Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

A trusted feature aggregator federated learning for distributed malicious attack detection

Xinhong Hei, Xinyue Yin, Yichuan Wang, Ju Ren, Lei Zhu*

Xi'an University of Technology School of Computer Science and Engineering, China

ARTICLE INFO

Article history:
Received 30 March 2020
Revised 31 August 2020
Accepted 2 September 2020
Available online 7 September 2020

Keywords:
IoT
Cloud security
Privacy protection
Federated learning
Blockchain

ABSTRACT

With the rapid development of IoT technology, millions of physical devices embedded with electronics or software are put into regular production. Each IoT device is connected to the user's life and property privacy. Without a credible intrusion detection and defense mechanism installed on the device, it may be attacked by hackers, such as monitoring events of home cameras and control of smart devices. These attack events will have a serious impact on users' production and life. This paper proposes a Blockchained-Federated Learning based cloud intrusion detection scheme. The scheme sends the local training parameters of the IoT intrusion alarm set to the cloud computing center for global prediction, and stores the model training process information and behavior on the blockchain. In order to solve the high probability of false alerts affecting the accuracy of the federated learning model, the scheme proposes an alert filter identification module. At the same time, through the erasure code-based blockchain storage solution, the traditional blockchain storage performance is improved to meet the storage needs of a large number of alert training data in real scenarios.

1. Introduction

The Internet of Things(IoT) is the use of information collection equipment such as radio frequency identification, sensors, infrared sensors, laser scanners, etc., according to specific protocols, to connect computable devices to the Internet. Build an IoT environment for edge-side information exchanges and communication to achieve intelligent identification, location, tracking, surveillance, and management in specific scenarios. In recent years, the IoT technology has been widely used in smart city, wearable device (Ren et al., 2017), agriculture, manufacturing industry, etc., which has led to a surge in the number of connected devices, which has reached billions (Soldatos et al., 2015). As the scale of the IoT system continues to expand, it also brings security challenges and a crisis of privacy disclosure. The network security environment cannot be predicted. The IoT devices are subject to special attack methods in different application scenarios. The traditional network security mechanism cannot be fully or adapted to the existing technology. In the absence of a reliable security mechanism, the more IoT devices penetrate into our lives, the more vulnerable users' privacy and even the safety of life and property are. The information interaction in the current IoT environment requires a security mechanism to solve the problems of massiveness, sensitivity, siloing and security of multi-party data computing.

Regarding the complex security issues of IoT, IoT security researchers have proposed different security protection strategies. Ren et al. (2016) propose a channel-aware reputation system with adaptive detection threshold, and find a solution to the danger of wireless sensor networks being vulnerable to selective forwarding attacks. Wang et al. (2020)propose a scheme for syntactic analysis of unknown protocols based on convolutional neural networks. The scheme helps to measure the boundary of a botnet in an IoT environment.

* Corresponding author.
E-mail address: chuan@xaut.edu.cn (Y. Wang).

https://doi.org/10.1016/j.cose.2020.102033
0167-4048/© 2020 Elsevier Ltd. All rights reserved.

Intrusion detection system(IDS) is a commonly used security strategy in current work. Zarpelo et al. (2017) mentioned that the IDS is one of the core components of the IoT system. The IDS is deployed in a heterogeneous IoT environment to enhance the security and robustness of the system. As a security middleware, IDS has been widely deployed in IoT security applications. Meng and Kwok (2011)proposed an adaptive expert knowledge-driven real-time IDS. At the same time, it is a comprehensive detection scheme for specific network functions. Midi et al. (2017) proposed a method to analyze abnormal network traffic by using the deep autoencoders method to extract network behavior, and perform intrusion detection and protection on the damaged IoT devices. Meidan et al. (2018) a real-time hybrid network intrusion detection framework based on MapReduce architecture is used to prevent two common routing attacks—tiankeng attack and selective forwarding attack.

Li et al. (2020) proposed to collaborative IDS for active intrusion detection, combined with multiple IDSs for data analysis, to judge and predict the detection results. But collaborative IDS has the drawbacks of delay and no real-time. The traditional cloud intrusion detection technology is that the IoT device sends the local detection results to the cloud for analysis and judgment, but the traditional solution has some drawbacks. Firstly,The detection result set will expose the vulnerable port of the damaged IoT device. If the detection set is maliciously captured by an attacker, it will cause a greater security crisis. At the same time, the detection data set is also the sensitive data of the device. If the device owner refuses to send it to the cloud, it will lead to data islands and the cloud detection library cannot be updated. Furthermore, there are a lot of false alarms in the detection alarm set, which increases the workload of the cloud server.

This paper proposes a Blockchained Federated Learning cloud intrusion detection system(BFL-CIDS) scheme. Under the premise of protecting the sensitive detection alert information, the distributed machine learning trains the local data set to generate weight parameters, and passes the weight parameters for global learning and prediction. However, IoT devices have weaknesses in storage capacity and computing power (Wang et al., 2019), and cannot perform local training better. At the same time, uploading a large number of weight parameters to the server will also cause broadband congestion, which is also one of the disadvantages of federated learning. For the low storage capacity and computing power of iot devices,we propose to set up regional service party to collect the detection and alert sets of IoT devices in the region. The regional service party performs local training after filtering the false alert information, and stores all the training results on the blockchain. It ensures that the training results are permanently saved and cannot be tampered with. This scheme provides CIDS with functions of data privacy security, preventive maintenance, and permanent storage.

The research contributions of this article are:

- Provide privacy protection for CIDS through Blockchained federated learning;
- Perform false alerts filtering on the detection alert set, reduce the working pressure in the cloud, and improve the quality of the federated learning model.
- Adopt Hyperledger Fabric expansion scheme based on erasure codes. Reduce the storage pressure of the blockchain, improve the storage performance of the blockchain, and meet the storage needs of a large number of alerts training data in real scenarios.

The rest of the paper is organized as follows: Section 2 presents the background as well as the principle of the federated learning and blockchain,and the current research work; Section 3 describes the architecture and workflow of the scheme; The alert filter identification module and blockchain storage module are introduced in Section 4. Section 5 is experimental analysis. Section 6 carries on the comprehensive summary of this scheme.

2. Related work

In this section, we introduce the technical background related to our system, such as federated learning and blockchain. Train alert samples of IoT devices through federated learning, while building trusted execution environments with blockchains for federated learning. At the same time, the research results of Privacy-Preserving Machine Learning (PPML) are discussed in the federated learning section. Finally, the research achievement of the combination of federated learning and blockchain technology are analyzed.

2.1. *Federated learning*

Federated learning (FL) is a training and analysis of local data of the data owner with multiple participants under the premise of protecting sensitive data. FL is a distributed machine learning architecture. FL consists of central server G and multiple local devices $\{C\}_{i=1}^{N}$. In epoch t, there are G participants P_K, D_k represents the data set owned by the k-th participant. The local device trains the global model M_G^t and the local data set D_K through a random descent gradient (SGD) algorithm. Generate local parameter weight w_k^t. Multiple participants send this round of parameter weights to the central server G, G set $\sum_{k=1}^{N} w_k^t$ through the federated average algorithm training to form global parameter weights W_G^t and new model $M_G{}^{(t+1)}$. Let the ideal model for training be M, the loss function $\nabla L(\cdot)$, and the learning rate λ. $m_i(w_k^t) = \lambda \nabla L(w_k^t, D_k)$. $N_K = |P_K|$ indicates the cardinality of P_K.

$$M = \sum_{k=1}^{K} \frac{N_k}{N} M_k(w_k^t) \qquad M_k(w_k^t) = \frac{1}{N_k} \sum_{i \in P_k} m_i(w_k^t) \tag{1}$$

If the data set D_K participating in federated learning is independently identically distribution(IID), you can get expectations $\mathbb{E}_{D_k}[M_k(w_k^t)] = M$. But in more practical applications, the data set is Non-IID (Zhu et al., 2018), which is also one of the problems to be solved by federated learning. If D_K is Non-IID, the global model $w_k(\cdot)$ maintained by P_k is very close to the target model.

As data breaches and privacy infringement events are well known by the public, privacy-oriented machine learning (PPML) has become a security solution that focuses on. Possible adversaries of FL are semi-honest adversaries and malicious adversaries. As for the way to protect data, existing research mainly focuses on secure multi-party computing (Lindell, 2006), homomorphic encryption (Xie et al., 2014), and differential privacy Abadi et al. (2016). (Mohassel and Zhang, 2017) is a PPML two-party (client, server) framework that uses a two-stage training model. Use linear homomorphic encryption or inadvertent transmission to generate the necessary triplets to complete the offline training task. Offline tasks mainly serve multiplication operations in the online phase. For the nonlinear activation function, an alternative solution to support safe multi-party calculations is proposed. Bostani and Sheikhan (2017) combining FL with multi-party secure computing (MPC), a protocol is proposed for the secure aggregation of high-dimensional data. The agreement is applicable to large-scale terminals to complete the input sum through the same server. But the premise is that specific terminal input cannot be exposed. Chai et al. (2019) realized and improved the $SPD\mathbb{Z}_{2^k}$ based on the active safety multi-party computing protocol proposed by Cramer et al. (2018), and realized the inadvertent evaluation algorithm of decision tree and support vector machine (SVM). Chai et al. (2019) through the Paillier homomorphic encryption algorithm, the safe federation matrix factors under the conditions of honest customers and honest but curious servers are decomposed. Geyer et al. (2017) proposed a Differentially Private Federated Learning model to prevent the training model from being subjected to differential attacks. Lu et al. (2020) integrate local difference privacy into FL to protect the privacy of the updated local model. Establish a secure federated learning program through differential privacy.

2.2. *Hyperledger fabric*

Hyperledger Fabric is a permissioned blockchain (Androulaki et al., 2018) that must be identified and approved by the organization before accessing the Fabric network. All members jointly maintain a distributed ledger. All data operations are public and will be permanently recorded and cannot be tampered with. The identity of the node in Fabric is realized by digital certificate. Digital certificates are managed and issued by Fabric CA. Chaincode is an electronic code that is not controlled by a third party and will be automatically executed when the condition is triggered. At the same time, Chaincode can meet different business needs and realize the application of blockchain in different scenarios. Hyperledger Fabric's innovations in modules such as multi-channel, multi-ledger, rights management, and consensus mechanisms provide new ideas for the development of executable services on the blockchain. Fabric's private data can be isolated by unique organizations and channels, providing more data protection methods (Benhamouda et al., 2018). In the Hyperledger Fabric network, all peer nodes may have three identities (can have more than two identities at the same time), namely Endorse Peer, Commit Peer and Orderer Peer. After peer initiates a transaction, the client will first send it to the endorse peer to simulate the transaction, then the transaction enters the transaction pool. The orderer peer sorts block, and distributes to the commit peer and the world state database for confirmation, after meeting the conditions of the consensus mechanism. After that, the block data was officially launched. When a node initiates a transaction, it is necessary to use a private key for digital signature. Hyperledger Fabric uses an Elliptic Curve Digital Signature Algorithm (ECDSA) as digital signature algorithm. ECDSA is one of the application examples of elliptic curve cryptography (ECC). ECC can use a shorter key than the asymmetric encryption algorithm RSA to achieve the same strength.

Elliptic Curve Digital Signature Algorithm (ECDSA): Let the private key and public key be k and K respectively. Where $K = kG$ and G is a point on the elliptic curve E. A random number r is randomly generated before encrypting the plaintext M.

Private Key Signature:

$$\begin{array}{l} \phi = rG(x,y); \quad h = Hash(M); \quad s = \frac{(h+kx)}{r} \\ Commit \quad Peer \leftarrow \{M, (\phi, s)\} \end{array} \tag{2}$$

Peer Public Key Verification Signature:

$$\begin{array}{l} Compute \quad h = Hash(M); \quad \psi = \frac{hG+xK}{s} \\ Verify \quad \phi == \psi \end{array} \tag{3}$$

The key to ECDSA is the introduction of a random number r, which improves the security of the signature. Even if the same message is changed, as long as the random number r is changed, the resulting signature will change accordingly.

2.3. *Research works of blockchained-FL and IoT*

The traditional distributed federated learning system protects the user's privacy by training the user's data locally and uploading the training parameters to the central computing node. However, at present, the federated learning is not perfect, and the user privacy data can be obtained by training parameter backward reasoning, and the process of parameter transmission is not credible, which cannot achieve the real comprehensive protection of user data (Ma et al., 2019). Improvements can be made in the storage, transmission, and traceability of training parameters and models. The BFL-CIDS scheme creates a trusted execution environment for federated learning by combining federated learning with permissioned blockchain. In order to prevent data tampering, the security of training parameters is guaranteed and reliable data support is provided for the correct model of training. At the same time, blockchain provides traceable data tracking for federated learning and encryption and protection for training parameters.

At present, there are also many research works on the combination of federated learning and blockchain. Martinez et al. (2019) proposed to use the EOS blockchain as an incentive layer for federated learning. Through records and incentives, the enthusiasm and high-quality data contributions of federated learning participants are guaranteed. Majeed and Hong (2019) designed the FLchain architecture and formed a blockchain network by edge devices. Through the concept of the blockchain channel, the global model is specifically allocated and the model is stored on the blockchain. Zhang et al. (2020) proposed a secure data sharing architecture based on the blockchain authorization of the Internet of Vehicles to protect the privacy of shared data through federated learning. Improve the utilization rate of system computing resources. Wang (2019) in view of gradient information

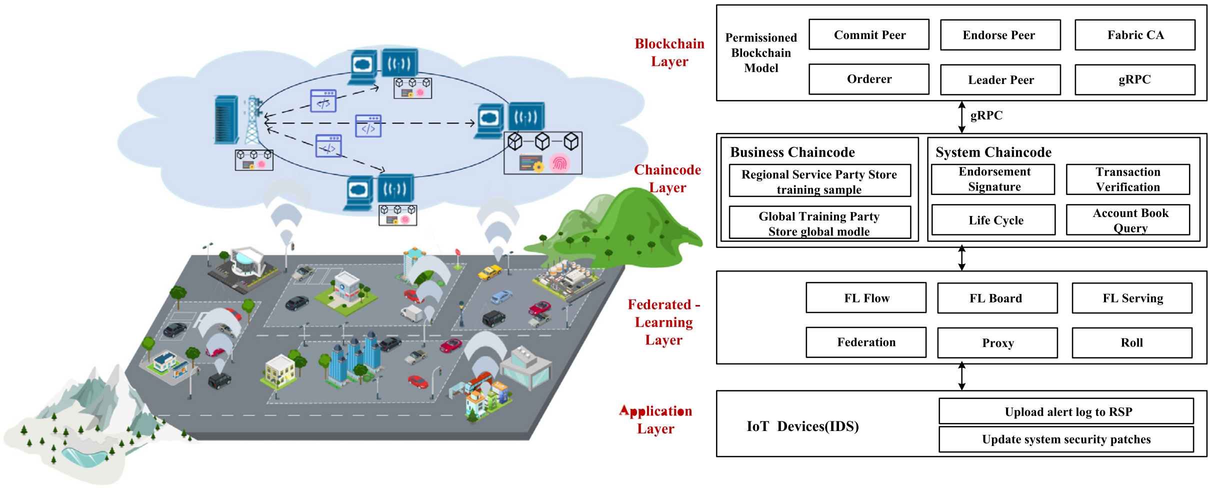


Fig. 1 – BFL-CIDS scheme architecture.

leakage and vulnerability to integrity attacks, a blockchained federated machine learning (BlockFedML) is proposed. At the same time, a security parameter aggregation mechanism, a checkpoint-only contract and an incentive mechanism were developed. Yin et al. (2020) apply blockchain and FL to the IoT environment to implement a secure data collaboration framework. Separate public data from private data, and protect the safe use and transmission of data through blockchain. Sharma et al. (2020) proposed a distributed computing defense framework for a sustainable society, and proposed an algorithm to meet the challenges of limited training data. Ramanan and Nakayama (2019) proposed an FL environment without aggregators, which eliminated the task of centralized aggregators through the blockchain. The efficient combination of blockchain and IoT technology (Tian et al., 2020) proposed blockchain-based secure key management scheme to protect the security of data collected by dynamic wireless sensor networks(DWSNs) in the Industrial Internet of Things(IIoT), and improve the performance of DWSNs in IIoT Credibility.

3. Scheme architecture and structure

3.1. *System architecture*

The architecture of BFL-CIDS scheme is shown in Fig. 1. There are four components including: Application layer, Federated-Learning layer, Chaincode layer,and Blockchain layer. The following is a detailed description of each layer.

Application Layer is a collection of edge IoT devices. Each device is equipped with IDS to detect and monitor network attacks through IDS. Collect and send abnormal alarms to the federated learning layer. At the same time receive the system security patch sent by the federated learning layer. For the latest abnormal network detection library, to avoid the same false positive alert multiple times.

Federated Learning Layer. The FL framework can be divided into offline training and online training, so as to ensure the high availability of the system and consider the system disaster tolerance The FL Serving module is an online inference to local model. FL Flow is a key module for executing task scheduling, in which the client's main task is the user submission training task, the server is used to handle the user's submission request scheduling, and the server is the entrance of the FL cluster. Federation is a module for learning data exchange between parties. FL Board is a visualization module that can timely display the status, amount completed, training results and index values of tasks. The Roll module provides distributed computing and storage support. Proxy module realizes model forwarding between parties and is the only network exit of parties, which can increase security and facilitate network communication management. The parameters and models after the FL training are stored in the blockchain through the chaincode.

Chaincode Layer is a digital protocol that can be used to implement different blockchain application scenarios. Chaincode is not controlled by third-party members, and execution can be triggered when the protocol conditions are met. Chaincode is divided into business chaincode and system chaincode. The former is responsible for implementing specific business rules, and the latter sets the system specifications for Peer endorsement and transactions, block queries, and chaincode lifecycle. This solution designs two types of business chaincode, one is to upload weight parameter settings for regional service party, and the other is to upload global training model settings for global training party. Through different business chaincode to achieve different role operation rules.

Blockchain Layer is based on open source Hyperledger Fabric as the underlying blockchain technology. Chaincode layer interacts with the blockchain layer through gRPC. gRPC is a modern open source high performance remote procedure call (RPC) framework that can run in any environment. In this solution, the blockchain provides storage of training parameters, evidence chain, and traceability services. By permitting the advantages of blockchain identity management, the operating environment of federated learning is purified and the quality of the training model is optimized.

3.2. *Scheme party description*

Learning − unabled Node(LN) : LN is an IoT edge device with weak learning ability and storage capacity. LN joins the BFL-CIDS scheme to monitor the abnormal behavior of the device network. The caught exception behavior is sent to the RSP in the form of a warning log. After receiving the result of RSP feedback alert information, LN will add false positive alert to the white list to avoid false positive alert again, and timely update the security patch of the defense system to enhance the defense function of the system. The common abnormal alerts are: abnormal process behavior, sensitive file tampering, abnormal network connection, abnormal flow alert, application intrusion, etc.

RegionalServiceParty(RSP) : RSP is the Service node that connects LN to GTP and is also a member node on the permissioned blockchain. RSP has certain computing and storage capacity, and can perform preliminary filtering and training for the submitted alerts information. The perliminary filtering work includes labeled false poitive alters and alerts classification. After local training, RSP send the common alerts and abnormal alerts of LN in the region to GTP for centralized training in the form of model parameters. RSP will also upload the model parameters to the permissioned blockchain to prevent the parameters from being tampered in the transmission process and ensure the reliability of parameters. The data on the permissioned blockchain is maintained by all the members together, and no node can change the ledger information, providing a secure and permanent storage for parameters.

GlobalTrainingParty(GTP) : GTP is the centralized Training node of the BFL-CIDS scheme and a member node in the permissioned blockchain. GTP has strong computing and storage capacity, and aggerates all RSP submission parameters for global training. Based on a large number of attacks and defense samples, after machine learning of alerts, the alerts attack behavior is simulated to form a new training model. The model is divided into three types: the trend of alert attack, identifying new attack algorithm and non-threat alerts. GTP also uploads the model to the permissioned blockchain. The RSP, as the node of the permissioned blockchain, will indirectly receive the global training results at the first time and feedback the final alarm status to LN. GTP can obtain the previously uploaded training model from the blockchain, and timely train the new model according to the latest submitted warning sample, in order to cope with the constant evolution of network attack means and virus samples. permissioned blockchain provides a safe and permanent version traceability model for the training model, and improves the application scenario and credibility guarantee of federated learning.

3.3. *Scheme workflow and network model*

Fig. 2 shows the workflow of the BFL-CIDS scheme. In round n, it is mainly divided into Collection module, Training module and Reporting module. In the Collection module, RSP is mainly responsible for the collection and filtering services of alerts. LN sends the alarm information to RSP, and RSP responds to LN and updates the latest security patch. When the network connection fails or the device is damaged, the RSP request will be rejected. The RSP filters the collected information and stores it in a local database. In the Training module, RSP performs local training according to the global model M_G^t, and sends the weight parameter W_k^t to GTP. In the Reporting module, GTP aggregate participates in the training of RSP weight parameters. GTP uses the FedAvg algorithm for global model training and obtains the new model M_G^{t+1}. Send M_G^{t+1} to the RSP node. In Training and Reporting module, RSP and GTP operate in the blockchain network. All training parameters and models will be permanently saved to the blockchain. Each round of models and parameters can realize the functions of permanent storage, non-tampering and real-time traceability.

Fig. 3 is the network topology of the BFL-CIDS scheme. RSP upon receive alerts submitted from LN by a region,RSP initially filters and classifies the alerts to store the available alerts in local database. After RSP conducts local training on the sample, the training parameters are formed and stored in the blockchain. GTP traces the latest training parameters of each RSP from the blockchain for global model learning. Based on a large number of alert samples, the attack is analyzed and predicted, the attack behavior or trend is inferred, and a new model is trained to be uploaded to the blockchain. RSP and GTP, as the nodes of premissioned blockchain, jointly maintain the training parameters and models.

4. Detailed description of scheme

4.1. *Alert filter identification module*

4.1.1. *Alert analysis*

The current memory capacity and computing capacity of edge nodes of the IoT are not enough to meet the needs of modern mass data. Each edge device has no memory for the tens of thousands of alerts it generates each day. The amount of alarm information for each node in the RSP collection area is even larger. Worse, false alerts account for 90% of the unfiltered alerts, and such a high false alert rate can reduce the accuracy of federated learning models. Snort is an open source intrusion detection system.We replay the data set DARPA1999 on the Snort system to test for false positive alerts. There were no attacks on the first and third weeks of DARPA1999, but Snort showed more than 6,000 alerts each week, which were false alerts. The sheer volume of false postive alerts is a huge challenge for federated learning.Therefore, RSP added the alert filter identification (AFI) module to preliminarily determine and identify alerts. Add the available alerts to the training database for machine learning. AFI modules can enhance the performance of federated learning, reduce redundant training and ensure the quality of training models.

The BFL-CIDS solution replayed a low-rate DDoS attack data set (The DDOS attack occurred at 16:42-17:00 on December 10, 2018) on the iot device to simulate the alarm handling of the iot device in the face of intrusion process. The Iot device sends the abnormal behavior flow information process to the RSP. After RSP sorts out the collected alarm traffic information, the preliminary alarm information is obtained in table 1. Perform the Apriori algorithm on the alarm information to complete the frequent item set mining, and obtain the frequent item-level rules of the alarm traffic information. Based on the prior knowledge (determined warning behavior) pro-

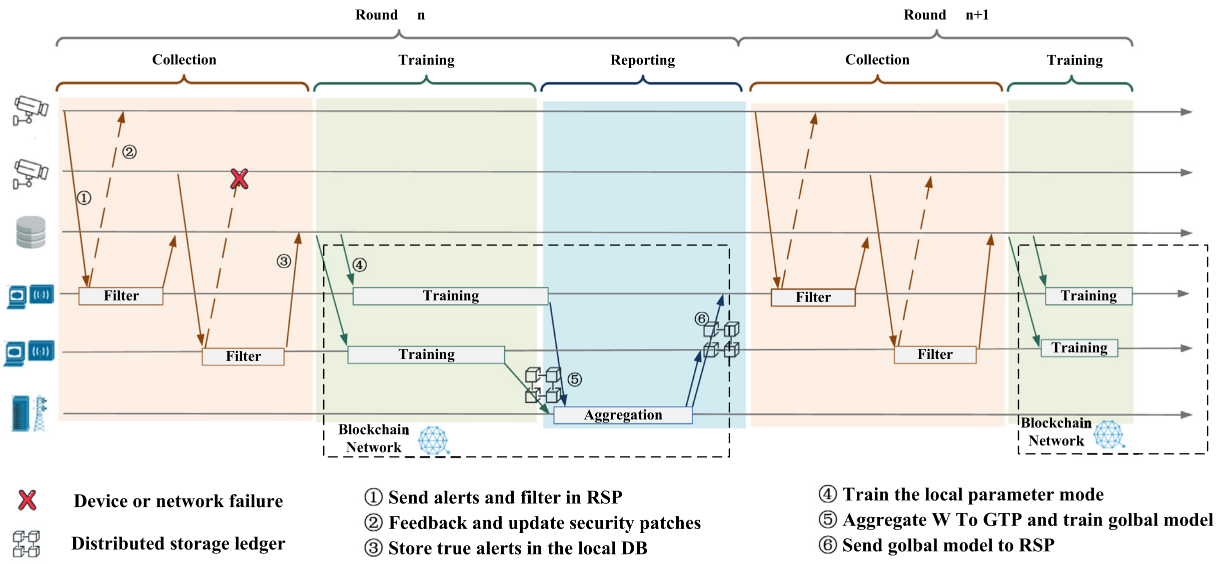

Round n
Collection
Training
Reporting
Round n+1
Collection
Training
①
②
③
④
⑤
⑥
Filter
Filter
Training
Training
Aggregation
Blockchain Network
Filter
Filter
Training
Training
Blockchain Network
Device or network failure
Distributed storage ledger
① Send alerts and filter in RSP
② Feedback and update security patches
③ Store true alerts in the local DB
④ Train the local parameter mode
⑤ Aggregate W To GTP and train golbal model
⑥ Send golbal model to RSP

Fig. 2 – The workflow of BFL-CIDS scheme.

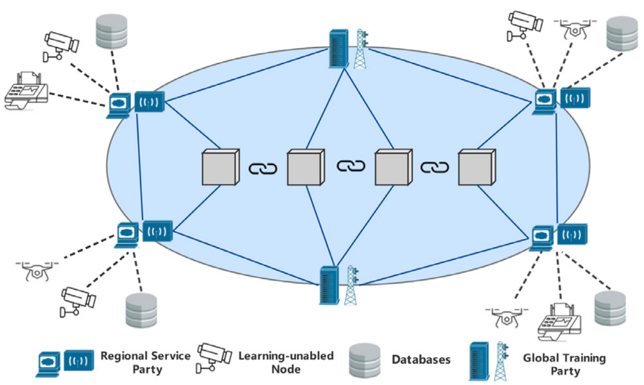


Fig. 3 – The network topology of BFL-CIDS scheme.

Table 1 – Analyze alarm traffic information.

Src_{ip}	Dst_{ip}	packats	Duration
10.42.0.215	52.40.109.206	30	16:43:42.4 -16:44:54.5
10.42.0.29	10.226.11.155	4354	16:48:19.3 -16:49:37.8
10.42.0.106 - 110.42.0.107(etc.)	10.42.0.1	2690	16:42:18.5 -16:59:59.6
10.42.0.1 -10.42.0.107(etc.)	10.42.0.107	431	16:42:19.7 -16:59:35.4
10.42.0.1, 10.42.0.36	10.42.0.106	3855	16:42:18.5 -16:59:59.6

vided by GTP, RSP first marks some rules. RSP advances the relevant features of the labeled and unlabeled alarms, and then executes local training (a semi-supervised learning algorithm based on divergence). After multiple iterations of training, determine whether the attribute of the alarm rule is a false alarm or a true alarm. RSP will also timely update the alarm rules of the local rule base from the model parameters fed back by GTP to obtain more accurate alarm judgments.

The solution can design different rules according to the fateures of the traffic in formula (4). According to the abnormal points of the intrusion information feedback, the relevant abnormal fateures are extracted. For example, if the packet flow is abnormal, the fateures of packet_threshold are extracted.

$$\begin{aligned}&[service : Protocoltype \,\&\, Dst_{ip} \,\&\, TimeStamps\\&\&\, Duration \,\&\, Features]\{support \,\&\, confidence\}\end{aligned} \tag{4}$$

Design frequent item rule format for alarm information in Table 1. Such as formula (5):

$$\begin{aligned}&[service : http \,\&\, Dst_{ip} = 10.42.0.1\\&\&\, TimeStamps = 2018-12-10;\ 16:42:18.5\\&\&\, duration = 1061.1s \,\&\, packet\_threshold = 4354]\\&\{support = 5\% \,\&\, confidence = 95\%\}\end{aligned} \tag{5}$$

The rule shows: When using http service to connect, for the rule item whose destination ip(10.42.0.1) starts 16:42:18.5 on December 10, 2018.The duration is 1061.1s and the packet-threshold is 4354. The support and reliability of this rule is respectively 5% and 95%.

4.1.2. *Detailed description of AFI module*

At present, the most common false alarms are characterized by periodicity and high density. Extract the features of an alert as a set A:

$$\begin{aligned}A = \{&timestamp, alert\_Type, protocal, src\_IP,\\&dst\_IP, dst\_port\}\end{aligned} \tag{6}$$

According to Viinikka et al. (2006), false alerts are periodic and intensive. We define the two characteristics of a false alert as follows.

Definition 1. Alerts Periodicity Feature. According to the statistics of alerts over a period of time at a specific time, the

Table 2 – Similar alert analysis.

Similar characteristics	Description
$A_i.alert\_Type{=}{=}A_j.alert\_Type$ & $A_i.src\_IP{=}{=}A_j.src\_IP$ & $A_i.dst_{IP}{=}{=}A_j.dst\_IP$	The same security incident
$A_i.src\_IP{=}{=}A_j.src\_IP$ & $A_i.dst_{IP}{=}{=}A_j.dst\_IP$	Same attackers against the same target IP
$A_i.src\_IP{=}{=}A_j.src\_IP$	Multiple attackers against the same target IP
$A_i.dst\_IP{=}{=}A_j.dst\_IP$	Same attackers against multiple target IPs

random process $\mu(t)$: $t \in N$ is used to represent the generated time series, and $\mu(t)$ is the number of alarms generated in time T. Periodic analysis is carried out on the time series. The time domain signal is transformed into frequency domain signal by Discrete Fourier Transform*DFT*, and the accurate periodic value T is calculated. The definition of *DFT* is:

$$DFT(f) = \sum_{n=0}^{N-1} \mu(t) e^{\frac{t2\pi jf}{N}} \tag{7}$$

$\frac{2\pi j}{N} \in \mathbb{C}$, $\mathbb{C}$ is the set of constant terms. Find the maximum value of f, f_{max}, and compute interval value $T = \frac{1}{f_{max}}$.

Definition 2. If Alert A and $\bar{A}$ can satisfy one of the conditions in Table 2, they are similar alerts.

Definition 3. Alerts Density Feature.Within the time interval T, alert density D_i is the ratio of the number of similar alarms to the difference of their time distribution.

$$D_i = \begin{cases} \frac{Number(A,\bar{A})}{t_{imax} - t_{imin}}. & |t_{imax} - t_{imin}| < T \\ 0, & Otherwise. \end{cases}$$

The AFI module uses frequent itemset mining algorithm (Hei et al., 2019) and disagreement-based semi-supervised Learning (Tri-training algorithm) and the functions that the algorithm needs to achieve are as follows. Frequent itemset mining algorithm discovers redundant alert items in the alert log after the format is standardized. Judging the periodic item set and the alarm density according to the redundant alerts set, the false alarm is initially marked. The Tri-training algorithm trains the marked and unmarked alerts in the local database through three classifiers and filters false alerts.

Frequent itemset mining algorithm(Apriori algorithm):For $I = \{i_1, i_2, ..., i_d\}$ is the set of all items in the data, while $T = \{t_1, t_2, ..., t_d\}$ is the set of all transactions. A collection of 0 or more items is called an itemset. If an item set contains k items, it is called an k-item set. Obviously, each transaction n contains a subset of I.

The association rule is the implication expression of $X \leftarrow Y$, where X and Y are disjoint item sets, namely $X \cap Y = \varnothing$. The strength of an association rule can be measured by its *Support* and *Confidence*. The support determination rule can be used for the frequency of a given data set, while the confidence determines the frequency of Y in transactions involving X. The forms of *Support* s and *Confidence* c. are defined as follows:

$$s(X \leftarrow Y) = \frac{\sigma(X \cup Y)}{N}$$
$$c(X \leftarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{8}$$

In Algorithm 1, C_k represents k candidate item set,F_k stands for frequent k item set. N is the number of term sets. T is the number of transactions.*subset*() is defined as a function that obtains subsets of candidates. *apriori_gen*() is the algorithm for generating frequent k-item sets. *minsup* is minimum support threshold. In steps 1-2, the algorithm first traverses the data set, detects the support degree of each item, and obtains frequent 1-item sets. Next in step 5, loop through the frequent $(k - 1)$ -itemset to derive k-candidate itemset. Calculate the candidate item set support by traversing the data set in steps 6-10. After the support is calculated in step 12, the infrequent item set is eliminated. At step 13, the algorithm ends when no new frequent item sets are generated.

Algorithm 1 Frequent itemset generation of the Apriori algorithm

```
1: $k$=1.
2: $F_k = \{i \mid i \in I \cap \sigma(i) \geq N * minsup\}$.
3: repeat
4:     $k = k + 1$.
5:     $C_k = apriori_{gen}(F_(k - 1))$.
6:     for each transaction $t \in T$ do
7:         $C_t = subset(C_k, t)$;
8:         for each candidate itemset $c \in C_t$ do
9:             $\sigma(c) = \sigma(c) + 1$;
10:        end for
11:    end for
12:    $F_k = \{c \mid c \in C_k \cap \sigma(c) \geq N * minsup\}$.
13: until $F_k = \varnothing$.
14: Result = $\cup F_k$.
```

If we solve all the frequent item sets directly, the time complexity will be very high. Apriori algorithm can reduce the time complexity of frequent item sets. The time complexity of Apriori algorithm is related to minimum support threshold and transaction number. Because frequent itemset generation of the Apriori algorithm is nested with this *apriori_gen* method, the time complexity of Algorithm 1 is $O(n^2)$.

Disagreement-based Semi-supervised Learning was proposed by Blum and Mitchell (1998). They assume that the dataset has two fully redundant views, each view has a strong learner, and two views Mutually independent under conditions. Zhou and Li (2005) proposed a single-view, tri-training algorithm (Algorithm 2). A significant feature of the algorithm is that it uses three classifiers, which can not only easily handle the problem of label confidence estimation and prediction of unseen examples. The flow of the algorithm is mainly as follows:

Algorithm 2 Tri-training algorithm

Input: L:Labeled training set; U:Unlabeled training set; LA:Learning Algorithm

```
for i ∈ 1,2,3 do
    $S_i \leftarrow BootstrapSample(L)$
    $h_i \leftarrow LA(S_i)$
    $e_i' \leftarrow 0.5; l_i' \leftarrow 0$
end for
repeat until None of $h_i(i = 1, 2, 3)$ changes
for i ∈ 1,2,3 do
    $h_i \leftarrow \emptyset; Update_i \leftarrow false$
    $e_i \leftarrow MeasureError(h_m, h_s)(m, s \neq i)$.
    if $(e_i < e_i')$ then
        for Each $x \in U$ do
            if $h_m(x) = h_s(x)(m, s \neq i)$ then
                $L_i \leftarrow L_iU(x, h_m(x))$
            end if
        end for
    end if
end for
if $(l_i' = 0)$ then
    $l_i' \leftarrow \frac{e_i}{e_i'-e_i} + 1$
end if
if $(l_i' < |L_i|)$ then
    if $(e_i|L_i| < e_i'l_i'')$ then
        $Update_i \leftarrow true$
    else if $l_i' > \frac{e_i}{e_i'-e_i}$ then
        $L_i \leftarrow Subsample(L_i, \frac{e_i'l_i'}{e_i} - 1), Update_i \leftarrow true$
    end if
end if
for i ∈ 1,2,3 do
    if $Update_i = true$ then
        $h_i \leftarrow LA(LUL_i); e_i' \leftarrow e_i; l_i' \leftarrow |L_i|$
    end if
end for
Output: $f(x) \leftarrow argmax_{y \in label} \sum_{h_i(x)=y} 1$
```

1) Repeatable sampling (bootstrap sampling) of the labeled example set to obtain three labeled training sets, and then generate a classifier from each training set.
2) In the collaborative training process, the new labeled examples obtained by each classifier are provided by the other two classifiers. Specifically, if the two classifiers predict the same unlabeled example, the example will be It is considered to have higher label confidence and is added to the labeled training set of the third classifier after labeling.
3) Predict unlabeled instances through the integrated voting of three classifiers.

Based on the above characteristics of error warning, we propose an RSP alert filter identification(RSP-AFI) algorithm,which based on frequent item mining algorithm and semi-supervised learning algorithm. Fig. 4 shows the alerts filter Identification model. The scheme mainly consists of five components:alerts standardization,frequent feature itemset mining,alerts storage, disagreement-based semi-supervised learning and filtration. The first component aims to standardize the alert information.the features are extracted from the alarm sample set and converted into a common format. Frequent feature itemset mining component is applied to the data items to obtain similar alerts information and conduct the first information filtering and analysis. The third component is to save the data items to the database for backup and read the labeled and unlabeled data items from the database. Disagreement-based semi-supervised learning component is through three different classifiers for the marked and the unmarked items of data through the tri-training algorithm are analyzed. Filtration component on the step analysis to filter results and true alerts output.

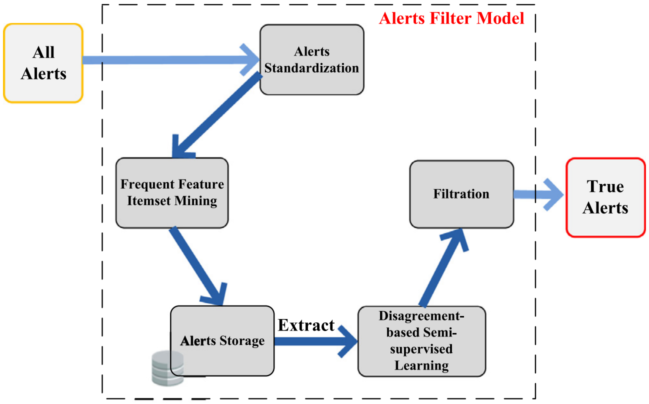


Fig. 4 – Alerts filter and identification model.

4.2. *Blockchain storage based on erasure code*

The current blockchain storage mechanism uses multi-copy redundancy, and the nodes participating in the consensus in the blockchain network need to store an identical copy.When the blockchain needs to store too much data, it increases the burden of nodes. Lu et al. (2019) also mentioned that the blockchain technology will enable federated learning, and the system will be applied to the Internet of vehicles environment. Faced with a large amount of system log information, Lu chooses to upload node behavior to the blockchain. This paper proposes a blockchain storage model based on erasure code and discusses the advantages of this method over redundant copy storage in storage occupancy and system availability.

Erasure correction code was first used in the communication industry, and then it was widely developed in distributed storage. Reed - Solomon (RS) is by far the most popular remedy delete code, is a kind of support arbitrary code number *n*, and check the number *m* every block of Maximun short separable (MDS) coding. RS code performs polynomial calculations for coding segments and check segments in Galois field GF (2^m). Thai et al. (2015) indicates that when *m*:=8, GF has the best computational efficiency.

In the Hyperledger Fabric, the blockfile *BF* of *j*-th block is segmented into *m* blockfile $\left\{F_0^j, F_1^j, \ldots, F_{m-1}^j\right\}$. Encode *m* blockfiles under finite field GF to generate check blocks $r = \{\xi_0, \xi_1, \ldots, \xi_r\}$.Specifies the maximum memory S_m for the blockfile.Before sharding the blockfile, fill the source blockfile with 0 to reach the maximum capacity S_m.This causes each blockfile to be split evenly $\frac{S_m}{m}$ bytes. The distributed storage method is adopted to store m blockfiles and r check blocks on different blockchain nodes respectively, so as to ensure that all data can be restored when some nodes are offline or used in the semi-honest user model environment. Adding RS code to blockchain storage technology can reduce the storage complexity from O(n) to O(1). The GF(2^m) domain is a scalar multiplication. A jth blockfile encoding is defined in the Eq. (9), and set the invertible matrix vector be α.

$$\xi_r{}^j = \alpha^j_{r\cdot 0}F^j_0 + \ldots + \alpha^j_{(r-1)\cdot(m-1)}F^j_{m-1} + \alpha^j_{r\cdot m}F^j_m \tag{9}$$

Assume that each organization on Hyperledger Fabric has only one node. There are n nodes on the blockchain, sum_B blocks. Through RS code, BF is divided into m blockfiles and r check blocks ($n \geq (m+r), m > r$). Storage optimization analysis of Fabric storage capacity expansion scheme. Blockchain storage capacity S_{EC} based on erasure code is:

$$S_{EC} = \frac{S_m}{m} \cdot num_B \cdot \frac{m+r}{n} \tag{10}$$

Traditional blockchain storage capacity S is:

$$S = num_B \cdot S_m \cdot n \tag{11}$$

The storage ratio ρ of S_{EC} and S is:

$$\rho = \frac{S_{EC}}{S} = \frac{\frac{S_m \cdot num_B \cdot (m+r)}{n \cdot m}}{num_B \cdot S_m \cdot n} = \frac{m+r}{m \cdot n^2} \tag{12}$$

When $n = m + r$, $\rho = \frac{1}{m \cdot n}$. The optimization effect depends on the number of m and n, which is a considerable blockchain expansion plan.

When the number of encoded partitions $(m + r)$ accessible in the Fabric is less than m, the changed blockfiles cannot be recovered. That is, when all the nodes storing blockfiles are offline or the number of offline organizations is greater than or equal to $\left\lceil \frac{r \cdot n}{m+r} \right\rceil$, an unrecoverable situation may occur. But this is a kind of extreme condition. As a permission blockchain, the control of identity authentication is more strict. Therefore, the fabric expansion scheme based on erasure codes can ensure the availability of blockfiles.

5. Experimental configuration and analysis

5.1. Experimental environment and evaluation indicators

In the experimental environment of this paper, the operating system is Ubuntu, and the blockchain project is based on the hyperledger fabric. In the experiment, four hosts in the same LAN are connected to a blockchain network to upload and retrive the experimental data. The experimental confguration is shown in Table. 3.

Model evaluation indicators. Obfusion matrix is often used to characterize the classification accuracy program of a classifier. In Rahman et al. (2020), the fractional calculation method of obfusion matrix is used to evaluate the model. For binary confounding matrices, the predicted results can be divided into four categories (TP, FN,FP and TN). The classification terms are described below.

Table 3 – Experimental configuration.

Environment	Version	Notes
Operating system	Ubuntu 18.04	-
Blockchain Network	Hyperledger Fabric 1.4.4	Premissioned blockchain
Go	v1.13	Development language
Docker	v19.03.1	Open source container
Snort	v2.9.9.0	Open source IDS
hikey960	CPU:4 Cortex A73	Development Board

True Postive (TP): The instance is positive and is predicted to be positive;

False Negative (FN) : The instance is positive but is predicted to be negative;

False Postive (FP): The instance is negative but is predicted to be positive;

True Negative (TN): The instance is negative but is predicted to be negative.

For each obfusion matrix, there is a set of metrics to evaluate. The following four evaluation methods are also the calculation standards of the model adopted in this experiment.

Accuracy: The proportion of the result judged correctly in the total observed value as a result of the classification model. The calculation formula is as (13).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

Precision: In all results where the model predicts a positive class, the model predicts the correct proportion. The calculation formula is as (14).

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

Recall: The model predicts the correct weight among all results whose true value is positive. The calculation formula is as (15).

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

F1 – Score: Combines the results of Precision and Recall. The calculation formula is as (16).

$$F1 - Score = 2 \cdot \frac{Precision * Recall}{Precision + Recall} \tag{16}$$

5.2. Experimental design and analysis

<***Member rule Settings***> In the Hyperledger Fabric experiment, each node is assigned an organization(Org) to facilitate the independence of the business. Two channels are set, RSchannel and GTchannel. The three experiment nodes will be set as RSP to be allocated to Org1, Org2, and Org3, and set one experiment node as GTP to be allocated to Org4. According to the business requirements of parties, Org1, Org2 and Org3 of

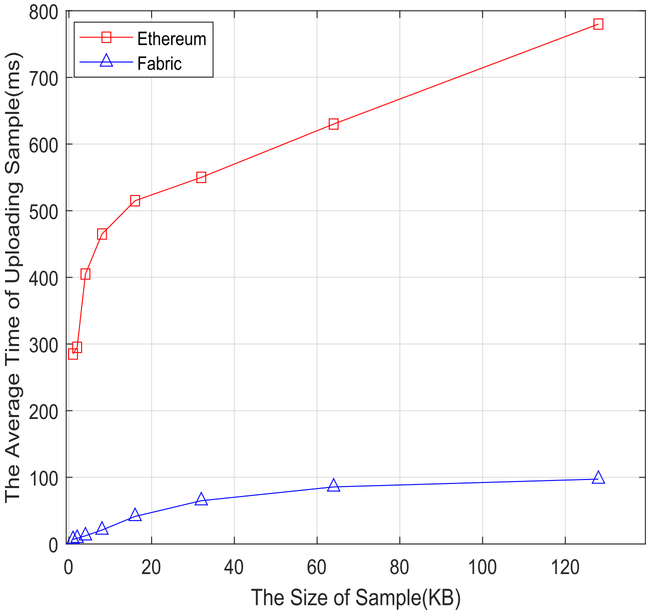


Fig. 5 – the comparison of uploading time between Ethereum and Fabric.

three RSP organizations upload model parameters in RSchannel through RSchaincode. The Org4 of GTP organizations uploads global model in GTchannel through GTchaincode. The separate channel is a blockchain ledger, which is separated by different channels, so that the training parameters and training model can have independent ledger database, and the retrieval efficiency of the ledger can be improved. When all versions of the global model need to be extracted, the entire account of GTchannel can be traced locally to facilitate direct extraction of the entire training model. When data retrieval is performed according to *GetHistoryForKey GHFK*(k), all states of k from genesis block to the last block of the given range are scanned (Gupta et al., 2018). If store both RSP and GTP business in the same ledger,will be to increase the time of data retrieval and tracing, and cause tedious work for separating training parameters and training model.

5.3. *Analysis of experimental results*

The experimental part analyzes and discusses the BFL-CIDS scheme from three angles. The first part is the time comparison of uploading data between public blockchain Ethereum and permissioned blockchain Fabric. The more efficient execution rate of the permissioned blockchain is the reason chosen by this scheme. The second part is the discussion of the optimal block file segmentation size in the Fabric storage scheme based on erasure codes. By comparing the fixed file $\frac{S_m}{m}$ block storage time, the minimum time for uploading and retrieving the fabric that meets the actual application is selected. The third part compares the alarm recognition classification scores of RSP-AFI algorithm and several popular machine learning algorithms in KDDCup99 data set. The alarm classification accuracy of the filtering algorithm is tested by the recognition rate of different identifiers. And the comparison results of AUC value of the proposed model after 100 iteration training with the graph convolutional neural network (Yao et al., 2019) and DNN (Vinayakumar et al., 2019) scheme models.

Ethereum (Zhu et al., 2018) is a public blockchain that can also perform specific business functions through smart contracts. Upload sample data of different sizes to Ethereum and Hyperledger Fabric to obtain the upload time of samples of different sizes. Fig. 5 is the comparison diagram of sample upload time in Ethereum and Fabric.Experimental data show that the time to store data on permissioned blockchain is better than that in public blockchain.

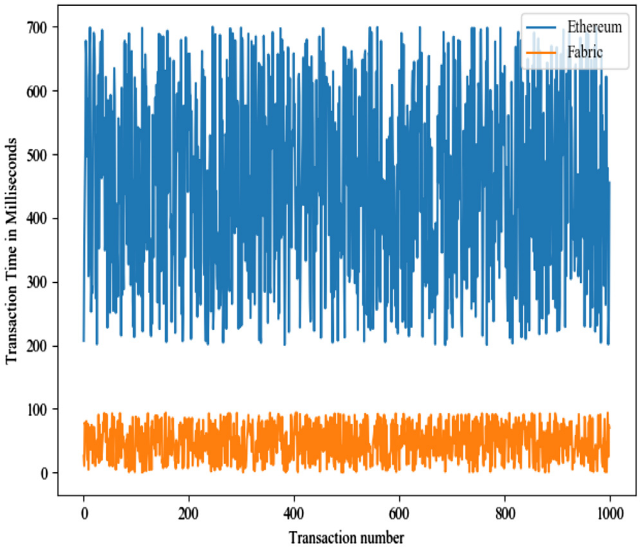


Fig. 6 – The time of 1000 transactions between Ethereum and Fabric.

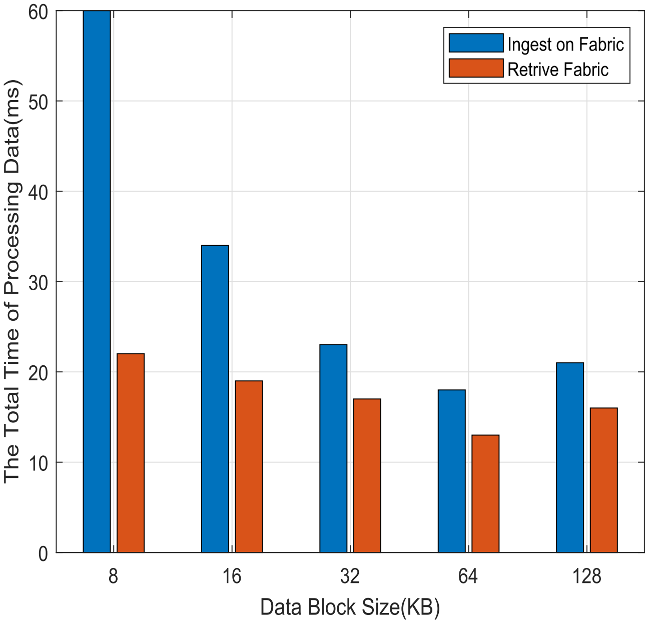


Fig. 7 – The optimal blockfile size for processing data.

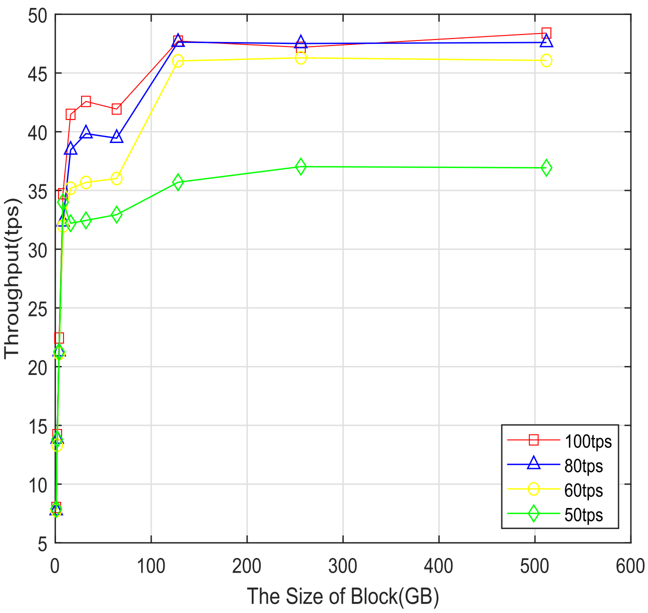
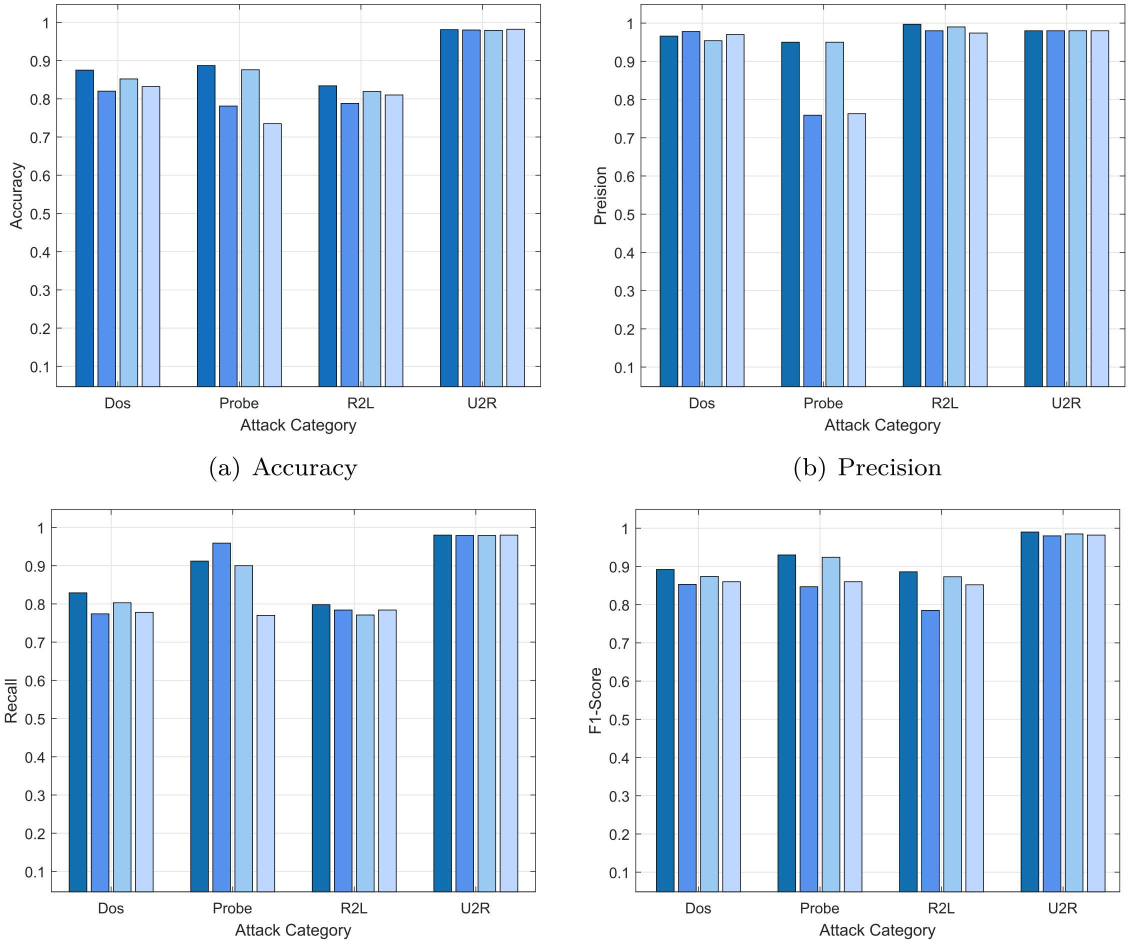


Fig. 8 – The retrieval throughput of Fabric.

In the same experimental environment, 1000 transactions are uploaded to Ethereum and Fabric. As shown in Fig. 6, the time comparison of 1000 transactions under different blockchain is shown. The total elapsed time of 1000 transactions uploaded in Ethereum is much greater than that of Fabric. In some specific application scenarios, permissioned blockchain has advantages over public blockchain in terms of security and performance.

In Section 4.2, the article proposes fabric storage based on erasure codes, and stores the training weight parameters and models of federated learning to Fabric Ledger. We propose to make the block file *BF* a fixed size S_m, so that after dividing m file blocks, each file block is a fixed size $\frac{S_m}{m}$ byte. The experiment ignores the time when blockfiles are encoded and decoded through erasure codes, and only considers the energy consumed by block uploading Fabric. Assuming S_m=1MB, $\frac{S_m}{m} = \{8, 16, 32, 64, 128\}$, Fig. 7 shows the comparison of the total time of BF ingests on Fabric and retrieve Fabric when S_m=1MB and *BF* is divided into five different data block sizes. From the experimental data, it can be seen that the smaller a single data block is, the more times it needs to be uploaded and retrieved. In this situation, transaction will lead to take more time for the process of *BF* upload and verification. With



(a) Accuracy

(b) Precision

(c) Recall

(d) F1-Score

Fig. 9 – Comparison of training model scores (left → right: RSP-AFI(MLP), RSP-AFI(DT),RandomForest and SVM).

Table 4 – The detailed description for KDDCup99 Database.

Attack Category	*Description*	*Features*	*Train*	*Test*
Normal	Normal records		97,278	60,593
DoS	Denial of service attack	$f_5\, f_6\, f_8\, f_{13}\, f_{29}\, f_{30}\, f_{32}\, f_{34}\, f_{38}\, f_{39}$ $f_{3-(\mathrm{ecr}-\mathrm{i})}$ $f_{4-(\mathrm{RSTR})}$ $f_{4-(S_0)}$	391,458	229,853
Probe	Monitor and detect other activities	f_5 f_6 f_{27} f_{34} f_{35} f_{36} f_{40} $f_{3-(ftp-data)}$ $f_{3-(\mathrm{http})}$ $f_{3-(\mathrm{private})}$ $f_{3-(\mathrm{telnet})}$ $f_{3-(smtp)}$	4,107	4,166
R2L	Illegal access to remote machines	$f_1\, f_5\, f_6\, f_{10}\, f_{11}\, f_{19}\, f_{32}\, f_{33}\, f_{34}\, f_{36}\, f_{37}$ $f_{3-(ftp-data)}$ $f_{3-(\mathrm{other})}$	1,126	16,189
U2R	Illegal access to local super user privileges by ordinary users	$f_1\, f_5\, f_6\, f_{10}\, f_{14}\, f_{17}\, f_{18}\, f_{24}\, f_{32}\, f_{34}\, f_{37}$ $f_{3-(ftp-data)}$ $f_{3-(\mathrm{other})}$	52	228
Total			494,021	311,029

the increase of data block capacity, the time of uploading file information is shorter. By comparing the experimental data, we can get the shortest time to trace the *BF* when the data block is 64KB. Therefore, while $\frac{Sm}{m}$=64KB=65536 bytes, the total time required for *BF* upload and download is the smallest.

The retrieval throughput of Fabric was benchmarking with the Hyperledger Caliper. Fig. 8 shows the throughput changes of retrieval data of different size blocks at different transaction arrival rates. The transaction arrival rate (τ) is specified in the fabric configuration $\tau = T/B$. T is defined as the number of transaction. B is defined as limits the minimum time in the ordering service for a batch transaction to package a block. Experimental data show that when the transaction arrival rate (τ) is 100 TPS and block size is 128 GB, retrieve the data in the Fabric of the highest throughput. But still need to consider the actual storage condition, can reduce the retrieval performance request, looking for the value of the optimal τ and block size, meet the actual needs of the business (Wickboldt, 2019).

The experimental data used KDDCup99 intrusion detection data set. KDDCup99 is the 9-week network connection data collected on the simulated US Air Force LAN and is divided into labeled training data and unlabeled test data. The test data and the training data have different probability distributions. The test data contains some types of attacks that do not appear in the training data, which makes intrusion detection more realistic. The identification types of KDDCup99 are shown in Table 4. A network connection of KDDCup99 is defined as a sequence of TCP data packets from the beginning to the end within a certain period of time. During this time, data is transferred from the source address to the destination address under a predefined protocol. In this experiment, 10% of KDDCup99 network traffic was selected. Among them, the training data set contains 494,021 pieces, among which 97,278 pieces of normal information and 398,743 pieces of intrusion data. The test data set is 311029, including 60593 normal data and 250,436 intrusion data. There are 24 attacks in the training set and 38 attacks in the test set. According to Selvakumar and Muneeswaran (2019), KDDCup99 after feature screening can achieve better training scores and improve training speed. So in this experiment, 13 features are extracted from each attack type to show better training results.

In the model test score experiment part, we compared two supervised learning algorithms and two semi-supervised learning algorithms. This experiment will focus on the training results of the test set in the training model. In the experimental training, all machine learning algorithms have undergone 50 iterations of training, and the training data of the semi-supervised learning algorithm is labeled with a 20% label. Fig. 9 is the training scores of KDDCup99 in RSP-AFI (MLP), RSP-AFI (DecisionTree), RandomForest and SVM. Because of the imbalance of the 10% KDDCup99 attack data set, U2R's alarm traffic is significantly smaller than the other three types of attacks. Therefore, U2R attacks with a smaller amount of data produced higher test scores in different training scores, so only the other three types of attacks are discussed. In Fig. 9(a), the MLP algorithm has a test score of 0.893 for Probe attack. Among the F1-Score scores in Fig. 9(d), the MLP algorithm also produced the highest score of 0.93 for Probe attacks. It can be seen from other test results that the prediction result of the RSP-AFI module in the MLP algorithm is better than DecisionTree. The Random Forest algorithm is to train multiple decision trees, generate models, and then comprehensively use multiple decision trees for classification. In most attack scenarios, the training effect of random forest is better than RSP-AFI (DecisionTree). The SVM algorithm is a supervised learning algorithm. It can be learned from the experimental scores that SVM and RSP-AFI (MLP) have similar training results. However, the RSP-AFI (MLP) algorithm is a semi-supervised learning algorithm, and it is a considerable result to obtain a test score similar to the supervised learning algorithm under training with 20% labeled data. In a real application scenario, there is no fully marked alarm training set. CIDS can only mark part of the data based on the alarms that have occurred. Therefore, the alarm filtering and identification of RSP-AFI (MLP) is more suitable for real scenarios.

The ROC (Receiver Operating Characteristic) curve is called the receiver operating characteristic curve. It was first used in the field of radar signal detection to distinguish between signal and noise. ROC is now widely used to evaluate the pre-

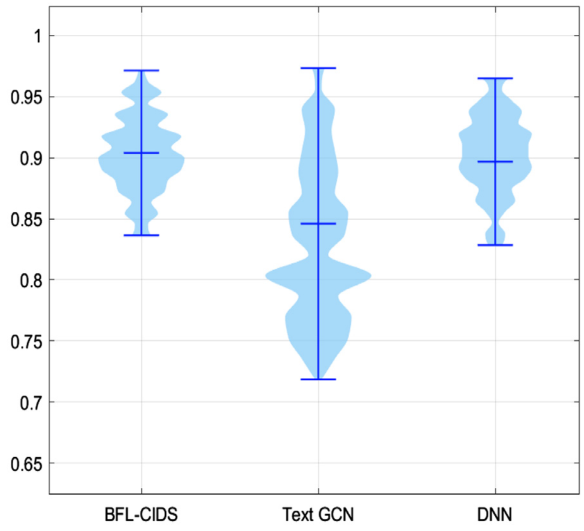


Fig. 10 – Comparison of AUC between different model algorithms.

dictive ability of models. AUC (Area Under Curve) is the area of ROC curve. The experiment uses AUC to evaluate the accuracy of the overall model. The experiment tests the overall model of the scheme from the machine learning algorithms of federated learning, graph convolutional neural network (Yao et al., 2019) and deep neural network (Vinayakumar et al., 2019), and sets the same learning rate (λ=1). Fig. 10 is a violin chart of the AUC accuracy distribution of the BFL-CIDS scheme and the other two models. In the 100 tests performed, the average AUC of the three models are 0.908, 0.849, 0,899. Different types of attacks on the same data set need to extract different features, and different accuracy affects the distribution of training results. According to the data distribution in the violin chart, the accuracy of the text-based GCN model is the largest. The training results of federated learning and DNN model are similar and relatively stable. The advantage of federated learning is that it has better privacy protection capabilities. And the semi-supervised learning algorithm is used in the BFL-CIDS model, which can exert better data processing and prediction capabilities in practical applications.

6. Conclusion

The BFL-CIDS scheme solves the problem of the data security protection and preventive maintenance of the current cloud intrusion detection system. Several advantages of BFL-CIDS scheme are as follows: (1) Strengthen the privacy protection for users: the combination of federated learning and blockchain can fully protect the security of private data; (2) Relieve client pressure: for LN, only a small amount of memory is occupied, and the calculation and storage pressure are transferred to RSP and GTP; (3) Efficient feedback: according to the existing model of permissioned blockchain, RSP provides timely feedback to some alarms to cope with the complex and changeable network environment. (4) Unified management: massive alerts samples are stored permanently based on blockchain ledger.The experimental part evaluates the scheme from the perspective of the storage efficiency of the blockchain and the intrusion warning recognition model. The proposed model was compared with important machine learning algorithms, and considerable test scores were obtained. The challenges and directions of future research are: (1) Explore the efficiency of RSP alarm analysis in real scenarios to meet the application requirements of IoT devices; (2) Set the white/black list mechanism of alarms in CIDS in IoT devices, and The end device accurately handles the intrusion crisis and reduces the bandwidth pressure caused by the upload of false alarms.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Xinhong Hei: Investigation, Conceptualization, Methodology, Project administration. **Xinyue Yin:** Software, Validation, Writing - original draft. **Yichuan Wang:** Data curation, Supervision, Funding acquisition. **Ju Ren:** Formal analysis, Visualization, Resources. **Lei Zhu:** Data curation, Writing - review & editing.

Acknowledgments

This research work is supposed by the National Key R&D Program of China (2018YFB1201500), National Natural Science Founds of China (61602376, 61773313, 61602374, 61702411), National Natural Science Founds of Shaanxi (2017JQ6020, 2016JQ6041), Key Research and Development Program of Shaanxi Province (2020GY-039, 2017ZDXMGY-098, 2019TD-014), Science Technology Project of Shaanxi Education Department (16JK1573, 16JK1552).

REFERENCES

Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In: In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16). Association for Computing Machinery; 2016. p. 308–18. doi:10.1145/2976749.2978318. New York, NY, USA

Androulaki E, Manevich Y, Muralidharan S, Murthy C, Laventman G. Hyperledger fabric: a distributed operating system for permissioned blockchains. the Thirteenth EuroSys Conference, 2018.

Benhamouda F, Halevi S, Halevi T. Supporting private data on hyperledger fabric with secure multiparty computation. In: 2018 IEEE International Conference on Cloud Engineering (IC2E); 2018. p. 357–63. doi:10.1109/IC2E.2018.00069. Orlando, FL

Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proc. Workshop Comput. Learn. Theory; 1998. p. 92–100.

Bostani H, Sheikhan M. Hybrid of anomaly-based and specification-based IDS for internet of things using unsupervised OPF based on mapreduce approach. Comput. Commun. 2017;98:52–71.

Chai D., Wang L., Chen K., et al. Secure federated matrix factorization[a/OL]. 2019. ArXiv.org(2019-6-12) http://arxiv.org/abs/1906.05108.

Cramer R, Damgård I, Escudero D, Scholl P, Xing C. $SPDZ_{2^k}$: Efficient MPC mod 2^k for dishonest majority 2018;2018:482. doi:10.1007/978-3-319-96881-0_26.

Geyer R.C., Klein T., Nabi M.. Differentially private federated learning: a client level perspective. 2017. ArXiv.org(2017) https://arxiv.org/abs/1712.07557.

Gupta H, Hans S, Mehta S, Jayachandran P. On building efficient temporal indexes on hyperledger fabric. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA; 2018. p. 294–301.

Hei X, Bai B, Wang Y, Zhang L, Zhu L, Ji W. Feature extraction optimization for bitstream communication protocol format reverse analysis. In: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), Rotorua, New Zealand; 2019. p. 662–9. doi:10.1109/TrustCom/BigDataSE.2019.00094.

Li W, Meng W, Au MH. Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments. Journal of Network and Computer Applications 2020;161:102631.

Lindell Y. In: Encyclopedia of Data Warehousing and Mining. Secure multiparty computation for privacy preserving data mining; 2006. doi:10.4018/9781591405573.ch189.

Lu Y, Huang X, Dai Y, Maharjan S, Zhang Y. Blockchain and federated learning for privacy-preserved data sharing in industrial iot. IEEE Transactions on Industrial Informatics 2019;PP(99). 1-1

Lu Y, Huang X, Dai Y, Maharjan S, Zhang Y. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. IEEE Transactions on Industrial Informatics 2020;16(3):2134–43. doi:10.1109/TII.2019.2942179. March

Ma C., Li J., Ding M., Yang H.H., Shu F., Quek T.Q.S., et al. On safeguarding privacy and security in the framework of federated learning. 2019. ArXiv.org(2019) https://arxiv.org/abs/1909.06512v2.

Majeed U, Hong CS. FLchain: Federated learning via MEC-enabled blockchain network. In: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan; 2019. p. 1–4. doi:10.23919/APNOMS.2019.8892848.

Martinez I, Francis S, Hafid AS. In: CyberC 2019 Workshop on Blockchain. Record and reward federated learning contributions with blockchain; 2019. doi:10.1109/CyberC.2019.00018.

Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Breitenbacher D, Shabtai A, et al. N-baiot: network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing 2018;17(3):12–22. JulSep

Meng Y, Kwok L. Practical Applications of Intelligent Systems Advances in Intelligent and Soft Computing, vol 124. In: Wang Y, Li T, editors. Adaptive false alarm filter using machine learning in intrusion detection. Berlin, Heidelberg: Springer; 2011.

Midi D, Rullo A, Mudgerikar A, Bertino E. Kalis - a system for knowledge-driven adaptable intrusion detection for the internet of things. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS); 2017. p. 656–66. June

Mohassel P, Zhang Y. SecureML: a system for scalable privacy-preserving machine learning. In: 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA; 2017. p. 19–38. doi:10.1109/SP.2017.12.

Rahman MA, Asyhari AT, Leong LS, Satrya GB, Tao MH, Zolkipli MF. Scalable machine learning-based intrusion detection system for IoT-enabled smart cities. Sustainable Cities andSociety 2020;61:102324. doi:10.1016/j.scs.2020.102324. ISSN 2210-6707

Ramanan P., Nakayama K.. Baffle : blockchain based aggregator free federated learning. 2019. ArXiv.org(2019) https://arxiv.org/abs/1909.07452.

Ren J, Guo H, Xu C, Zhang Y. Serving at the edge: A scalable IoT architecture based on transparent computing. IEEE Network 2017;31(5):96–105. doi:10.1109/MNET.2017.1700030.

Ren J, Zhang Y, Zhang K, Shen X. Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks. IEEE Transactions on Wireless Communications 2016;15(5):3718–31. doi:10.1109/TWC.2016.2526601. May

Selvakumar B, Muneeswaran K. Firefly algorithm based feature selection for network intrusion detection. Computers & Security 2019;81:148–55. doi:10.1016/j.cose.2018.11.005. ISSN 0167-4048

Sharma PK, Park JH, Cho K. Blockchain and federated learning-based distributed computing defence framework for sustainable society. Sustainable Cities andSociety 2020;59:102220. doi:10.1016/j.scs.2020.102220. ISSN 2210-6707

Soldatos J, Kefalakis N, Hauswirth M, Serrano M, Herzog R. Openiot: open source internet-of-things in the cloud. Lecture Notes in Computer Science 2015;9001:13–25.

Thai L., Jonathan S., Steven B., Code sample: Intel ISA-L erasure code and recovery[E B/OL]. 2015. (2015-4-16) https://software.intel.com/zh-cn/articles/intel-isa-l-erasure-code-and-recovery.

Tian Y, Wang Z, Xiong J, Ma J. A blockchain-based secure key management scheme with trustworthiness in DWSNs. IEEE Transactions on Industrial Informatics 2020;16(9):6193–202. doi:10.1109/TII.2020.2965975. Sept.

Viinikka J, Debar H, Ludovic, et al. Time series modeling for IDS alert management[c]. In: ACM Symposium on Information Computer and Communications Security. ACM; 2006. p. 102–13.

Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. IEEE Access 2019;7:41525–50. doi:10.1109/ACCESS.2019.2895334.

Wang S.. BlockfedML: Blockchained federated machine learning systems. 2019. 751–756, 10.1109/ICICAS48597.2019.00162

Wang Y, Bai B, Hei X, Zhu L, Ji W. An unknown protocol syntax analysis method based on convolutional neural network. Trans. Emerging Tel. Technol. 2020:e3922. doi:10.1002/ett.3922.

Wang Y, Zhu H, Hei X, Kong Y, Ji W, Zhu L. An energy saving based on task migration for mobile edge computing. EURASIP Journal on Wireless Communications and Networking 2019;2019. doi:10.1186/s13638-019-1469-2.

Wickboldt C.. Benchmarking a blockchain-based certification storage system. 2019. 10.13140/RG.2.2.32684.31360

Xie P., Bilenko M., Finley T., Gilad-Bachrach R., Lauter K., Naehrig M.. Crypto-nets: Neural networks over encrypted data. 2014. ArXiv preprint arXiv preprint arXiv:1412.6181.

Yao L, Mao C, Luo Y. Graph convolutional networks for text classification, 33; 2019. p. 7370–7.

Yin B, Yin H, Wu Y, Jiang Z. FDC: A secure federated deep learning mechanism for data collaborations in the internet of things. IEEE Internet ofThings Journal 2020. doi:10.1109/JIOT.2020.2966778.

Zarpelo BB, Miani RS, Kawakani CT, de Alvarenga SC. A survey of intrusion detection in internet of things. Journal of Network and Computer Applications 2017;84:25–37.

Zhang Y, Lu Y, Huang X, Zhang K, Maharjan S. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. IEEE Transactions on Vehicular Technology 2020;PP(99). 1–1

Zhou ZH, Li M. Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge & Data Engineering 2005;17(11):1529–41.

Zhu H, Wang Y, Hei X, Ji W, Zhang L. A blockchain-based decentralized cloud resource scheduling architecture. In: 2018 International Conference on Networking and Network Applications (NaNA), Xi'an, China; 2018. p. 324–9. doi:10.1109/NANA.2018.8648712.

Xin-Hong Hei received his B.S. degree and M.S. de-gree in computer science and technology from Xi'an University of Technology, Xi'an, China, in 1998 and 2003, respectively, and his Ph.D. degree from Nihon University, Tokyo, Japan, in 2008. He is currently a professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His current research in-terests include intelligent systems, safety-critical system, and train con-trol system.

Xin-Yue Yin is studying for master's degree in computer application in Xi'an University of technology. Her research area is network security and blockchain.

Yi-Chuan Wang received his Ph.D. degrees in computer system architecture from Xidian University of China in 2014. He is an ACM member and a CCF member. Now he is a Lecturer in Xi'an University of Technology and with Shaanxi Key Laboratory of Network Computing and Security Technology. His research areas include cloud computing and networks security.

Ju Ren received the B.Sc. (2009), M.Sc. (2012), Ph.D. (2016) degrees all in computer science, from Central South University, China. During 2013-2015, he was a visiting Ph.D. student in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Currently, he is a professor with the School of Computer Sci-ence and Engineering, Central South University, China. His research in-terests include Internet-of-Things, wireless networking systems, net-work computing and edge computing. Dr. Ju Ren has published more than 70 papers in top journals and conferences, including IEEE JSAC, TIFS, TMC, TCC and IEEE INFOCOM, ICDCS, etc. He is a recipient of the best paper award of IEEE ICC'19 and IEEE IoP'18, the outstanding paper award of IEEE HPCC'19 and the most popular paper award of Chinese Journal of Electronics (2015-2018). He currently serves/has served as an associate editor for IEEE Transactions on Vehicular Technology and Peer-to-Peer Networking and Applications, a guest editor for IEEE Transactions on Industrial Informatics and IEEE Network, and a TPC member of many international con-ferences including IEEE INFOCOM'20/19/18, Globecom'17, WCNC'17, WCSP'16, etc. He also served as the TPC chair of IEEE BigDataSE'19, a poster co-chair of IEEE MASS'18, a track co-chair for IEEE/CIC ICCC'19, IEEE I-SPAN'18 and VTC'17 Fall, and an active reviewer for over 20 international journals. He is a member of IEEE and ACM.

Lei Zhu received his Ph.D. degree in computer science and technology from Xi'an Jiaotong University, China, in 2014. He is currently working as Xi'an University of Technology, Department of computer science and engineering. His research interests include secret sharing scheme, data mining and graph mining.