Blockchain-Assisted Cross-silo Graph Federated Learning for Network Intrusion Detection

Hang Shen
Nanjing Tech University
Yanjing Zhou
Nanjing Tech University
Tianjing Wang (✉ wangtianjing@njtech.edu.cn)
Nanjing Tech University
Yu Zhang
Nanjing Tech University
Guangwei Bai
Nanjing Tech University
Xiaodong Miao
Nanjing Tech University

Research Article

Keywords: Network intrusion detection, Cross-silo graph federated learning, Blockchain, Incentive

Posted Date: September 12th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3330608/v1

Additional Declarations: No competing interests reported.

Blockchain-Assisted Cross-silo Graph Federated Learning for Network Intrusion Detection

Hang Shen[1], Yanjing Zhou[1], Tianjing Wang[1*], Yu Zhang[1], Guangwei Bai[1], Xiaodong Miao[2]

[1*]College of Computer and Information Engineering (College of Artificial Intelligence), Nanjing Tech University, 30 South Puzhu Road, Nanjing , 211816, China.
[2]School of Mechanical and Power Engineering, Nanjing Tech University, 30 South Puzhu Road, Nanjing , 211816, China.

*Corresponding author(s). E-mail(s): wangtianjing@njtech.edu.cn;
Contributing authors: hshen@njtech.edu.cn; 202161220055@njtech.edu.cn; 202161120001@njtech.edu.cn; bai@njtech.edu.cn; mxiaodong@njtech.edu.cn;

Abstract

In this paper, a blockchain-assisted cross-silo graph federated learning (B-CGFL) framework is presented for large-scale network intrusion detection, aiming to break down barriers among different organizations and achieve a secure and transparent multi-party collaboration ecosystem. The network scenario is divided into multiple regions. Organizations in each region leverage graph neural networks to analyze local network flow topology information and identify traffic types accurately. With cross-silo graph federated learning, coordinators and organizations collaboratively complete the training and updating of global intrusion detection models. Multiple coordinators jointly maintain a chain to improve the global model's scalability and storage security. Oracle nodes bridge the off-chain data provider and on-chain smart contracts, enabling secure transmission in off-chain model accuracy testing. For fair competition, a reputation-aware model incentive mechanism is designed to improve global model quality. Security analysis confirms that B-CGFL can defend against inference attacks, model plagiarism, and tampering with model test results. Experiments on three challenging datasets ToN-IoT, CSE-CIC-IDS2018, and BoT-IoT demonstrate that compared with benchmark machine learning methods, B-CGFL exhibits superior performance in accuracy and F1-score and facilitates model quality improvement.

Keywords: Network intrusion detection, Cross-silo graph federated learning, Blockchain, Incentive

1 Introduction

With the widespread adoption of network technology and increasing reliance on the Internet, security issues, including DDoS attacks and Web attacks, have become more prominent [1]. These issues lead to significant economic losses and threaten national and social stability. The 2023 State of Security Report[1] released by Splunk points out that 62% of respondents' business-critical applications experience unexpected downtime due to network security incidents at least once a month, a higher percentage than 54% in 2022. Effective identifying and defending against network attack behaviors has become an urgent problem.

As a necessary measure for network defense, intrusion detection aims to monitor potential network intrusion and promptly discover malicious activities [2], which has extensive applications in transportation, Internet of Things security, and industrial control. Many machine learning studies have been applied to network intrusion detection. Traditional methods (e.g., support vector machine, artificial neural network, and genetic algorithms) often require manual feature selection [3], making them challenging to analyze and predict accurately. By automatically extracting the inherent patterns and representation levels of sample data, the deep learning model can capture and identify anomalies and malicious activities from network traffic [4, 5]. Nevertheless, deep learning-based intrusion detection still faces challenges with expanding network scale and the continuous evolution of attack means and types.

1.1 Challenging Issues and Related Works

1) Data Silos refer to storing data in various organizations [6], preventing centralized training and model updates. Each organization relies purely on local data for training, which leads to difficulties in improving model quality. As a distributed machine learning paradigm, federated learning (FL) ensures that each participant has absolute control over his data and only uploads local model parameters to a coordinator as an FL controller, responsible for parameter aggregation to form and distribute the global model to participants. This collaboration paradigm breaks the data silo problem. Generally, FL can be divided into cross-device and cross-silo [7]. Compared with cross-device FL with a scale of hundreds to tens of thousands of local devices, cross-silo FL consists of two to dozens of organizations. They participate in model training in each round. By introducing client filtering and local model weighting, DAFL [8] can mitigate the impact of underperforming local models on the global model during training with reduced communication overhead. Zhao *et al.* [9] proposed an intrusion detection method based on semi-supervised cross-device FL, which utilizes unlabeled open data to improve classifier performance. GöwFed [10] is an industrial-level network threat detection system that incorporates gower dissimilarity matrices and federated

[1] https://www.splunk.com/zh_cn/campaigns/state-of-security.html

averaging. XGBoost [11] is a cross-silo FL approach combining anonymity-based data aggregation and local differential privacy in anomaly detection.

2) Limited Feature Extraction. Network flows have complex topological structures [12], including node connections, interaction patterns, and propagation behaviors. Conventional feature extraction methods focus on individual nodes or local features, making capturing and utilizing topological structure information difficult. By representing network flows as graph structures, graph neural networks (GNNs) can directly operate on and learn the rich structural information, node, and edge attributes in network flow [13], thereby identifying anomalous behaviors and intrusion attacks. Anomal-E [14] adopts a self-supervised approach combining edge features and graph topology to detect network intrusions and anomalies. Xiao *et al.* [15] constructed a control area network graph attention network model that improves anomaly detection accuracy by capturing correlations among different flow byte states. Through network embedding feature representations, Zhang *et al.* [16] proposed a GNN-based intrusion detection framework that can handle high-dimensional redundant but imbalanced and rare labeled data in industrial Internet-of-Things, distinguishing between cyber-attacks and physical failures.

3) Model Security and Credibility Issues. Attackers may attempt to obtain models stored at the coordinator to extract sensitive information [17, 18] or infer privacy of participating parties [19, 20]. Blockchain is a distributed ledger that can solve model security issues in distributed environments. Any changes to the contents require consensus among all participants [21, 22], thus ensuring the integrity and non-comparability of model data. Liu *et al.* [23] proposed a collaborative intrusion detection mechanism based on FL and blockchain. The multi-party aggregation approach reduces the central server's resource utilization, and the blockchain ensures the security of the global model. BFLC [24] is a blockchain-based FL framework without relying on centralized servers to store global models and exchange local model updates. By introducing context-aware transformer networks, FED-IDS [25] can learn spatial-temporal representations of vehicular traffic flows under different attacks. Miners validate distributed local updates from vehicles to prevent unreliable updates from being stored on the chain.

1.2 Contributions and Organization

To address the above challenges, we propose a blockchain-empowered cross-silo graph federated learning (B-CGFL) framework to form a secure, transparent, fair collaborative intrusion detection ecosystem for large-scale networks. The main contributions are three folded:

- **Cross-silo graph federated learning (CGFL) for intrusion detection:** The network flow collected by each organization is constructed into graph structures to train local network intrusion detection models. As the central controller of CGFL, the coordinator aggregates local model parameters uploaded by the organizations in its region to form a cluster-optimized model.

- **Model automation testing:** Trusted execution environment run by an off-chain data provider to test model accuracy. The test result is safely returned to the coordinator after being dual-verified by oracle nodes and smart contracts for reputation calculation.
- **Reputation-aware model incentive:** Coordinators compete for opportunities to get their models on the blockchain. The probability of the coordinator getting the model on-chain is positively correlated with the coordinator's reputation value and model quality.

Last, we present security analysis against inference attacks, model plagiarism, and model result tampering. We also evaluate the detection performance on multiple challenging datasets, demonstrating that compared with typical distributed and centralized machine learning approaches, the proposed scheme achieves significant performance improvement in accuracy and F1-score.

The remainder of this paper is organized as follows: Section 2 provides preliminaries. The system model is introduced in Section 3. Section 4 presents the B-CGFL method. Section 5 provides security analysis. We explain the experimental setup in Section 6, followed by the experimental results in Section 7. Section 8 summarizes the work.

2 Preliminaries

2.1 E-GraphSAGE

In GNNs, the graph structure is defined as a set of nodes, edges, and node attributes, to describe relationships among samples. As a classic GNN algorithm, GraphSAGE [26] considers node feature information propagation and sets whole-graph sampling as node-centric mini-batch neighbor sampling. Figs. 1(a)-(c) explains the training process. E-GraphSAGE [27] can effectively process large-scale graph-structured data by propagating edge feature. By performing GraphSAGE on each edge, the embedding feature of each endpoint pair is concatenated as the edge embedding representation. The training of E-GraphSAGE is summarized into the following three steps as in Figs. 1(d)-(f):

① Random node sampling. From Fig. 1 (d), two endpoints 0 and 1 of edge $e_{0,1}$ are randomly sampled, obtaining the first-order neighbors ($\{2, 4, 5, 8\}$ and $\{5, 6, 7\}$), the second-order neighbors ($\{9, 10, 11, 12, 17\}$ and $\{13, 14, 15\}$) of nodes 0 and 1 respectively. The first and second-order neighbors form a subgraph.

② Edge feature aggregation. From Fig. 1 (e), nodes 0 and 1 subgraphs are executed to aggregate second- and first-order neighbor edge features. For example, the edge features of $e_{2,9}$ and $e_{2,10}$ are aggregated to obtain the aggregated neighbor edge information of node 2, which is concatenated with the node embedding of the previous layer to update node 2's embedding information. Similarly, the embedding of all sampled nodes can be updated.

③ Edge label generation. From Fig. 1 (f), the node embeddings of 0 and 1 are concatenated as the final edge embedding representation of $e_{0,1}$. Edge classification is completed through Softmax to generate the flow label of $e_{0,1}$.

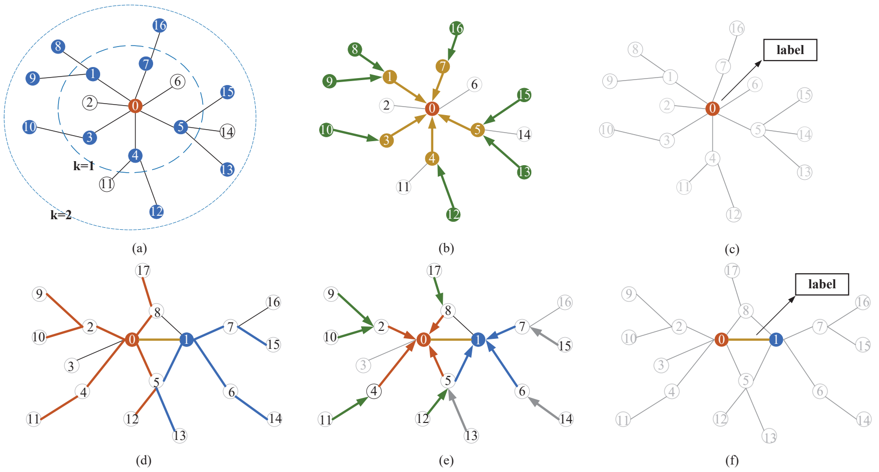


Fig. 1 Examples of GraphSAGE and E-GraphSAGE algorithm training. (a)-(c) are example diagrams of the GraphSAGE algorithm's random node sampling, node feature aggregation, and node label generation. (d)-(f) are example diagrams of the E-GraphSAGE algorithm's random node sampling, edge feature aggregation, and edge label generation.

2.2 Cross-silo FL

Cross-silo FL contains two prominent roles: organization and coordinator. The former only share model parameters instead of original data to protect privacy. Multiple organizations can train in parallel on local datasets. The coordinator periodically aggregates model parameters uploaded from the organizations under its jurisdiction. Cross-silo FL aims to handle non-independent and identically distributed data [28], which may come from different domains, industries, or institutions.

2.3 Oracle

Blockchain provides data query interfaces through oracle contracts. Oracles obtain external data from trusted sources such as the Internet and hardware sensors [29]. After receiving a request, an oracle verifies its authenticity, accuracy, and validity by querying actively or passively received data. Once verified, the oracle writes the data to the chain through transactions. Multiple oracles can form a decentralized network to improve reliability. By solving the trust issue, oracles enable blockchain to obtain verifiable external information in a decentralized manner.

3 System model

3.1 Main Roles

Four entities exist in our scenario as shown in Fig. 2:

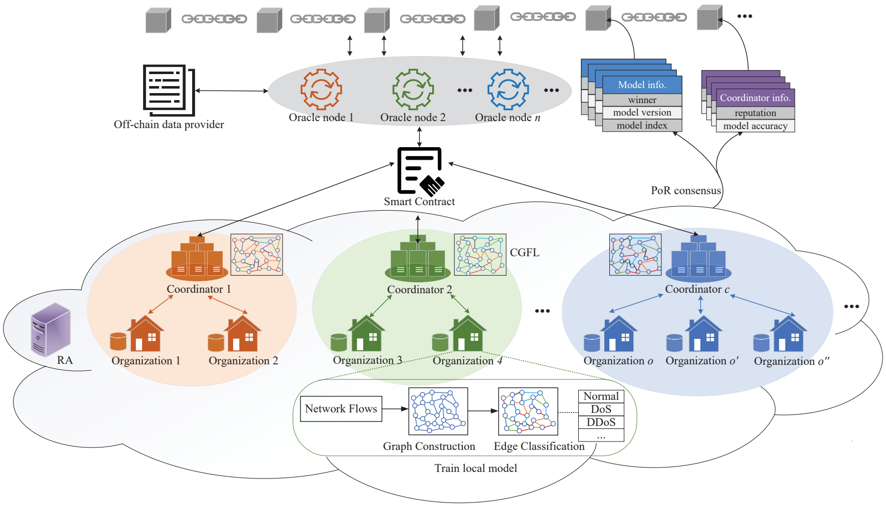


Fig. 2 Blockchain-enabled collaborative intrusion detection framework.

- **Registration Authority (RA)**, as an authorized audit institution, manages the identities of coordinators and organizations joining the platform to ensure data transmission security in network communication.
- **Organizations**, possibly companies, schools, or government agencies, train and update local models. Based on E-GraphSAGE, regional network flows are constructed into a graph structure. By learning the features of edges and topological structures in the graph, organizations transform intrusion detection into an edge classification problem.
- **Coordinators**, manage organizations in their respective regions. Assume the network scenario is divided into multiple regions. For example, coordinator 2 works organizations 3 and 4 in Fig. 2. The coordinator is responsible for distributing secrets and models stored on the chain to all regional organizations and training a global model by aggregating local organizations' models. By incentive, multiple coordinators compete to train a high-quality model.
- **Off-chain data provider**, maintains datasets required for model testing and adopts encryption, access control, and other measures to ensure the security and authenticity of test data, after obtaining complete information on global parameters, off-chain data provider feedbacks test results.

3.2 Collaboration Framework

Regional organizations collaborate under the coordinator's direction to train and update global models. The coordinators jointly maintain a chain to record model information. After preparing an optimal global model, a coordinator sends the hash of the model parameters to smart contracts. After the authorized oracle obtains the complete global parameters, the off-chain data provider calculates the accuracy on a public test dataset. The result is returned to the requesting node through smart contracts. The coordinator broadcasts its global model information and accuracy. A coordinator's current reputation value is determined by model accuracy and historical reputation. A reputation-based consensus mechanism, called proof-of-reputation (PoR), is designed. The coordinator with the higher reputation value wins and obtains the bookkeeping right. The winner's personal information, model version, model index, all coordinators' reputation values, and model accuracies are recorded in the model and coordinator information lists, respectively, for subsequent CGFL model training.

3.3 Design Goals

Our design goals include the following aspects:

- *Scalability:* The proposed framework is not limited by network scale and the number of organizations. The coordinator and organizations in each region can join CGFL after RA certification.
- *Authenticity*: Under the B-CGFL framework, the organizations in each region collaboratively train an intrusion detection model. The off-chain data provider tests the accuracy of the federated aggregation model generated by a coordinator and returns the test result to the coordinator through a dual-verification mechanism to ensure the result's authenticity.
- *Reliability*: Depending on the PoR consensus mechanism, the storage operations on the chain are transparent, and any coordinator can audit the storage.
- *Security*: Each organization's local model parameters are securely uploaded to the coordinator by adding sub-secrets. At the same time, free riders cannot copy the models of other coordinators when broadcasting.

4 Solution

In this section, we present the implementation details of CGFL. Then, an oracle-assisted automation testing and reputation-aware model incentive are tailored for CGFL to build a safe, fair, and transparent collaboration ecosystem.

4.1 Local Model Training

Based on E-GraphSAGE, an organization represents local network flow in graph structures. By learning the edge embedding information in the graph, the organization can capture the relationships and interaction patterns between nodes in the network and extract valuable features for intrusion detection. The local model training process is summarized in Algorithm 4.1.

In the proposed framework, the network traffic dataset of local organization is constructed as a bipartite graph $G(S, D, E)$, with S, D, and E being the sets of source nodes, destination nodes, and edges, respectively. The source IP address and source port number, the destination IP address, and the destination port number of packets from organization o are formed into two binary groups to identify the source node and the destination node. All nodes are defined as an all-1 feature vector. The remaining flow features are represented as feature vectors of edges connecting source and destination nodes.

Algorithm 1 Local Model Training.

```
Input: Local network flow
Output: Weight matrix for training $W$
1: for $i = 1, ..., I$ do
2:     for $k = K, ..., 1$ do
3:         Randomly sampling nodes in the graph;
4:     end for
5:     $h_v^0 \leftarrow (1, ..., 1)^T$;
6:     for $k = 1, ..., K$ do
7:         for nodes sampled at layer $k$ do
8:             $h_{N_v}^k \leftarrow AGG^k(\bigcup_{u \in N_v} e_{u,v}^{k-1})$;
9:             $h_v^k \leftarrow \sigma \left[M^k \cdot (h_v^{k-1} \| h_{N_v}^k)\right]$;
10:        end for
11:    end for
12:    $z_{u,v} \leftarrow \|\{ (h_u^K, h_v^K, \mathrm{e}_{uv})\}$;
13:    $z_{u,v}$ is converted to a category probability;
14:    $L_i \leftarrow -\frac{1}{J} \sum_{j=1}^{J} \log p_j$;
15:    for $k = 1, ..., K$ do
16:        $M^k \leftarrow M^k - \varepsilon \frac{\partial L_i}{\partial M^k}$;
17:    end for
18: end for
19: Output the trained Weight matrix $W$;
```

In the constructed graph structure, the neighborhood of node set V is denoted as $N_v = \{u \in S \cup D | e_{u,v} \in E\}$, representing a set of fixed size and uniform sampling. After sampling the 2-hop 8-neighborhood of N_v, E-GraphSAGE iteratively aggregates the features of adjacent edges layer by layer. Since nodes in the bipartite graph have no feature information, set the initial feature vector of a node $h_v^0 = (1, ..., 1)^T$ (i.e., the initial node embedding), whose dimension is equal to the edge feature vector's dimension. In line 8, the neighboring edge feature vector $h_{u,v}^{k-1}$ of layer k-1 aggregated to node v can be expressed as

$$h_{N_v}^k = AGG^k(\bigcup\nolimits_{u \in N_v} e_{u,v}^{k-1}). \tag{1}$$

In (1), $AGG^k(\cdot)$ chooses a mean aggregation function to compute the edge feature information of the sampled neighborhood of v. In line 9, by concatenating the aggregation vectors, $h_{N_v}^k$, with the embedding vectors, h_v^{k-1}, of nodes in layer k-1 and multiplying them by the trainable weight matrix, M^k, the embedding vector of node v in layer k is updated by the activation function σ as

$$h_v^k = \sigma \left[M^k \cdot (h_v^{k-1} \| h_{N_v}^k)\right] \tag{2}$$

where || denotes the concatenation of information. After K iterations of aggregation are completed, the final edge embedding vector $z_{u,v}$ between nodes u, v in layer K is denoted as the concatenation of two node embedding vectors, i.e.,

$$z_{u,v} = h_u^K \, \| h_v^K \ . \tag{3}$$

In line 12, raw edge feature information may be lost in average aggregation and cannot be well represented in the final edge embedding. For this problem, the two endpoint embeddings of the edge with the original edge features are concatenated as the final edge embedding, expressed as

$$z_{u,v} = \| \ (h_u^K, h_v^K, \mathrm{e}_{uv}). \tag{4}$$

$z_{u,v}$ is converted to a category probability p_j through fully-connected layers and the Softmax layer, i.e., the probability that the jth piece of data is predicted to be the correct label, the loss function for the ith epoch of local training is expressed as

$$L_i = -\frac{1}{J}\sum_{j=1}^{J} \log p_j. \tag{5}$$

In line 16, when the loss function is computed according to the gradient, the weight matrix is updated as

$$M^k = M^k - \varepsilon \frac{\partial L_i}{\partial M^k} \tag{6}$$

to completed one training iteration, where ε is the learning rate. Output the set of weight matrices $W = \left\{M^1, ..., M^K\right\}$ at the end of local training.

4.2 CGFL Approach

Fig. 3 explains the execution process of CGFL. Let C represent the set of coordinators and O_c stand for the set of organizations under the jurisdiction of coordinator $c \in C$. Coordinator c randomly initializes the global model parameters and distributes these to the governing organizations as the global parameter for the first training. In the rth training, organization $o \in O_c$ downloads $W^{(r-1)*}$ from coordinator c stored in the blockchain. Organization o updates $W^{(r-1)*}$ on the local dataset to get the local model parameters, W_o^r. Inference attacks [30] can infer an organization's private data based on its model parameters. For this purpose, organization o combines the upload W_o^r to coordinator c with a distributed sub-secret.

The following describes the implementation details of secret sharing. First, a secret number x^r and Y-1 random numbers $s_1^r, s_2^r, ..., s_{Y-1}^r$ are generated, where $Y = |O_c|$, i.e., the number of organizations under the jurisdiction of coordinator c. Then, the Y-th number is computed as $s_Y^r = x^r - \sum_{y=1}^{Y-1} s_y^r$. Finally, the sub-secrets are ordered as $x_1^r = s_1^r, x_2^r = s_2^r, ..., x_Y^r = s_Y^r$. These Y sub-secrets are distributed to organizations $o \in \{1, ..., O_c\}$. Organization o adds sub-secret x_o^r to the model parameters after updating

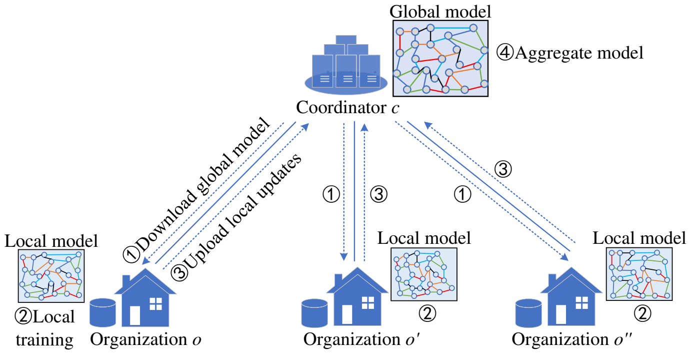


Fig. 3 CGFL framework.

the local parameters, and the updated local model parameters $\tilde{W}_o^r$ are denoted as

$$\tilde{W}_o^r = W_o^r + x_o^r. \tag{7}$$

The encrypted local model is uploaded to coordinator c. Coordinator c aggregates the model parameters of all organizations within its jurisdiction. Since the sub-secrets, $x_1^r, x_2^r, ..., x_Y^r$, are randomly distributed, one or more of them will not reveal any information about the model parameters. The uploaded model can be decrypted when all sub-secrets are combined. By federating all original local model parameters, the coordinator c obtains the global model parameters, expressed by

$$W_c^r = \frac{1}{Y}\left(\sum_{o\in O_c} \tilde{W}_o^r - \sum_{y=1}^{Y} x_y^r\right). \tag{8}$$

After several round iteration epoch, coordinator c obtains optimal global model parameters, denoted as ${W_c}^*$.

4.3 Model Automation Testing

After a coordinator has trained an optimal global model, the model must be tested for accuracy using the public intrusion detection dataset. Fig. 4 shows a framework of on-chain-off-chain interaction in oracle-based model testing, and the main process includes:

① Coordinator c sends a hash of ${W_c}^*$ to smart contracts.

② Smart contracts record the hash of ${W_c}^*$ and calls oracle service contract to request complete parameter information. Oracle receives the request, generates a task, and distributes it to oracle node n.

③ Oracle node n utilizes hardware security module (HSM) to obtain the complete ${W_c}^*$, where HSM ensures the confidentiality of parameters transmission and storage.

④ Oracle node n uses a public key to encrypt ${W_c}^*$ to transmit it to the authorized off-chain data provider.

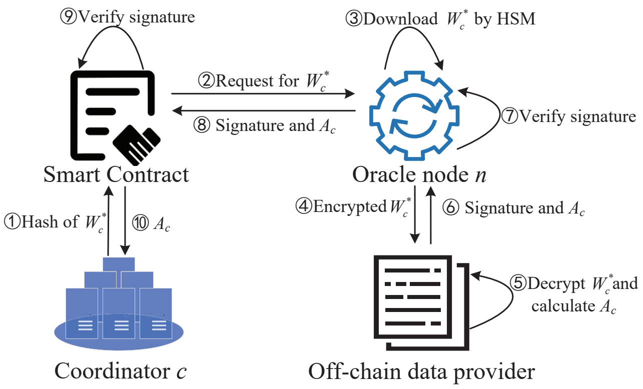


Fig. 4 Model accuracy testing.

⑤ The off-chain data provider decrypts $W_c{}^*$ using the private key in an isolated secure environment. With the public test dataset, the accuracy is computed as A_c.

⑥ The off-chain data provider signs A_c using a digital certificate and sends it to oracle node n.

⑦ Oracle node n verifies the correctness of the signature after receiving the digital signature and A_c.

⑧ Oracle node n submits A_c and its signature to smart contracts as a transaction.

⑨ A predefined function in smart contracts is called to verify signature correctness.

⑩ After verification, smart contracts return the verified A_c to coordinator c. Oracle node n and the off-chain data provider delete the copy of $W_c{}^*$.

4.4 Reputation-aware Model Incentive

Multiple coordinators jointly maintain a blockchain that relies on the PoR consensus mechanism to ensure secure data updating, storage, and model distribution. Coordinators are trusted to assess by combining historical reputation with global model accuracy, which motivates coordinators to compete for on-chain opportunities in global model training. Driven by a reputation-aware incentive mechanism, each coordinator competes for the opportunity to put the model on the chain.

Coordinator c receives A_c and then broadcasts $W_c{}^*$, A_c, pseudo-noise sequence [31], identity, and digital signature. Some coordinators may act as free riders when broadcasting messages, copying model parameters from other coordinators. The pseudo-noise sequence prevents model plagiarism by free riders during message broadcasting.

After receiving $\{W_b^*, A_b\}_{b\in C\setminus\{c\}}$ from other coordinators at time t, coordinator c goes to the chain to query the reputation, $Q_b^{(t-1)}$, of coordinator b that was recorded at time $t-1$. If it is less than threshold ξ, the model of coordinator b will not be accepted even if A_b is high. The accuracy deviation between coordinator c and the other coordinators is computed as

$$\beta_c^{(t)} = \left(1 + \frac{\sum_{b \in C' \setminus \{c\}} (A_c - A_b)}{|C'| - 1}\right) \tag{9}$$

where $C^{'} = \{b \in C | Q_b^{(t-1)} \geq \xi\}$ denotes the set of coordinators whose reputation is larger than ξ at time $t-1$.

The reputation of coordinator c at time $t-1$ is recorded as $Q_c^{(t-1)}$. Based on (9) and $Q_c^{(t-1)}$, the reputation of coordinator c at time t is computed as

$$Q_c^{(t)} = \frac{1}{1 + \exp\left\{-\alpha \cdot \left[Q_c^{(t-1)} \cdot \beta_c^{(t)}\right]\right\}}. \tag{10}$$

with the range of (0,1). Coherence factor α controls the range of variation in reputation over time. To incentivize coordinators to participate honestly and actively in high-accuracy model training, coordinators with higher model accuracy are rewarded with a reputation. Coordinator c is increased if $\beta_c^{(t)}$ is larger than ξ, and decreased otherwise. Accordingly, $Q_c^{(t)}$ is expressed as

$$Q_c^{(t)} = \begin{cases} Q_c^{(t)} - \dfrac{\exp(A_c)}{\sum_{m \in \{1,2,\ldots,|C'|\}} \exp(A_m)}, & 0 < \beta_c^{(t)} < \xi \\ Q_c^{(t)} + \dfrac{\exp(A_c)}{\sum_{m \in \{1,2,\ldots,|C'|\}} \exp(A_m)}, & \xi \leq \beta_c^{(t)}. \end{cases} \tag{11}$$

considering the model's accuracy with the coordinator's historical behavior. A coordinator with a high reputation is expected to win, and its global model parameters are recorded as W^{r*}. After validation by other coordinators, W^{r*} and the reputations of all coordinators are packed into a new block. In the next training, coordinators distribute the on-chain models to organizations, opening a new CGFL and competition for on-chaining.

5 Security Analysis

In this section, we analyze the defense capabilities of the proposed scheme against potential attacks.

- **Defend Inference Attacks:** Coordinators receive local model parameters with sub-secrets. Suppose a coordinator tries to reverse the relationship between the parameters and training data. He has to break the sub-secret to obtain the parameter values, but this process is costly and cannot be realized quickly.
- **Prevent Model Plagiarism:** Coordinators broadcasting models add meaningless pseudo-noise sequences to the parameters with similar statistical properties to the real parameters. A coordinator receiving a noisy model cannot extract the actual

parameters directly. Even if algorithmic denoising is attempted, only models with severely degraded quality are obtained.

- **Prevent Tampering with Model Test Results:** Once the test results are submitted to the smart contract and verified, the test results cannot be modified. The proposed dual validation mechanism introduces a trusted oracle node and smart contracts, both of which will verify the exactness of the signature to prevent tampering with model test results.

6 Experimental Setting

6.1 Dataset Selection

Datasets ToN-IoT[2], CSE-CIC-IDS2018[3], and BoT-IoT[4] validate the proposed method's intrusion detection performance. These three datasets, consisting of different types of benign and attack flows, have been widely used to evaluate the performance of intrusion detection algorithms for the Internet and IoT.

ToN-IoT was generated by ACCS at the Cyber Range Lab on behalf of real large-scale networks. The dataset consists of network flows and logs collected from IoT devices and systems. The dataset contains 796380 (3.56%) benign flows, and 21542641 (96.44%) attack flows, totaling 22339021 flows.

CSE-CIC-IDS2018 was published by the Communications Security Establishment (CSE) in collaboration with the Canadian Institute for Cybersecurity (CIC). Multiple real environment captures are integrated to generate network flows. The dataset contains 13,484,708 (83.07%) benign flows, and 2,748,235 (16.93%) attack flows, totaling 16,232,943 flows.

BoT-IoT was published by the Cyber Range Lab at the Australian Center for Cybersecurity (ACCS) and covers multiple types of botnet attacks in IoT. The dataset contains 477 (0.01%) benign flows and 3668045 (99.99%) attack streams, totaling 3668522 streams.

6.2 Baseline Algorithms and Hyperparameters

To evaluate the performance of the proposed scheme, we selected typical centralized and distributed machine learning methods as baselines:

- **Centralized GNN:** Based on E-GraphSAGE, the data collected by each organization is centrally trained into one GNN model, which can be regarded as the optimal method to achieve a performance upper bound.
- **Distributed GNN:** Each organization relies on their local data and trains a GNN model by E-GraphSAGE.
- **CFL-LSTM:** Each organization's local model is replaced with long short-term memory (LSTM) [32], which detects anomalies in the temporal correlation of traffic and realizes the dynamic detection of unknown network attacks by effectively learning the long-range.

[2] https://ieee-dataport.org/documents/toniot-datasets
[3] https://www.unb.ca/cic/datasets/ids-2018.html
[4] https://ieee-dataport.org/documents/bot-iot-dataset

Table 1 Learning rate setting

Algorithms	ToN-IoT	CSE-CIC-IDS2018	BoT-IoT
CFL-LSTM	0.005	0.01	0.05
CFL-MLP	0.002	0.001	0.01
CGFL	0.01	0.03	0.01

- **CFL-MLP:** Each organization's local model is replaced with multilayer perceptron (MLP) [33], which realizes network intrusion detection by automatically learning complex nonlinear feature mapping that distinguishes normal and malicious traffic based on multidimensional network traffic features.
- **B-CFL-LSTM:** CFL-LSTM is incorporated into the proposed blockchain-enabled framework.
- **B-CFL-MLP:** CFL-MLP is integrated into the proposed blockchain-enabled framework.

Parameters α and ξ were set to 0.1 and 1 to observe the impact of model accuracy on reputation. The learning rate settings for local organizations under each method are listed in Table 1.

6.3 Performance Metrics

Accuracy and F1-score were used to evaluate the test results. Accuracy expresses the number of correctly classified samples out of all samples as a percentage of the total number of samples. F1-score is the reconciled mean of precision and recall. The evaluation criteria are calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

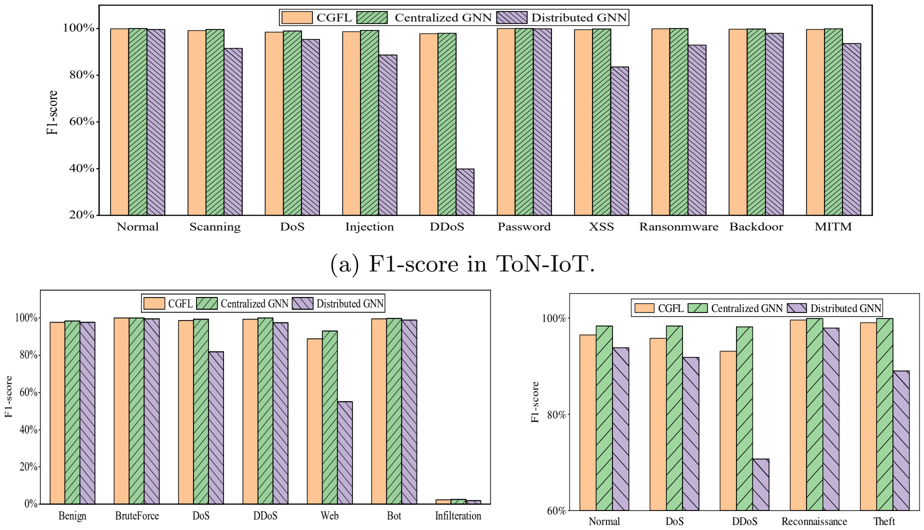
$$\text{F1} - \text{score} = \frac{2 \times TP}{2 \times TP + FP + FN}. \tag{13}$$

TP (True Positive) implies that the model correctly predicts positive case samples as positive cases; TN (True Negative) means that the model correctly predicts negative case samples as negative cases; FP (False Positive) implies that the model incorrectly predicts negative case samples as positive cases; FN (False Negative) implies that the model incorrectly predicts positive case samples as negative cases.

7 Experimental Results

We conducted four sets of experiments on each of the three datasets:

- **Case Study 1**: Comparing centralized GNN, distributed GNN, and CGFL aims to observe and analyze the performance bounds of the proposed CGFL.
- **Case Study 2**: Compares the proposed CGFL with the baseline algorithms CFL-LSTM and CFL-MLP, aiming to observe the impact of different local models on the detection effect, where each coordinator manages ten organizations. In each epoch, each organization's training data was 500, and 2 epochs were trained.



(a) F1-score in ToN-IoT.

(b) F1-score in CSE-CIC-IDS2018.

(c) F1-score in BoT-IoT.

Fig. 5 F1-score of CGFL, Centralized GNN, Distributed GNN.

- **Case Study 3**: Ablation experiments were designed to compare the methods with and without blockchain assistance to observe the enhancement of the proposed blockchain-empowered framework.
- **Case Study 4**: Visualization analysis of the effects of multi-class classifications of proposed solutions.

7.1 Performance Boundary Analysis

This set of experiments compares the F1-score for multi-class classifications. From Fig. 5, the F1-score of centralized GNN and distributed GNN can be seen as upper and lower bounds, respectively. For ToN-IoT and BoT-IoT, the F1-score of centralized GNN intrusion detection is more than 98%. Except for Infilteration, the centralized GNN can achieve F1-score above 93% in CSE-CIC-IDS2018. For DDoS in ToN-IoT and BoT-IoT and Web in CSE-CIC-IDS2018, the F1-score of distributed GNN decreases by 58.11%, 27.46%, and 37.97%, respectively, compared with centralized GNN. The F1-score of the proposed CGFL is close to centralized GNN and significantly higher than distributed GNN. For Web in CSE-CIC-IDS2018, Normal, and DDoS in BoT-IoT, the F1-score of CGFL decreases by 4.21%, 2.47%, and 5.13%, respectively, compared to centralized GNN. For the other types in the three datasets, the F1-score decrease of CGFL compared to centralized GNN is within 1%.

Centralized training can access global data for modeling network topology and achieving more robust detection performance. Specifically, there are three factors. First, the GNN in CGFL is trained on local data, so the sub-models of each organization cannot obtain a sufficient network topology. Second, local data isolation prevents different organizations from getting more diverse data to improve generalization. Third, the training process of federated aggregation of sub-models of each

Table 2 F1-score for benign and individual attack classes.

Dataset	Method	Accuracy
ToN-IoT	CGFL	96.80%
	B-CGFL	98.57%
	CFL-LSTM	92.10%
	B-CFL-LSTM	94.17%
	CFL-MLP	91.81%
	B-CFL-MLP	95.71%
CSE-CIC-IDS2018	CGFL	96.82%
	B-CGFL	98.40%
	CFL-LSTM	92.38%
	B-CFL-LSTM	95.74%
	CFL-MLP	91.08%
	B-CFL-MLP	93.31%
BoT-IoT	CGFL	96.88%
	B-CGFL	98.97%
	CFL-LSTM	91.82%
	B-CFL-LSTM	93.94%
	CFL-MLP	90.32%
	B-CFL-MLP	93.27%

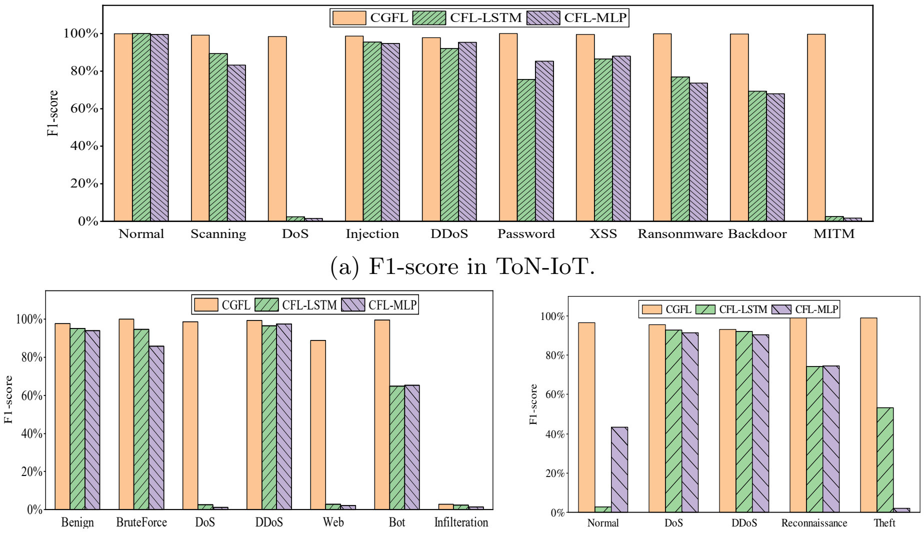
organization increases the difficulty of model expression ability. However, coordinators can only obtain limited model parameters, limiting control and optimization of model training.

7.2 Impact of Local Model

Fig. 6 demonstrates the F1-score of the three cross-silo FL methods in identifying different flow types. For ToN-IoT, the F1-score of the proposed CGFL can reach more than 98.5%, the highest among all forms. In CSE-CIC-IDS2018, none of the plans could identify Infilteration accurately (as explained in Section 7.4 through visualization methods). For DoS and Web, CFL-LSTM and CFL-MLP have low F1-score, while CGFL has an F1-score of 98.56% and 88.76%. For BoT-IoT, CFL-LSTM and CFL-MLP maintain low F1-score when facing Normal and Theft, while CGFL reaches 96.47% and 98.99%. Under the proposed CGFL framework, organizations consider the complex relationship and topology information between nodes in the local network flow and use the connections between nodes for information transfer and learning. When running CFL-LSTM and CFL-MLP, organizations detect network flows centered on a single flow and cannot mine the correlation in multiple flows. Both are suitable for processing sequence or vector data but cannot model complex network structures.

7.3 Ablation Experiment

From Table 2, all three cross-silo FL methods improve accuracy under the proposed blockchain-enabled framework. In particular, compared to CGFL, the proposed B-CGFL enhances the accuracy of the three datasets by 1.77%, 1.58%, and 2.09%,



(a) F1-score in ToN-IoT.

(b) F1-score in CSE-CIC-IDS2018.

(c) F1-score in BoT-IoT.

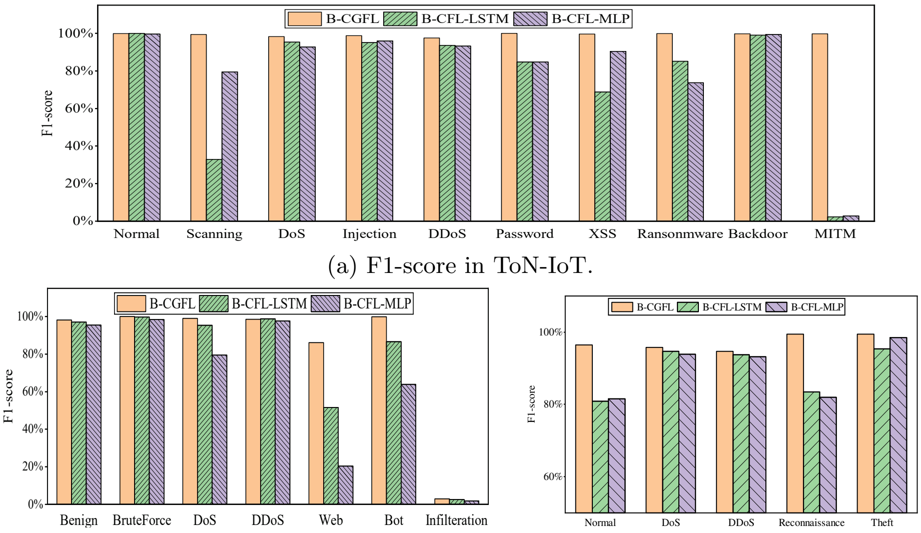
Fig. 6 F1-score of CGFL, CFL-LSTM, and CFL-MLP.

respectively, because coordinators compete to train a higher quality global model based on the reputation-aware model incentive.

As shown in Fig. 7, the F1-score of B-CGFL reaches 99.94%, 98.19%, and 96.50%, respectively, for the three datasets' normal flow. When detecting Benign, BruteForce, and DDoS in CSE-CIC-IDS2018, the F1-score of B-CGFL, B-CFL-LSTM, and B-CFL-MLP exceeds 95.5%. Compared with B-CFL-LSTM and B-CFL-MLP, the proposed B-CGFL reaches 34.52% and 65.65% F1-score enhancement in detecting Web for BoT-IoT. After incorporating our proposed blockchain framework, B-CGFL still outperforms B-CFL-LSTM and B-CFL-MLP. Organizations are modeled on graph structures in the proposed framework through distributed collaborative training to learn the complex relationships between local network flows. The coordinator with the highest reputation can write the global model to the chain and use it for the next global model training.

7.4 Visualization Analysis

t-SNE [34] is a nonlinear dimensionality reduction algorithm that can map high-dimensional data to low-dimensional spaces. Based on this, we visualize the feature representation of the last layer for the proposed B-CGFL. The visualization results generated in ToN-IoT, CSE-CIC-IDS2018, and BoT-IoT are shown in Fig. 8. The types with more classification errors are circled with red boxes for easy observation. Based on the stealthiness of Infiltration, some features are close to normal flow, and Infiltration overlaps with Benign in the red box in Fig. 8(b). Due to multiple types of DoS attacks, some DoS types in the red box in Fig. 8(c) are often misclassified as DDoS. For the rest flow types, the points of the same category are spatially close

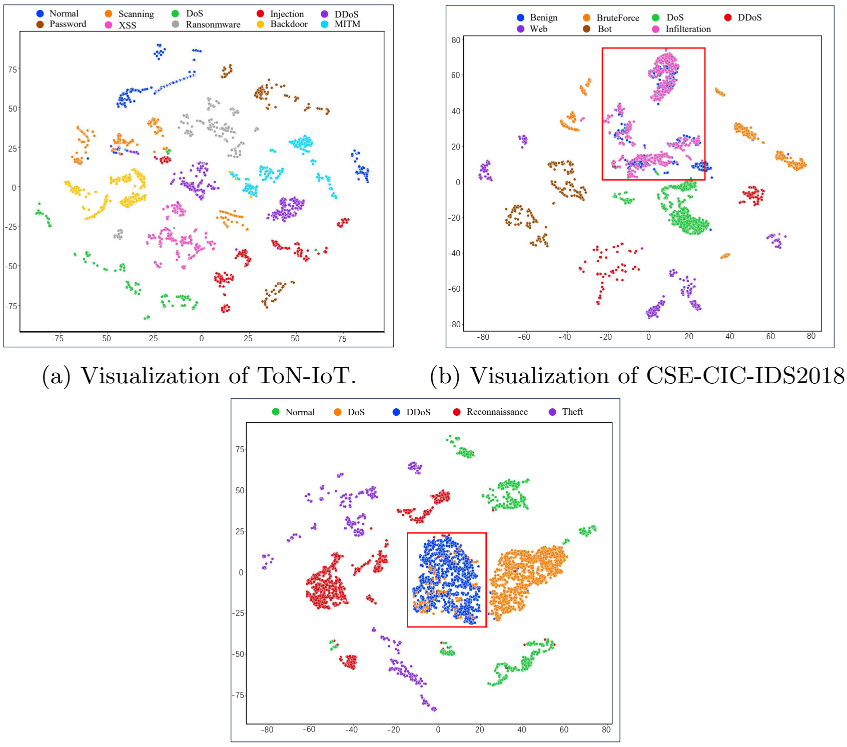


(a) F1-score in ToN-IoT.

(b) F1-score in CSE-CIC-IDS2018.

(c) F1-score in BoT-IoT.

Fig. 7 F1-score of B-CGFL, B-CFL-LSTM, and B-CFL-MLP.



(a) Visualization of ToN-IoT.

(b) Visualization of CSE-CIC-IDS2018.

(c) Visualization of BoT-IoT.

Fig. 8 Visualization analysis of network flow classification for B-CGFL.

to each other, forming a cluster-like structure, and the boundaries of the different categories are relatively clear.

8 Conclusion

By fusing CGFL and blockchain, we have presented a novel intrusion detection approach to realize a scalable, accurate, and trustworthy ecosystem for large-scale network intrusion. Security analysis and experiments on multiple datasets demonstrate the proposed scheme's effectiveness, reliability, and flexibility. The proposed framework can be further developed and improved for constructing an end-edge-cloud collaborative hierarchical federated learning framework for network security. Follow-up work will explore CGFL optimization strategies to enhance detection for specific attack types. Reducing on-chain and off-chain interaction cost also requires further investigation.

Declarations

Ethical Approval and Consent to participate

This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication

All authors agree to publish the paper and related research results of the paper.

Availability of supporting data

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Competing interests

We declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. We declare that there is no financial interest/personal relationship which may be considered as potential competing interests.

Funding

This work was supported by the National Key R&D Program of China under Grant 2021YFB2012301, the National Natural Science Foundation of China under Grants 61502230 and 61501224, the Natural Science Foundation of Jiangsu Province under Grants BK20201357, and the Six Talent Peaks Project in Jiangsu Province under Grant RJFW-020.

Authors' contributions

Hang Shen and Yanjing Zhou wrote the main manuscript text. Tianjing Wang, Yu Zhang, Guangwei Bai and Xiaodong Miao provided guiding ideas and suggestions. All authors reviewed the manuscript.

Acknowledgements

The authors gratefully acknowledge the financial assistance provided by the National Natural Science Foundation of China, Natural Science Foundation of Jiangsu Province, and other research projects.

Authors' information

Hang Shen, Yanjing Zhou, Tianjing Wang, Yu Zhang, and Guangwei Bai are with the College of Computer and Information Engineering (College of Artificial Intelligence), Nanjing Tech University, Nanjing 211816, China.

Xiaodong Miao is with the School of Mechanical and Power Engineering, Nanjing Tech University, Nanjing 211816, China.

Corresponding author: Tianjing Wang

References

[1] Yi, T., Chen, X., Zhu, Y., Ge, W., Han, Z.: Review on the application of deep learning in network attack detection. Journal of Network and Computer Applications **212**, 103580 (2023)

[2] Zhang, C., Costa-Perez, X., Patras, P.: Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms. IEEE/ACM Transactions on Networking **30**(3), 1294–1311 (2022)

[3] Wang, N., Chen, Y., Xiao, Y., Hu, Y., Lou, W., Hou, Y.T.: MANDA: On adversarial example detection for network intrusion detection system. IEEE Transactions on Dependable and Secure Computing **20**(2), 1139–1153 (2022)

[4] Kasongo, S.M.: A deep learning technique for intrusion detection system using a recurrent neural networks based framework. Computer Communications **199**, 113–125 (2023)

[5] Gupta, N., Jindal, V., Bedi, P.: CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. Computers & Security **112**, 102499 (2022)

[6] Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: An experimental study. In: IEEE 38th International Conference on Data Engineering (ICDE), pp. 965–978 (2022)

[7] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., *et al.*: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)

[8] Li, J., Tong, X., Liu, J., Cheng, L.: An efficient federated learning system for network intrusion detection. IEEE Systems Journal **17**(2), 2455–2464 (2023)

[9] Zhao, R., Wang, Y., Xue, Z., Ohtsuki, T., Adebisi, B., Gui, G.: Semi-supervised federated learning based intrusion detection method for Internet of Things. IEEE Internet of Things Journal **10**(10), 8645–8657 (2023)

[10] Belenguer, A., Pascual, J.A., Navaridas, J.: Göwfed: A novel federated network intrusion detection system. Journal of Network and Computer Applications, 103653 (2023)

[11] Yang, M., Liu, S., Xu, J., Tan, G., Li, C., Song, L.: Achieving privacy-preserving cross-silo anomaly detection using federated XGBoost. Journal of the Franklin Institute **360**(9), 6194–6210 (2023)

[12] Lan, J., Lu, J.Z., Wan, G.G., Wang, Y.Y., Huang, C.Y., Zhang, S.B., Huang, Y.Y., Ma, J.N.: E-minBatch GraphSAGE: An industrial internet attack detection model. Security and Communication Networks **2022** (2022)

[13] Haghshenas, S.H., Hasnat, M.A., Naeini, M.: A temporal graph neural network for cyber attack detection and localization in smart grids. In: IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), pp. 1–5 (2023)

[14] Caville, E., Lo, W.W., Layeghy, S., Portmann, M.: Anomal-E: A self-supervised network intrusion detection system based on graph neural networks. Knowledge-Based Systems **258**, 110030 (2022)

[15] Xiao, J., Yang, L., Zhong, F., Chen, H., Li, X.: Robust anomaly-based intrusion detection system for in-vehicle network by graph neural network framework. Applied Intelligence **53**(3), 3183–3206 (2023)

[16] Zhang, Y., Yang, C., Huang, K., Li, Y.: Intrusion detection of industrial Internet-of-Things based on reconstructed graph neural networks. IEEE Transactions on Network Science and Engineering (2022)

[17] Rehman, A., Abbas, S., Khan, M., Ghazal, T.M., Adnan, K.M., Mosavi, A.: A secure healthcare 5.0 system based on blockchain technology entangled with federated learning technique. Computers in Biology and Medicine **150**, 106019 (2022)

[18] Yazdinejad, A., Dehghantanha, A., Parizi, R.M., Hammoudeh, M., Karimipour, H., Srivastava, G.: Block hunter: Federated learning for cyber threat hunting

in blockchain-based IIoT networks. IEEE Transactions on Industrial Informatics **18**(11), 8356–8366 (2022)

[19] Abou El Houda, Z., Hafid, A.S., Khoukhi, L.: MiTFed: A privacy preserving collaborative network attack mitigation framework based on federated learning using SDN and blockchain. IEEE Transactions on Network Science and Engineering **10**(4), 1985–2001 (2023)

[20] Saraswat, D., Verma, A., Bhattacharya, P., Tanwar, S., Sharma, G., Bokoro, P.N., Sharma, R.: Blockchain-based federated learning in UAVs beyond 5G networks: A solution taxonomy and future directions. IEEE Access **10**, 33154–33182 (2022)

[21] Khan, A.A., Khan, M.M., Khan, K.M., Arshad, J., Ahmad, F.: A blockchain-based decentralized machine learning framework for collaborative intrusion detection within UAVs. Computer Networks **196**, 108217 (2021)

[22] Du, Y., Leung, C., Wang, Z., Leung, V.C.: Accelerating blockchain-enabled distributed machine learning by proof of useful work. In: IEEE/ACM 30th International Symposium on Quality of Service (IWQoS), pp. 1–10 (2022)

[23] Liu, H., Zhang, S., Zhang, P., Zhou, X., Shao, X., Pu, G., Zhang, Y.: Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing. IEEE Transactions on Vehicular Technology **70**(6), 6073–6084 (2021)

[24] Li, Y., Chen, C., Liu, N., Huang, H., Zheng, Z., Yan, Q.: A blockchain-based decentralized federated learning framework with committee consensus. IEEE Network **35**(1), 234–241 (2020)

[25] Abdel-Basset, M., Moustafa, N., Hawash, H., Razzak, I., Sallam, K.M., Elkomy, O.M.: Federated intrusion detection in blockchain-based smart transportation systems. IEEE Transactions on Intelligent Transportation Systems **23**(3), 2523–2537 (2021)

[26] Xiao, L., Wu, X., Wang, G.: Social network analysis based on graph SAGE. In: International Symposium on Computational Intelligence and Design (ISCID), vol. 2, pp. 196–199 (2019)

[27] Lo, W.W., Layeghy, S., Sarhan, M., Gallagher, M., Portmann, M.: E-GraphSAGE: A graph neural network based intrusion detection system for IoT. In: IEEE/IFIP Network Operations and Management Symposium (NOMS), pp. 1–9 (2022)

[28] Heikkilä, M.A., Koskela, A., Shimizu, K., Kaski, S., Honkela, A.: Differentially private cross-silo federated learning. arXiv preprint arXiv:2007.05553 (2020)

[29] Beniiche, A.: A study of blockchain oracles. arXiv preprint arXiv:2004.07140 (2020)

[30] Luo, X., Wu, Y., Xiao, X., Ooi, B.C.: Feature inference attack on model predictions in vertical federated learning. In: IEEE 37th International Conference on Data Engineering (ICDE), pp. 181–192 (2021)

[31] Ma, C., Li, J., Shi, L., Ding, M., Wang, T., Han, Z., Poor, H.V.: When federated learning meets blockchain: A new distributed learning paradigm. IEEE Computational Intelligence Magazine **17**(3), 26–33 (2022)

[32] Aydın, H., Orman, Z., Aydın, M.A.: A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment. Computers & Security **118**, 102725 (2022)

[33] Ramchoun, H., Chikh, M.A., Idrissi, J., Ghanou, Y., Ettaouil, M.: Multilayer perceptron: Architecture optimization and training. IJIMAI **4**(1), 26–30 (2016)

[34] Chatzimparmpas, A., Martins, R.M., Kerren, A.: t-viSNE: Interactive assessment and interpretation of t-SNE projections. IEEE Transactions on Visualization and Computer Graphics **26**(8), 2696–2714 (2020)