w_t^j, x, y\right)$,

$$\min l = \sum_{t=0}^{n} l\left(y_t^j, f\left(x_t^j\right)\right), \tag{1}$$

where l represents the loss function, j represents one of the smart factorys, $\left(x_t^j, y_t^j\right)$ is the time series data generated by the smart factory j at time t, x_t^j is the feature vector, y_t^j is the label, $f(\cdot)$ represents the machine learning model, and n represents the size of the local dataset.

We use the binary cross entropy loss function to train the anomaly detection model. We set the label of abnormal data to 1, $y_a = 1$, and the label of normal data to 0, $y_n = 0$. The local loss function defined by us is shown in (2):

$$l(w, x, y) = -\sum_{t=0}^{n} y_t \log \sigma(x_t) + (1 - y_t) \log(1 - \sigma(x_t)), \tag{2}$$

$\sigma(\cdot)$ is the sigmoid function.

In this paper, we use the differential privacy to protect model parameters w_k by adding a random gauss perturbation of model parameter, in order to guarantee (ε, δ)-LDP the LDP, local produce random noise n follow gaussian distribution $N\left(0, \delta^2\right)$, and then to get local model after disturbance. The perturbation δ^2 satisfies the following conditions $\sigma^2 \geq \frac{2\log(1.25/\delta)}{\varepsilon^2}\Delta^2$, Where represents the norm sensitivity. For the function, we assume $S = \|f(D) - f(D')\|_2 \leq \Delta_f$. The sensitivity S of the function is limited by Δ_f, where D and D' are two adjacent datasets. If and only if $\varepsilon = \frac{\Delta_f}{\sigma}\sqrt{2\log\frac{1.25}{\delta}}$ Gaussian noise satisfy (ε, δ)-LDP [27], in this paper, local difference privacy (LDP) is used, the node adds Gaussian noise to local model and then uploads the local model.

$$w_j \leftarrow w_j + N\left(0, \sigma^2 S^2\right) \tag{3}$$

Note that since LDP is used, there is no limit to how much noise a user can add to the local model. If user k adds a lot of noise, he can of course protect the privacy of his own data well, but this will lead to poor model effect, and then affect the global model. Therefore, we designed a new model aggregation algorithm Fed_Acc to alleviate the impact of local model noise on the whole region model.

3) Fed_Acc algorithm: In general, the global objective function in federated learning is:

$$F(w) = \sum_{i=1}^{|K|} \frac{D_i}{\sum_{i=1}^{|K|} D_i} l_i(w, x, y), \tag{4}$$

for $i \in K, l_i(w, x, y)$ is used to measure the loss function based on local dataset D_i. Fed_Avg algorithm is the most commonly used gradient aggregation process. Firstly, the gradient W_{t-1} at time $t-1$ is obtained, and each node participating in the training optimizes its target by local stochastic gradient descent (SGD).

$$w_t^i = w_{t-1}^i - \eta \nabla F_i\left(w_{t-1}^i\right) \tag{5}$$

η is the learning rate and $\nabla F_i(\cdot)$ is gradient of node i.

In this paper, firstly, some models are excluded in the model test stage of the side-chain, and at the same time, the weights of all models are changed in the model aggregation stage. In this way, the influence of some models on the convergence rate can be slowed down without losing the information of these models. In [28], the federated learning study the

measurement method of the fairness and effectiveness of α-fairness measurement method is proposed, and put forward the optimal solution of model weights are as follows:

$$p_j = \frac{a_j^{\frac{1-\alpha}{\alpha}}}{\sum_{i=1}^{K} a_i^{\frac{1-\alpha}{\alpha}}}, \tag{6}$$

where a_j is the accuracy of node j,p_j is aggregation of weight for the node j model, α is the parameter of α-fairness [28]. In order to reduce the impact of data heterogeneity and improve the performance of the model, we propose a new aggregation algorithm Fed_Acc. The aggregation node downloads all the client model w_t^j from the blockchain, tests the model w_t^j with the common dataset, and obtains the trained model accuracy acc. (6) is used to assign the weight of the client model parameters. For the purpose of reducing the communication consumption of the aggregation model, we further processed a local model in each round. When the local model's $loss_j^t > loss_j^{t-1}$ of j node, we would not choose this local model for global aggregation in this round of training. This can not only reduce the calculation amount of model aggregation, but also eliminate the model with poor quality and resist the model poisoning attack. The participating nodes pass their update model $\Delta_i^t = w_i^t - w_i^{t-1}$ to the aggregation node, which updates the global model as (7):

$$\Delta^t = \sum_{j \in S_t} p_j \Delta_j^t, \tag{7}$$

$$w^t = w^{t-1} + \Delta^t, \tag{8}$$

where p_j is the aggregation weight of the model corresponding to node j, which can be obtained from (6). Through the consensus algorithm, the node with the highest weight is found as the consensus node to aggregate the global model and broadcast to the blockchain.

When the model was updated, we designed a sliding window $B = \{b_0, b_1, \cdots, b_d\}$, to judge that the loss of several updates of a node is decreasing on the overall trend. The window size is d, which records the loss value of a local model tested on the common dataset in the latest round d. The window size and threshold limit the loss floating size of the node training model. If the window size is larger than the threshold, the overall loss trend is rising, which can be assessed as a malicious model. In this paper, we set the threshold to 0.

Let's first calculate before we aggregate the model:

$$s = \sum_{i=0}^{d-1} b_{i+1} - b_i, \tag{9}$$

b_i is the model loss value at time i.s represents the sum of the difference between adjacent elements in the window to evaluate whether the loss of the model rises or falls in round d. If the value of d is 2, it means to evaluate whether the current loss is lower than that of the last round each time. If it is lower, it will not be included in the reference range for model updates, that is, the aggregation weight of the updated local model will be set to 0. This causes the global model to stop training in subsequent rounds once all models have lost more than the previous round. If the value of d is too large, the algorithm will not respond immediately once a node launches a model poisoning attack and reset the corresponding model weight to 0, resulting in serious pollution of global model parameters. A similar method is also used in [29], which directly eliminates nodes with substandard accuracy in the training process. In our study, sliding windows were added instead of directly eliminating models with low elimination accuracy, so that the training process included those models that added too much noise but also contained meaningful information. The detailed process of Fed_Acc is shown in Algorithm 1.

Algorithm1: Fed_Acc

procedure SERVER AGGREGATIONS
  $Input : \{clientsModel_j\}_{j \in k}$
  for $each \quad round \quad t = 0, 1, 2 \cdots$ **do**
    $\{acc_j\}_{j \in \{K_n\}}^t \{loss_j\}_{j \in \{K_n\}}^t \leftarrow TestModel()$
    for $each \quad acc_j, loss_j$ **do**
      if $s \quad in \quad (9) \geq 0$ **then**
        $delete \quad LocalModel_j \quad in \quad this \quad epoch$
      end if
      $p_j \leftarrow \frac{acc_j^{\frac{1-\alpha}{\alpha}}}{\sum_{i=1}^{K} acc_i^{\frac{1-\alpha}{\alpha}}}$
      $w_j \leftarrow UpdateGlobalModel()$
    end for
  end for
end procedure
function $sub\_window(loss_j^t)$
  result = 0
  for $loss_i, loss_{i-1} \quad in \quad loss_j^t$ **do**
    if $i - 1 \geq 0$ **then**
      $result += loss_i - loss_{i-1}$
    end if
  end for
  $return \quad result$
end function
procedure CLIENT TRAINMODEL
  Input: Local data
  $initialize \quad local \quad minibatch \quad size : B, local \quad epochs : E, learningrate : \eta$
  for $each \quad epoch \quad i \in E$ **do**
    $samples \quad S_i \quad based \quad on \quad B$
    $w_i \leftarrow w_{i-1} - \eta \bigtriangledown g(w_{i-1}, S_i)$
    $w_i \leftarrow w_i + N(0, \sigma^2 S^2)$
  end for
end procedure

C. Proof of Accuracy Consensus Algorithm

In this IIoT scenario, traditional permissioned-blockchain consensus algorithms such as PBFT, Paxos and Raft, which automatically elect the primary node and package and broadcast blocks, are not necessarily appropriate for consortium blockchain systems. The reasons are as follows. First, in federated learning, nodes participating in the training process

are able to independently decide whether to use their own local private datasets for training to obtain a local model. Therefore, some nodes will participate in the consensus while others will not. Some nodes can be evil nodes, PBFT has a fault tolerance of 1/3, and Raft algorithms require participants to be completely trusted. In the scenario of this paper, nodes are likely to upload fake models to carry out model poisoning attacks, and consensus algorithm should be able to resist such attacks. Therefore, this paper designs a Proof of Accuracy (PoA)consensus algorithm to select the nodes of model aggregation and ensure that the local model uploaded by the nodes participating in federated learning is safe and reliable.

Assume that the number of nodes taking part in the consensus algorithm is m each time, and all nodes participating in the consensus are $\{K_m\}$. We choose the nodes that participate in the consensus and the nodes that aggregate the model. The detailed steps are as follows

Step1: Firstly, multiple smart factories form a blockchain system with double-chain structure of main-side blockchain, and an initial global anomaly detection model is disclosed. Intelligent factory j uses local dataset D_j to train the global model $LocalModel_j$, and then uploads the local training model to the side-chain, which collects the model $\{LocalModel_j\} \in \{K_m\}$, triggers the smart contract using the common dataset D_{pub} on the side-chain to test the local model uploaded by the intelligent factory j. Obtain the model accuracy, the model accuracy is higher than the threshold γ node selected as the consensus to participate in the main-chain of the next round of consensus. The primary node is elected by finding the one with the highest accuracy.

Step2: The primary node executes the federated aggregation algorithm, starts model iterative aggregation with other nodes, and finally gets a new global model.

Step3: The primary node records the global model in the main-chain, and records the hash of the global model in the side-chain.

This completes a PoA consensus and federated learning training process. Different from PBFT algorithms, PoA algorithms require nodes to actively participate in the consensus process rather than passively select the primary node. Nodes that do not participate in this consensus will not get the latest master chain data, and only the old master chain is stored locally, that is, only the old global model. The accuracy process of side-chain calculation can resist model poisoning attack. After the accuracy of side-chain calculation, the model with low accuracy is eliminated, and the remaining nodes start to carry out the iterative process on the main-chain until the specified number of rounds is stopped, and each updated global model is recorded in a block of the main-chain. The result has 2 blocks, a block on the side-chain that holds all the original local models, and the other block on the main-chain that holds the final global model.

D. Main-chain Access Control Scheme

The main-chain access in this paper is controlled by smart contracts. After the side-chain verifies the node's local model, it gives the node the corresponding access control rights.

Algorithm2: Proof of acc

procedure SELECT NODES
  $Input : \{clients\}$
  for $client_j \quad in \quad \{clients\}$ **do**
    $checkclient_j$
    $Upload \quad hash(localmodel_j) \quad on \quad sidechain$
    $\gamma_j = Testmodel(localmodel_j)$
    if $\gamma_j < r$ **then**
      $Delete \quad client_j$
    end if
  end for
end procedure
procedure UPLOAD GLOBALMODEL
  $Input : GlobalModel$
  $W_t \leftarrow Update(GlobalModel)$
  $Upload \quad W_t \quad on \quad mainchain$
  $Upload \quad hash(W_t) \quad on \quad sidechain$
end procedure

First, the node uploads the local model $\langle GID, LocalModel \rangle$ through the smart contract, and the GID is the globally unique identifier of the node. The smart contract waits for the node to upload the local model within T time after the time stamp of the last block. After T time, $TestModel\,()$ is tested on all the collected models to get the accuracy and eliminate those models whose accuracy is not up to the standard.

After passing the verification contract of the side-chain, $Proof$ will be formed on the side-chain, and the $Proof$ information is $\langle \{GID\}, \{LocalMode\}, \{H\,(LocalModel)\}, BlockHeight \rangle$. $BlockHeight$ is the block height of the side-chain, and $\{\cdot\}$ represents the set. The node uses $Proof$ as the input of the master chain smart contract to trigger the master chain access control smart contract, which stores the transaction as the current $Proof$ access control policy. The next node starts to access the main-chain. The node enters $\langle GID, LocalModel, SignedLocalModel \rangle$ into the access control smart contract of the main-chain, and SignedLocalModel is the model signed by the node with the private key. The main-chain verifies that the input data of the node meets the $Proof$ access control policy, verifies that the signed model is valid, and if so, the node obtains the access control permission of the main-chain.

The main-chain only validates the $Proof$ of the current side-chain $BlockHeight$, so if a node is verified with the previously generated $Proof$, it will not pass.

The smart factory sends its own $Proof$ information to the main-chain, triggering the smart contract. The smart contract verifies the $Proof$ sent by the smart factory and verifies $SignedLocalModel$. $SignedLocalModel$ is signed by the local private key for the smart factory, and the smart contract is verified by the public key corresponding to the GID. For the node that fails to pass the verification, the smart contract will directly execute the $return$, and the node that passes the verification will continue to execute the following smart contract. Nodes that are not authenticated will not execute subsequent smart contracts and will not have access to the

Algorithm3: Access control contract

procedure $Sidechain \quad access \quad control \quad contract$
  $Input :< GID, LocalModel >$
  $accuracy = TestModel(LocalModel_{GID})$
  if $accuracy \geq Threshold$ **then**
    $\{GID\}.append(GID)$
    $\{LocalModel\}.append(LocalModel)$
    $\{H(LocalModel)\}.append(H(LocalModel))$
  else
    return
  end if
  $Proof \leftarrow < \{GID\}, \{LocalModel\}, \{H(LocalModel)\}, BlockHeight >$
  $Output : Proof$
end procedure
procedure $Mainchain \quad access \quad control \quad contract$
  $Input : Proof, \{GID_j, LocalModel_j, SignedLocalModel_j\}_{j\in\{K_n\}}$
  for $each \quad client$ **do**
    $bool = Verify(Proof, \{GID, LocalModel, SignedLocalModel\}_{j\in\{K_n\}}$
    if $bool == True$ **then**
      $continue$
    else
      return
    end if
  end for
  $Execute \quad Algorithm1 : Fed\_Acc$
end procedure

main-chain.

The access control contract is shown in Algorithm3. The blocks in the main-side blockchain are one-to-one correspondence, each node stores all the side-chain blocks, the node that participates in training every time stores all the main-chain blocks, and the node that participates in training occasionally stores some main-chain blocks,only these nodes can get the latest global model. The node that does not participate in this round can only get the global model that he participated in the model aggregation last time. If a node has never participated in federated learning training, only the initial model can be obtained.

E. Security Analysis

1) Access control security: Adversary model. Adversary is a smart factory that does not have relevant training data but wants access control of the main-chain to get the latest anomaly detection model. Access control is divided into two parts: side-chain access control and main-chain access control.

Side-chain access control: Smart factories can only send access control requests to the side-chain through smart contracts, and the GID in the request is one-to-one corresponding to the localmodel, and only the localmodel that satisfies a certain accuracy rate can access the side-chain. If an adversary uses its own GID to make an access control request, it needs the relevant data to train the model and get a qualified localmodel. This is not consistent with our hypothesis assume.

Main-chain access control: If the opponent does not pass the side-chain access control, he cannot forge the signature, and therefore cannot get the main-chain access control permission.

2) Blockchain fork: According to the PoA consensus designed in this paper, aggregation nodes are selected from the nodes participating in training to aggregate the local model and generate the main-chain block. The nodes generating blocks in each round are fixed, so there will be no bifurcation.

F. Incentive Mechanism

Existing blockchain-based federated learning schemes assume that every client is willing to actively participate in model training, but this does not necessarily hold true in an IIoT environment, so these schemes need to provide certain incentives for eventual implementation. This paper does not use token as incentive like other literatures [23], but to obtain the latest global model as incentive. In essence, the scheme in this paper extends the data quantity and labels of the federated learning algorithm, and only the nodes participating in training can obtain the access permission of the main-chain, and then participate in training to obtain the latest model, and update their own model parameters after obtaining the latest model. The trained global anomaly detection model has higher accuracy, which motivates smart factories to participate in the federated learning training. In this paper, the side-chain is jointly stored by all nodes. In the case that not every node take part in every round of the process of training, each block of the main-chain is stored by some nodes, but the block stored by all nodes can still be combined to obtain a complete main-chain.

IV. Performance Evaluation and Discussion

The simulation tool of this paper adopts python3.7, tensorflow2.0, the CPU is Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz, GPU is NVIDIA GeForce RTX 3060, and the dataset is Skoltech anomaly benchmark (SKAB) dataset [30]. SKAB is an anomaly benchmark designed to evaluate anomaly detection algorithms. SKAB designs for evaluating the anomaly detection algorithms. SKAB allows working with two main problems (there are two markups for anomalies): 1. Outlier detection (anomalies considered and marked up as single-point anomalies). 2. Changepoint detection (anomalies considered and marked up as collective anomalies). And the experimental data in SKAB has two markups for anomalies, so two kinds of problems can be handled.

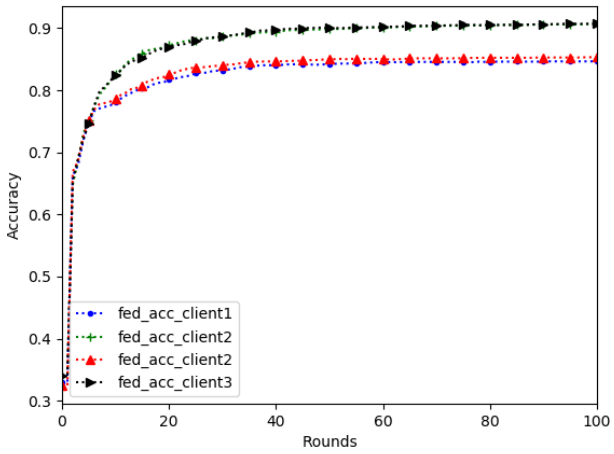
1) When anomalies are detected, it is marked as single-point anomalies.
2) When anomalies are detected, it is marked as collective anomalies.

The first is outlier detection and the second is change-point detection. The data properties in both tests are shown in Table I.

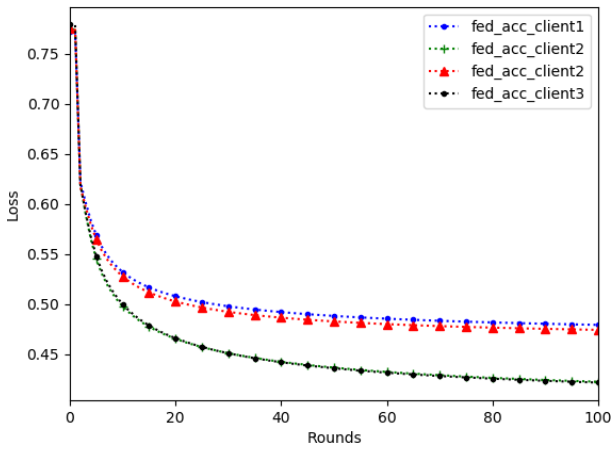
Our global model adopts logistic regression model, with the learning rate set as 0.02, epoch set as 100 and batch_size set as 200.

TABLE I
PROPERTIES FOR SKAB DATASET.

Properties	Description
datetime	Identifies the date and time when the value was written to the database (Unit: YYYY-MM-DD hh:mm:ss)
Accelerometer1RMS	Represents the vibration acceleration (Unit: g)
Accelerometer2RMS	Represents the vibration acceleration (Unit: g)
Current	Represents the amperage on the electric motor (Unit: Ampere)
Pressure	Represents the pressure in the loop after the water pump (Unit: Bar)
Temperature	Represents the temperature of the engine body (Unit: The degree Celsius)
Thermocouple	Shows the temperature of the fluid in the circulation loop (Unit: The degree Celsius)
Voltage	Represents the voltage on the electric motor (Unit: Volt)
RateRMS	Shows the circulation flow rate of the fluid inside the loop (Unit: Liter per minute)
anomaly	0 represents the point is failure-free, 1 represents the point is anomalous
changepoint	0 represents the point is an outlier for single-point anomalies, 1 represents the point is a changepoint for collective anomalies



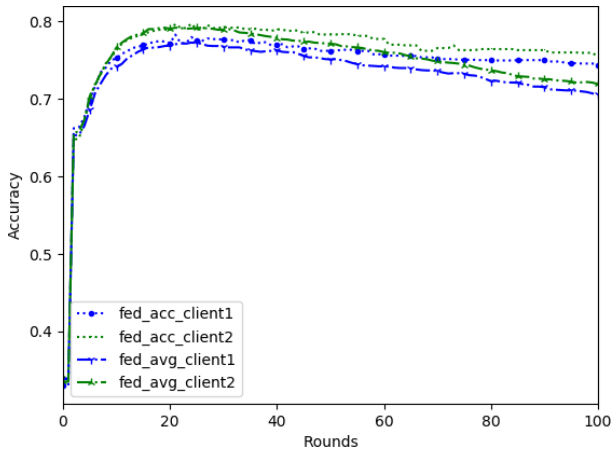
(a)



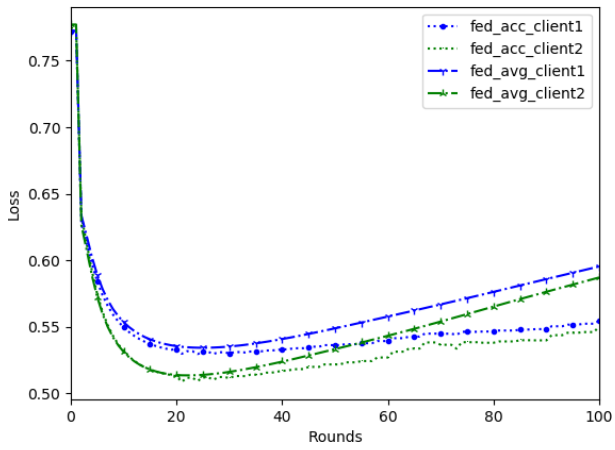
(b)

Fig. 2. Model accuracy and loss with $\sigma^2 S^2 = 0.1$.

a. We set the local Gaussian noise added by the four clients as $N(0, 0.1)$ and the size of the sliding window as 3. After that, Fed_Avg and Fed_Acc algorithms were respectively used to conduct federation learning training to obtain the simulation diagram as shown in Fig.2 It can be seen that in the case of low noise, the training process of all nodes is not affected much, the rising trend of accuracy and the declining trend of loss are relatively smooth, and eventually convergence will be successful.



(a)



(b)

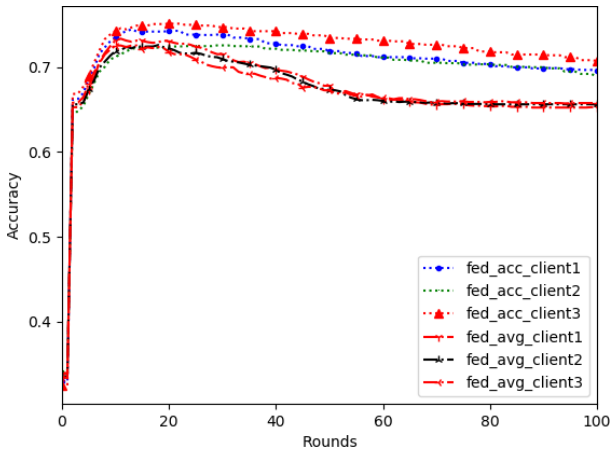
Fig. 3. Model accuracy and loss with $\sigma^2 S^2 = 0.3$.

b. We set the local Gaussian noise added by client1 and client2 as $N(0, 0.3)$, and then conduct federation learning training with Fed_Avg and Fed_Acc algorithms respectively to obtain simulation diagrams as shown in Figs. 3(a) and 3(b).

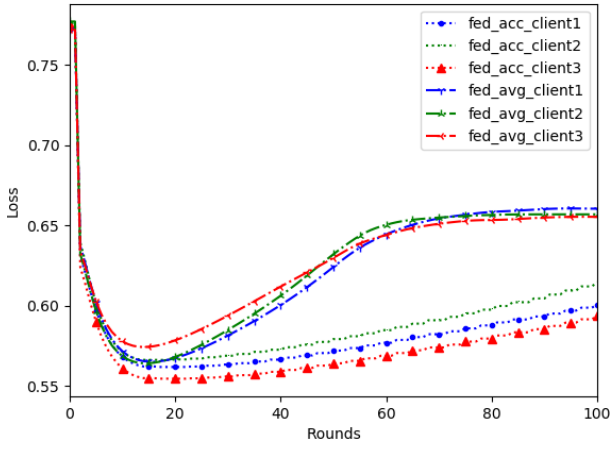
As can be seen from Fig.3, when the Gaussian noise added by the user is very high, the accuracy of the two algorithms tends to decline in the late training iteration due to the high noise. The accuracy of the global model of Fed_Avg algorithm will decrease significantly and the loss will increase quickly. In contrast, our Fed_Acc scheme can resist the influence of user noise on the global model to a certain extent. Even if the user adds large noise to protect privacy, the accuracy of the global model is higher than that of Fed_Avg.

c. In the case of evil nodes, we add Gaussian noise of $N(0, 0.8)$ to client1 in the 20th iteration. The simulation results are shown in Figs. 4(a) and 4(b).

It can be seen from Figs. 4(a) and 4(b) that in the case of malicious nodes, the accuracy of Fed_Avg algorithm will decline sharply, eventually converging around 0.64, and the

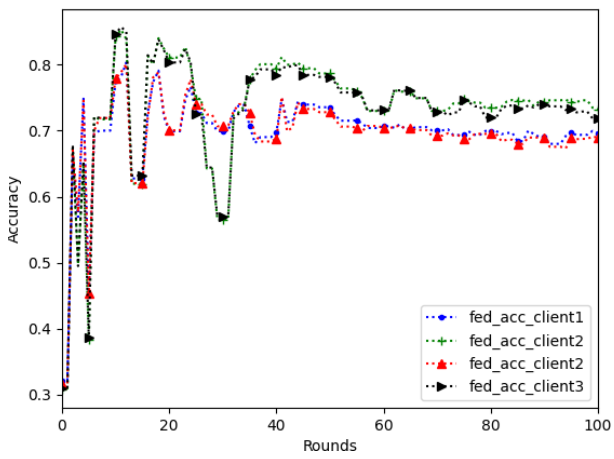


(a)

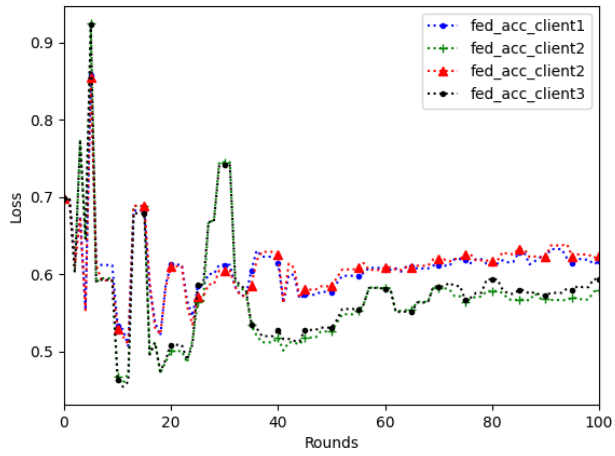


(b)

Fig. 4. (a) is the Comparison of algorithm accuracy in case of evil node and (b) is the loss comparison in case of evil node.



(a)



(b)

Fig. 5. Model accuracy and loss with $\sigma^2 S^2 = 0.3$ use BP neural network.

loss will also reach a very high value, and finally reach a flat value around 0.65. Compared with Fed_Avg, the global model obtained by Fed_Acc has higher accuracy and lower loss, so Fed_Acc has better ability to resist model poisoning attacks.

d. We try to use different models such as BP neural network for federated learning training. Figs. 5(a) and 5(b) respectively show the training results of BP neural network. The BP neural network is a four-layer neural network. The first layer consists of 8 units corresponding to the 8 features. Then, there are two hidden layers that both have 200 units. The output layer is a Softmax function to classify normal data and abnormal data. Note that the hidden layers apply ReLu function as the activation function.

The learning rate is set to 0.02.It can be seen that compared with LR model, the fluctuation of BP neural network is larger, but it can also converge at last. We can choose different training models in different IIoT environments. Under the SKB dataset in this paper, it is obvious that LR model has better effect.

Next, we conducted experiments on the influence of sliding window size on training results. Fig. 6(a) is the simulation result of LR model and Fig. 6(b) is the simulation result of BP neural network.

As can be seen from Fig. 6(a), all the accuracy after sliding window is added are higher than the one without sliding window, and $d = 0$ is equivalent to no sliding window. Among all the results, the model accuracy obtained by training is the highest when $d = 20$. Therefore, with the increase of the sliding window size, the model accuracy increases first and then decreases, which is also the same as our previous analysis. The window size should not be too big or too small.

A similar conclusion can be drawn from Fig. 6(b). In Fig. 6(b), the accuracy of BP neural network still fluctuates greatly, but the accuracy of different window sizes is similar to that of Fig. 6(a), and the highest accuracy is still obtained when $d = 20$.

V. Conclusion

This paper studies the IIoT device anomaly detection system based on blockchain and federated learning. It makes innovations in the federated learning training process, consensus algorithm and incentive mechanism, and improves the accuracy of the federated learning algorithm while considering the privacy of user nodes. The solution in this paper relies on the selection of global public dataset, so the system should update the public dataset frequently, preferably with multiple

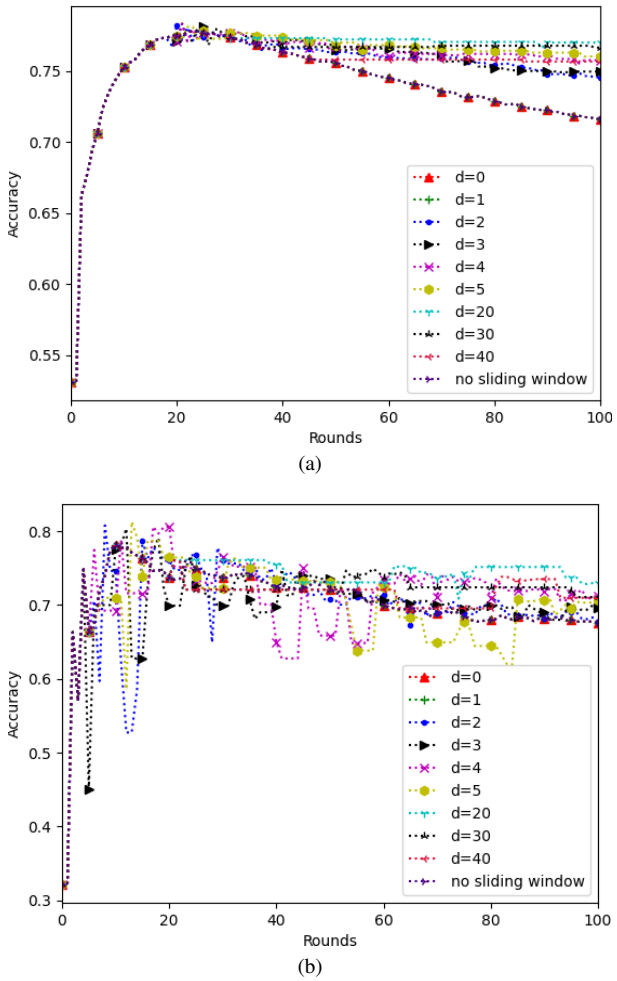


Fig. 6. Model accuracy and loss with different size of sliding window. (a) is the variation of LR model accuracy with the size of sliding window; (b) is the variation of BP model with the size of sliding window.

anomaly so that the global model can better withstand model poisoning attacks. How to choose the test dataset to improve the performance of the global model is beyond the scope of this paper, and it is our future research direction. At the same time, blockchain-based systems inevitably face problems such as high delay and poor scalability. In the future, we will study blockchain sharding and other technologies to improve the efficiency of system operation. In this paper, we only consider the updated anomaly detection global model as the incentive. In the future, we will study how to provide effective economic incentives to encourage smart factories to actively participate in the federated learning training.

REFERENCES

[1] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial Internet of things security: Requirements and fog computing opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2489–2520, 2020.

[2] 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, 2021.

[3] A. Khadidos *et al.*, "A classy multifacet clustering and fused optimization based classification methodologies for SCADA security," *Energies*, vol. 15, no. 10, p. 3624, 2022.

[4] J. Leng *et al.*, "Secure blockchain middleware for decentralized IIoT towards industry 5.0: A review of architecture, enablers, challenges, and directions," *Machines*, vol. 10, no. 10, p. 858, 2022.

[5] X. Wang *et al,*, "Toward accurate anomaly detection in industrial Internet of things using hierarchical federated learning," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7110–7119, 2021.

[6] S. Selvarajan *et al.*, "An artificial intelligence lightweight blockchain security model for security and privacy in IIoT systems," *J. Cloud Comput.*, vol. 12, no. 1, p. 38, 2023.

[7] L. Tseng, X. Yao, S. Otoum, M. Aloqaily, and Y. Jararweh, "Blockchain-based database in an IoT environment: Challenges, opportunities, and analysis," *Cluster Comput.: J. Netw. Software Tools Appl.*, vol. 23, no. 3, pp. 2151–2165, 2020.

[8] J. W. Bos *et al.*, "Unsupervised, federated and privacy-preserving detection of anomalous electricity consumption in real-world scenarios," in *Proc. IEEE iSPEC*, 2022.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017.

[10] D. Li *et al.*, "Blockchain for federated learning toward secure distributed machine learning systems: A systemic survey," *Soft Comput.*, vol. 26, no. 9, pp. 4423–4440, 2022.

[11] W. Liang *et al.*, "Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14741–14751, 2022.

[12] A. Yazdinejad *et al.*, "Block hunter: Federated learning for cyber threat hunting in blockchain-based IIoT networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8356–8366, 2022.

[13] B. Jia *et al.*, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4049–4058, 2022.

[14] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "Fed-IIoT: A robust federated malware detection architecture in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8442–8452, 2021.

[15] V. Mothukuri *et al.*, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, 2022.

[16] D. Wu, Y. Deng, and M. Li, "FL-MGVN: Federated learning for anomaly detection using mixed Gaussian variational self-encoding network," *Inf. Process. Manag.*, vol. 59, no. 2, 2022.

[17] S. Ma *et al.*, "Privacy-preserving anomaly detection in cloud manufacturing via federated transformer," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8977–8987, 2022.

[18] B. Li, S. Ma, R. Deng, K.-K. R. Choo, and J. Yang, "Federated anomaly detection on system logs for the Internet of things: A customizable and communication-efficient approach," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 2, pp. 1705–1716, 2022.

[19] H. Wu and P. Wang, "Node selection toward faster convergence for federated learning on non-IID data," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3099–3111, 2022.

[20] W. Zhang *et al,*, "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, 2021.

[21] Y. Zhan *et al.*, "A survey of incentive mechanism design for federated learning," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 2, pp. 1035–1044, 2022.

[22] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "FLChain: A blockchain for auditable federated learning with trust and incentive," in *Proc. BIGCOM*, 2019.

[23] J. Weng *et al.*, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2438–2455, 2021.

[24] J. Liu, D. Yang, M. Lian, and M. Li, "Research on intrusion detection based on particle swarm optimization in IoT," *IEEE Access*, vol. 9, pp. 38254–38268, 2021.

[25] G. Rathee, C. Kerrache, and M. Ferrag, "A blockchain-based intrusion detection system using Viterbi algorithm and indirect trust for IIoT systems," *J. Sensor Actuator Netw.*, vol. 11, no. 14, p. 71, 2021.

[26] T. Hewa, A. Braeken, M. Liyanage, and M. Ylianttila, "Fog computing and blockchain-based security service architecture for 5G industrial IoT-enabled cloud manufacturing," *IEEE trans. ind. informat.*, vol. 18, no. 10, pp. 7174–7185, 2022.

[27] Y. Zeng, Y. Lin, Y. Yang, and J. Liu, "Differentially private federated temporal difference learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2714–2726, 2021.

[28] J. Tian, X. Lu, R. Zou, B. Zhao, and Y. Li, "A fair resource allocation scheme in federated learning," *J. Comput. Research Development*, vol. 59, no. 6, pp. 1240–1254, 2022.
[29] Y. Tian *et al.*, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1918–1929, 2022.
[30] I. D. Katser and V. O. Kozitsin, "Skoltech anomaly benchmark (SKAB)." *Kaggle*, 2020. [Online]. Available: https://github.com/waico/SKAB
[31] M. Cao *et al.*, "DAG-FL: Direct acyclic graph-based blockchain empowers on-device federated learning," in *Proc. IEEE ICC*, 2021.

Xiaojun Jin received the B.S. degree from Xiangtan University in 2019.He is currently studying for a doctor's degree in Electronic Science and Technology from Beijing University of Posts and Telecommunications (BUPT).His current research interests are in blockchain, privacy computing and federated learning.

Chao Ma received his PhD in Communications Engineering from Aston University, Birmingham, UK in 2014. He joined the China Academy of Information and Communications Technology, as the Director of the Business Development Department of the Institute of Industrial Internet and Internet of Things in 2020. He is also the rapporteur of the Q3 of the ITU-T SG20, the registered expert of ISO/IEC JTC1 SC41 IoT Technical Committee, the expert of W3C Web of Things, Zigbee Alliance expert, Zhijiang Laboratory expert. Research areas include IoT and smart city architecture and standards, blockchain technology and applications, industrial Internet.

Song Luo received his master's degree in Communication and Information System from the Academy of Telecommunication Science and Technology in 2006. He is the Deputy Director of the Institute of Industrial Internet and Internet of Things of the China Academy of Information and Communications Technology, the head of the domestic counterpart group of the ITU-T Internet of Things and Smart City Study Group (SG20), and the associate rapporteur of the ITU-T SG20 Q3. His current research interests include Industrial Internet policies and technologies, basic common technical standards and industrial development of the Internet of Things.

Pengyi Zeng is recently studing for a master's degree in Electronic engineering at Beijing University of Posts and Telecommunications. Her research areas are blockchain, industrial Internet of Things, and federal learning.

Yifei Wei received the B.Sc. and Ph.D. degrees in Electronic Engineering from Beijing University of Posts and Telecommunications (BUPT, China), in 2004 and 2009, respectively. He was a visiting Ph.D. student in Carleton University (Canada) from 2008 to 2009. He was a postdoctoral research fellow in the Dublin City University (Ireland) in 2013. He was the vice dean of school of science in BUPT from 2014 to 2016. He was a visiting scholar in the University of Houston (USA) from 2016 to 2017. He is currently a professor and the vice-dean of school of electronic engineering at BUPT. His current research interests are in intelligent optimization of network resources, deep learning and blockchain technology.