Cross-Cluster Federated Learning and Blockchain for Internet of Medical Things

Hai Jin, *Fellow, IEEE*, Xiaohai Dai, *Student Member, IEEE*, Jiang Xiao, *Member, IEEE*, Baochun Li, *Fellow, IEEE*, Huichuwu Li, *Student Member, IEEE*, and Yan Zhang, *Fellow, IEEE*

***Abstract*—Federated learning (FL) has been gaining popularity as a way to provide privacy-preserving data sharing for the Internet of Medical Things (IoMT). As a complementary, blockchain technology is used in recent literature to make FL secure. However, existing blockchain-based FL (BFL) solutions do not perform well when data in a BFL cluster are sparse. A direct solution is to collect as many devices as possible to establish a large BFL cluster. However, these devices may locate in geographically distant areas and be separated by great distance, which further results in high communication latency. The high latency will lead to BFL's low system efficiency due to frequent communications in the blockchain consensus. In this article, we propose that the large cluster should be divided into multiple smaller clusters, each in its own geographical area and organized with a BFL. In this context, we propose CFL, a cross-cluster FL system facilitated by the cross-chain technique. CFL connects multiple BFL clusters, where only a few aggregated updates are transmitted over long distances across clusters, thus improving the system efficiency. The design of CFL focuses on a cross-chain consensus protocol, which guarantees the model updates to be exchanged securely across clusters. We carry out extensive experiments to evaluate CFL in comparison with BFL, and show both CFL's feasibility and efficiency.**

***Index Terms*—Blockchain, cross-chain technology, federated learning (FL), Internet of Medical Things (IoMT).**

I. Introduction

NOWADAYS, a growing number of medical devices are connected to build a new network, namely, Internet of Medical Things (IoMT) [1]. By aggregating the data generated in different devices, IoMT is expected to contribute to a valuable machine learning (ML) model, which can be useful in multiple scenarios, such as health monitoring, auxiliary diagnoses, and pathophoresis prediction [2]. However, the health-related data in an IoMT device are usually closely related to people's privacy, which cannot be shared or aggregated casually [3]. To address it, federated learning (FL) can be utilized [4], which enables the on-device training without transferring the data outside the device. On the other hand, the conventional FL framework relies on a central server to aggregate the model updates and orchestrate the training tasks, which is vulnerable to the malfunction of the central server. Fortunately, the emerging blockchain technology makes a complementary to it [5]. With a blockchain, the aggregation of model updates and the orchestration of the training tasks can be conducted in a distributed and secure manner [6].

However, the existing studies to combine the blockchain and FL [blockchain-based FL (BFL)] mainly focus on the system design and algorithm optimization [7], which ignore a critical problem of data sparsity in a BFL cluster [8]. For example, when the IoMT in a hospital comes into service for the first time, it can only hold sparse data samples in its early stage. Before the data accumulating to a large amount, the devices in the hospital are incapable of acquiring a good model or benefiting from ML. Besides, if the number of IoMT devices in the hospital is small, it would be too long for the BFL cluster to accumulate enough data.

A direct solution to enrich the data samples is to enlarge the size of a BFL cluster, which covers as many devices as possible. These devices may locate in different hospitals, which are far away from each other. Therefore, network latency between devices may be very high. On the other hand, the blockchain system deployed in the cluster requires frequent network communications to reach a consensus. Taken together, resulted from the frequent and high-latency communications, the consensus efficiency and the corresponding system efficiency can be fairly low. Besides, BFL requires the model updates to be disseminated across the FL cluster. As Wang *et al.* pointed out, the privacy data could be partially recovered from the model updates [9]. As a result, the larger the cluster scope, the larger the probability of data privacy leakage, especially when the cluster is established across multiple hospitals.

To deal with the data sparsity and privacy leakage problems while providing high system efficiency, we propose a cross-cluster FL framework via cross-chain technique (CFL) in this article. Multiple small clusters instead of a large cluster are built for geographically distant areas (e.g., hospitals). In each cluster, BFL is conducted with the model updates being aggregated. The aggregated updates are then exchanged across clusters, which actually enriches the updates for each cluster. Compared with the massive communications in BFL, only a

Manuscript received November 14, 2020; revised April 9, 2021; accepted May 3, 2021. Date of publication May 18, 2021; date of current version October 22, 2021. This work was supported by the National Science Foundation of China under Grant 62072197. *(Corresponding author: Jiang Xiao.)*

Hai Jin, Xiaohai Dai, Jiang Xiao, and Huichuwu Li are with the National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Laboratory, and the Cluster and Grid Computing Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: hjin@hust.edu.cn; daixh@hust.edu.cn; jiangxiao@hust.edu.cn; credolee@hust.edu.cn).

Baochun Li is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: bli@ece.toronto.edu).

Yan Zhang is with the Department of Informatics, University of Oslo, 0316 Oslo, Norway (e-mail: yanzhang@ieee.org).

Digital Object Identifier 10.1109/JIOT.2021.3081578

few aggregated updates are transmitted over a long distance in CFL. Therefore, the system efficiency can be improved largely. Besides, the aggregated updates conceal the detailed updates of each node, which can protect the data privacy better.

To enable the secure cross-cluster model exchange, we design two consensus mechanisms of the blockchain, namely, hasty consensus (HstCon) and deferred consensus (DefCon). Both HstCon and DefCon involve two subprotocols, i.e., the single-chain consensus and the cross-chain consensus. HstCon can be considered as the basis of DefCon and easy to understand, while it suffers from low system efficiency. The DefCon mechanism is proposed to improve the system efficiency. In particular, DefCon periodically elects a cluster representative to conduct the model aggregation, thus decreasing the frequency of slow cross-chain consensus.

To evaluate our system, we implement the prototypes of CFL and BFL, and make a comparison between them. The experimental results demonstrate that CFL can increase the model accuracy from 39.3% to 75.8% since it feeds more updates into the model. Besides, CFL can speed up the model convergence if the same amount of data is fed into the model. The reason for it is that CFL aggregates the computing power of multiple clusters in a training round.

In summary, our major contributions include the following.

1) We identify the difficulty in the existing BFL solutions, namely, the problem of data sparsity and the problem of low efficiency plus privacy leakage.
2) We propose CFL to deal with the difficulty by extending the single-cluster FL to cross-cluster FL, which takes advantage of cross-chain technology to provide secure communications across clusters.
3) Prototypes of CFL are implemented and extensive experiments are conducted to demonstrate its feasibility and efficiency.

II. Background and Motivation

In this section, we introduce the preliminary background of this article. Following that, we present the reason why the existing BFL schemes cannot meet the requirements in the IoMT scenario.

A. Background

With the availability of various wearable devices, medical monitors, and environmental sensors, a new network named IoMT comes into existence. As time goes by, IoMT plays a more and more important role in the healthcare field, especially when it is integrated with the ML technology. However, the conventional ML technology requires a collection of data from different devices, which may lead to serious privacy leakage in the IoMT scenario, where the data are quite privacy sensitive.

To protect users' data privacy in IoMT, the FL technology comes into researchers' consideration. FL is envisaged to conduct ML in a distributed manner without collecting the data together [4]. Taking a round-by-round workflow, FL asks each node to do a local training task in each round and then aggregates the model updates for the next-round training. In the vanilla design of FL, a centralized server is specified to orchestrate the training tasks and aggregate the model updates. However, this renders some centralization-related issues, such as single-point failure and malicious behaviors.

To tackle the issues of a central server, blockchain technology is introduced to FL [10]. Stemmed from Bitcoin [11], blockchain technology is expected to organize the peers to cooperate in a decentralized fashion [5]. With the blockchain system, the orchestration of training tasks and aggregation of model updates can be conducted distributedly via the consensus algorithm [12]. Besides, since the blockchain can be considered as a tamper-evident ledger, the model updates recorded on it are auditable and traceable, which will discourage a peer from behaving arbitrarily. The BFL can be modeled as follows. Assuming BFL is run in a hospital and all the devices in the hospital make up a cluster. In the following, we interchangeably use the term "node" to mean IoMT "device" and "cluster" to mean "hospital." Suppose the training data set is distributed among K nodes in a BFL cluster, with each node e_k possessing n_k samples, where $1 \leq k \leq K$. The training objective of node e_k is to minimize $f_k(w)$

$$f_k(w) \stackrel{\text{def}}{=} \frac{1}{n_k} \sum_{m=1}^{n_k} l\left(x_{k_m}, y_{k_m}, w\right) \tag{1}$$

where (x_{k_m}, y_{k_m}) denotes the sample indexed by m in node e_k and $l(x_{k_m}, y_{k_m}, w)$ is the loss function to do the prediction on this sample. Particularly, when the learning model is a deep learning model, the gradient descent algorithm is utilized to reduce the model loss. The calculation process is shown as follows:

$$w_{t+1} \leftarrow w_t - \eta \nabla f_k(w_t) \tag{2}$$

where t and η denote the step number and learning rate of gradient descent, respectively. $\nabla f_k(w_t)$ computes the gradients of $f_k(w_t)$.

Based on the model updates acquired by single-node learning, BFL tries to aggregate all the updates from all the nodes in the cluster. Its training objective is set as follows:

$$g(w) \stackrel{\text{def}}{=} \sum_{k=1}^{K} \frac{n_k}{n} f_k(w) \tag{3}$$

where n is the total size of training data set samples. To find the value of w which minimizes the loss function $g(w)$, a gradient aggregation method is proposed. To be more specific, in this method, each node submits its local gradient, which is aggregated by other nodes to generate a new model

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla f_k(w_t). \tag{4}$$

B. Motivation

In the scenario of IoMT [1], BFL is usually taken under consideration in a limited space. A commonly taken example is to deploy BFL in a hospital where all the devices are close to each other and connected via a high-speed local area network (LAN).

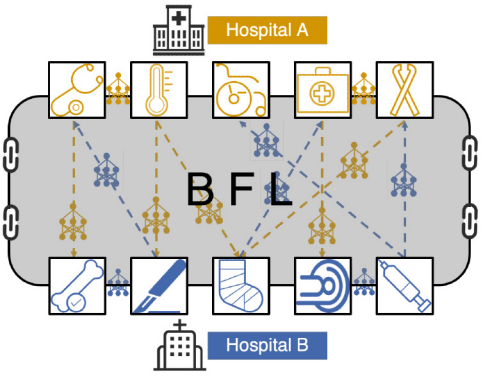


Fig. 1. Simple solution to do BFL across two hospitals.

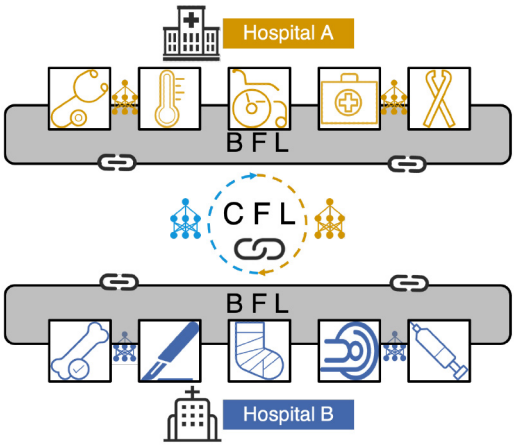


Fig. 2. System overview of CFL across two hospitals.

However, the BFL system designed for a limited space may encounter some issues. First, there may be insufficient data set samples to acquire a good model in such a small space, especially when the IoMT devices have just been put into use for a short time. In this regard, the new devices have to experience a long period, during which they cannot gain the benefits of ML technology. To address it, an area with insufficient data could probably turn to other areas for data supplement. Second, in terms of an ML task, more data usually bring a better model. Since there may be multiple areas holding the data with similar feature space or sample space, they can cooperate to create a better model than any single-area one.

The issues above entice researchers to utilize data from different areas. A simple solution is to establish a large BFL cluster, which collects massive devices dispersed in different areas/hospitals, as exemplified by Fig. 1. The communication between devices can be divided into two types: 1) intrahospital and 2) interhospital. The former is implemented via the high-speed LAN within a hospital, while the latter is supported via the low-speed wide area network (WAN) across the hospitals. In the BFL system, the blockchain consensus must be done among the devices located in different hospitals. Since interhospital communication may bring a long latency, it may take a long time to reach the consensus and result in the slow FL process. Therefore, establishing a large BFL cluster across multiple hospitals seems to be impractical.

Fortunately, maybe we can look at the problem from another angle. Since the BFL framework is more practical in a small area (e.g., a hospital) to provide high efficiency, a potential direction is to run BFL in each hospital separately and exchange BFL learning models across hospitals. In this regard, the long-latency communications between hospitals can be reduced by a substantial margin.

III. CFL Design

To address the difficulties of data sparsity and high-latency communication in BFL, we propose a cross-cluster FL framework (i.e., CFL), which connects multiple BFL clusters to build a learning model. Fig. 2 shows the system overview of CFL, which also takes the scenario of two hospitals as an example. More specifically, an instance of BFL is run in each hospital and all the devices in a hospital make up a cluster. At the end of each training round in FL, the model updates are exchanged between clusters. In this section, we start with a mathematical model of CFL. After that, we expand on the design of the CFL framework. To facilitate the secure data exchange, two sets of consensus mechanisms are proposed, namely, HstCon and DefCon. The former can be considered as the basis of the latter, while it suffers from low system efficiency. In contrast, DefCon improves the system efficiency by deferring and merging the consensus processes.

A. Mathematical Model

In our proposed CFL, we unite the model updates (e.g., gradients in the deep learning model) of multiple BFL clusters and thus exploit more data features to generate a better model. Compared with the BFL made up of all the nodes across clusters, CFL is expected to gain comparable model performance, while it can work more efficiently. In this section, we describe the mathematical model of CFL, which is based on the BFL model presented in Section II-A.

Consider that there are M BFL clusters, each of which C_i has a training data set containing n_i samples, where $1 \leq i \leq M$. In the following, we build the CFL model to connect M BFL clusters. From the view of the whole cluster, the training objective of CFL is the same as (3). To find the optimal w, the gradient descent algorithm is also utilized. At each step of gradient descent, w is computed by BFL in each cluster according to (4). w is then exchanged between clusters, which is aggregated by each cluster.

Compared with the gradient updates generated by (4) in BFL, these aggregated updates in CFL are given by

$$\begin{aligned} w_{t+1} &\leftarrow w_t - \eta \sum_{i=1}^{M} \frac{n_i}{n} \sum_{k=1}^{K_i} \frac{n_{i_k}}{n_i} \nabla f_{i_k}(w_t) \\ &= w_t - \eta \sum_{i=1}^{M} \sum_{k=1}^{K_i} \frac{n_{i_k}}{n} \nabla f_{i_k}(w_t) \\ &= w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla f_k(w_t) \end{aligned} \tag{5}$$

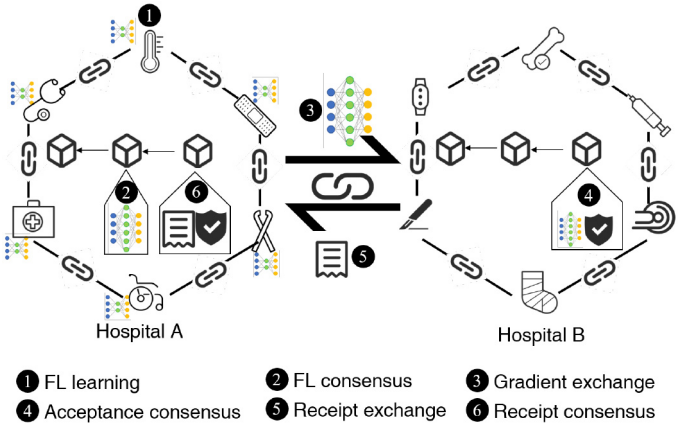


Fig. 3. Example of cross-cluster gradient aggregation.

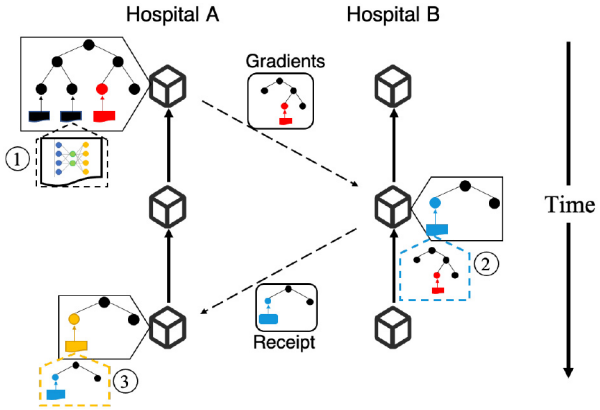


Fig. 4. Structures of transactions.

where K_i and K represent the number of nodes in C_i and the total number of nodes across the clusters, respectively. Besides, n denotes the number of samples in all the clusters. Therefore, $n = \sum_{i=1}^{M} n_i$ and $K = \sum_{i=1}^{M} K_i$. By comparing the last expression in (4) and (5), we can observe that the gradient descent process in CFL is the same as that in BFL. In other words, CFL can generate a model comparable to that of BFL if they are run on top of the same data set samples.

B. Our Proposed Architecture

Fig. 3 depicts the overview of CFL, which demonstrates the cross-cluster gradient aggregation between two hospitals. All the IoMT devices in a hospital make up a cluster to conduct FL and a permissioned blockchain system is deployed in each cluster.

1) Cross-Cluster Gradients Aggregation: As exemplified in Fig. 3, the gradients generated in hospital A are transmitted to hospital B and then aggregated in the latter. Concretely speaking, each device/node in hospital A does the on-device training to generate a local model update (❶ FL training). Gradients of the model update are then disseminated within the hospital, which are then recorded on the blockchain ledger via the intrachain consensus for FL (❷ FL consensus). The gradients from different devices are aggregated during the consensus process. In fact, the above three steps experience a round of BFL in hospital A.

Then, the aggregated updates are transmitted from hospital A to hospital B (❸ gradient exchange). Once receiving the updates, the device in hospital B will do a dual validation against the updates, whose details will be presented in Section IV. The correctness of the updates is confirmed by the intrachain consensus for acceptance (❹ acceptance consensus), which will be recorded as a transaction (receipt transaction) on the blockchain ledger in hospital B. The receipt indicating the confirmation will be transmitted back to hospital A (❺ receipt exchange), which is then validated by the devices in hospital A. The validation results from different devices are coordinated via the intrachain consensus for receipt (❻ receipt consensus), whose consensus results will be recorded as another transaction on the ledger in hospital A.

2) Data Structures: Since the data structures related to ML are identical to that in the conventional one, we focus on the structures relevant to the blockchain system here. There are mainly four types of transaction structures (i.e., gradient transaction, confirmation transaction, application transaction, and reward/punishment transaction), corresponding to different consensus targets in CFL. The latter two types of transactions are closely correlated with the DefCon protocol, which will be presented in Section III-D. In general, different types of transactions can be packaged in a block, where all the transactions are organized as a Merkle tree.

The gradient transaction is proposed by each device after the local on-device training, which contains the gradient updates, as shown by ① in Fig. 4. It should be pointed out that apart from the original gradient transactions proposed by the devices, there is an additional transaction containing the aggregated gradients, as the red part in ① shows. The confirmation transaction is proposed by the block packager, which is used for either the acceptance consensus or the receipt consensus. Accordingly, a transaction of the confirmation type can include either the gradients or receipts, as shown by ② and ③ in Fig. 4, respectively.

C. Hasty Consensus: HstCon

The consensus process in CFL includes two subprotocols, namely, the intrachain consensus and the interchain consensus. The former can be implemented by the conventional single-chain blockchain consensus (e.g., practical Byzantine fault tolerance (PBFT) [13]), while the latter is implemented by a two-phase cross-chain consensus (2PCC) mechanism. 2PCC enables the secure data exchange between two clusters, which is inspired by the two-phase commit (2PC) protocol in the database field [14]. The combination of the conventional single-chain consensus and 2PCC is named as the HstCon protocol, since nodes in this protocol take action in haste once they receive the model updates.

Fig. 5 demonstrates the workflow of the HstCon protocol. Taking a close look at the 2PCC mechanism, it consists of two phases: 1) *prepare* and 2) *merge/discard*. In the *prepare* phase, for each cluster, the model updates received from the other cluster are validated through the single-chain consensus.

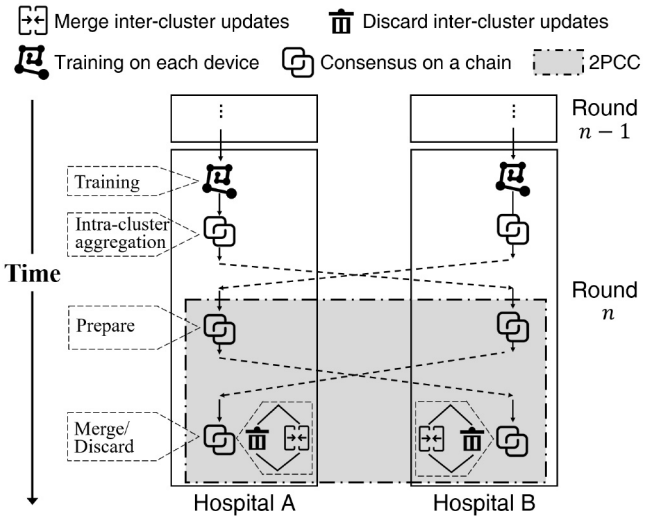


Fig. 5. Workflow of the HstCon algorithm.

More precisely, each device will validate the received updates on the local data set, and send a *merge/discard* decision to the remote cluster. Next, the devices in a cluster will do the consensus based on the local validation result and the decision from the remote, which starts the *merge/discard* phase. If both the local result and the remote decision indicate the acceptance of the received updates, the local updates and the received updates will be merged as the parameters for the next-round training. Otherwise, the local updates and the received updates will not be merged. In this regard, the received updates will be discarded by the cluster, and only the local updates are input to the next-round training.

D. Deferred Consensus: DefCon

Although CFL with HstCon seems to work, it encounters the serious challenge of low system efficiency. As can be seen from Fig. 5, each round in HstCon involves three local consensuses in each cluster. One consensus process is responsible for aggregating the updates from different nodes in a cluster, while the other two play the parts of the cross-cluster data exchange. Since the consensus process in the blockchain takes a long time, especially when the number of nodes is relatively large, the system efficiency of CFL with HstCon can be quite low.

To deal with this challenge, DefCon is proposed, which introduces a representative for each cluster and devises a corresponding reward/punishment mechanism. In general, to weaken the bad effects of frequent consensuses, DefCon elects a representative in each cluster to orchestrate intracluster learning and coordinate intercluster learning. To prevent the representative from doing evil, DefCon asks the representative to mortgage some assets and establish the reward/punishment mechanism according to its acts.

1) Election of Representative: The representative is elected every k rounds. In other words, the tenure of a representative is k rounds. At the end of a tenure, each peer can apply to be the representative of the next tenure. By issuing an application transaction, a peer becomes a representative candidate.

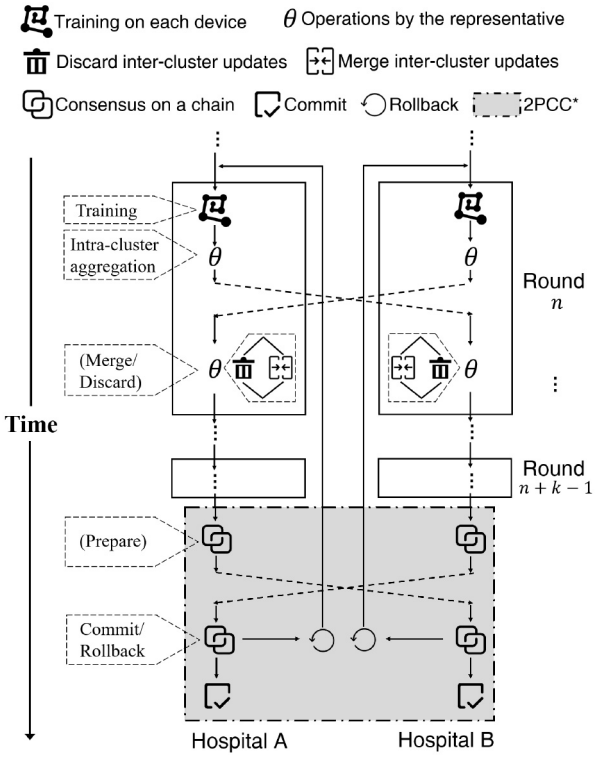


Fig. 6. Workflow of the DefCon algorithm.

The candidate has to mortgage some assets in the application transaction. The mortgage value is in connection with two aspects. First, the reward/punishment amount is positively associated with the mortgage value, which will be detailed in Section III-D3. Second, the election process of the representative relies on the mortgage value. Assume that the ith candidate takes the assets of value m_i as the mortgage, and all the mortgage values constitute a set S_m. S_m is then sorted by comparing the mortgage values. Following this, the top t elements are picked out to form a nomination pool. From the nomination pool, the representative is finally selected with a random algorithm. Presently, the random algorithm is simply implemented by taking the hash of the last block to modulo t. All the application transactions will be recorded in the blockchain, to make the election process verifiable and traceable. Compared with the gradient or confirmation transactions, the data structures of the application transactions add two additional fields. One denotes the value mortgaged by the node, and the other represents the number of terms this node is running for.

2) Workflow With DefCon: Fig. 6 depicts the DefCon workflow with the representative to improve system efficiency. The workflow is divided into successive cycles, each of which corresponds to the tenure of a representative. A cycle consists of k rounds of tentative cross-cluster learning tasks and a modified 2PCC (i.e., 2PCC*) process. The latter is used to confirm the results of tentative learning tasks in this cycle. Different from a cross-cluster FL round in HstCon, a round in DefCon involves neither the intrachain consensus nor the interchain consensus. Instead, the representative is responsible for aggregating the updates from different nodes in this cluster. Besides, it also makes the decision to merge or discard

the updates from the remote cluster. Since there is no inefficient consensus in a round, the system efficiency of CFL with DefCon is expected to be improved largely.

At the end of a cycle, a 2PCC* process is conducted to examine the acts of the representative in this cycle. As the gray box in Fig. 6 shows, the 2PCC* process involves two steps, which is similar to the original 2PCC process. Both of these two steps rely on the local consensus in a cluster. In the "prepare" step, all the peers in a cluster cooperate to reach a consensus to accept or reject all the representative's operations in the past k rounds. To be more specific, each peer will do two validations. One validates the model before the starting of this cycle, while the other validates the model at the end of this cycle. If the latter validation results are better than the former ones, the peer will vote to accept all the past operations. Otherwise, the peer will vote to reject it. Besides, the election of the new representative is conducted in the prepare step.

The consensus results in the prepare step will be sent to the remote cluster, which brings the system into the second step (i.e., "commit/rollback" step). If both of the consensus results from local and remote clusters indicate an acceptance, the operations in the past cycle will be committed, and the system will enter into a new cycle. Otherwise, the operations in the past cycle will be rolled back, and all the operations in the past cycle will be invalid. It should be noted that the rollback is not a fork on the blockchain. Only the updates in the past cycle are discarded, while all the operations in the past cycle are kept on the blockchain ledger.

3) Reward/Punishment Mechanism: To stimulate the representative to behave honestly, a reward/punishment mechanism is introduced in DefCon. As presented in Section III-D1, the representative has to mortgage some assets on the blockchain. Let the value of mortgage assets as v. Recall the prepare step in Section III-D2, each peer will figure out two validation results. The validation results will be aggregated via the blockchain consensus. Let the two validation results after consensus be t_b and t_a, respectively. If t_b is larger than t_a and the difference is more than a threshold value λ, the representative's operations will be sentenced as useless, and all the mortgage value will be confiscated. If t_b is larger than t_a while the difference is less than λ, the operations of the representative will also be sentenced as useless while the mortgage value will not be confiscated. The reason for it is that the validation results may include some errors, which should not be blamed on the representative. If t_a is larger than t_b, the rewards to the representative are positively associated with both the mortgage value and the difference value of t_a and t_b. The calculation of the reward/punishment value rp is expressed more clearly by (6), where μ is a factor to affect the value

$$\text{rp} = \begin{cases} -v, & t_b - t_a > \lambda \\ 0, & 0 \le t_b - t_a \le \lambda \\ \mu \cdot v \cdot (t_a - t_b), & t_a > t_b. \end{cases} \tag{6}$$

IV. Security Analysis

In this section, we do the analysis of both system security and data security. As for the former, we analyze if CFL can resist the attacks, in particular, the poisoning attacks. In terms of the latter, we investigate if CFL can provide higher protection of data privacy.

The problem of poisoning attacks within a BFL cluster has been studied in Krum [15] and FABA [16]. Therefore, we mainly consider the poisoning attacks across the clusters. To protect the model from potential poisoning attacks, CFL demands the representative to do dual validations. One recurs to the verifiability of the blockchain system, while the other relies on the local validation of ML.

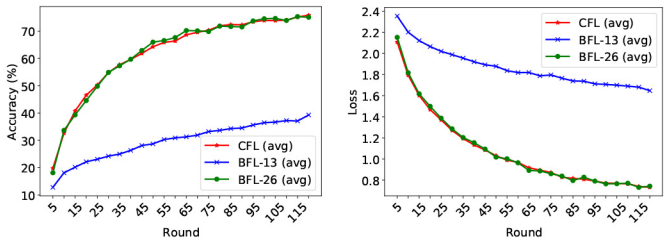
CFL takes the aggregated updates as a leaf node to build a Merkle tree, as shown in Fig. 4. When performing a round of cross-cluster FL, the Merkle branches related to the aggregated updates are exchanged between clusters. With the aid of the Merkle branch, the updates received from the remote cluster can be verified locally. Since there may be an eclipse attack, the representative is asked to validate the updates from the perspective of model performance. If the validation result is poor, the updates will be discarded by the representative directly. Otherwise, the updates will be merged with the local updates. In this regard, the possibility of poisoning attacks can be reduced to a very low level.

According to the findings of Wang *et al.* [9], data privacy could be partially recovered from the model updates exchanged between devices. In BFL of a large cluster, which covers devices in geographically distant areas (e.g., hospitals), the model updates of each device have to be disseminated across the cluster. For example, the health status recorded by a device in a hospital has to be transmitted to another hospital, which increases the probability of privacy leakage. In contrast, CFL only transmits the aggregated updates rather than the original ones outside the hospital, the updates detail of a particular device are concealed. In this way, the data privacy of the devices can be protected better.

V. Numerical Results

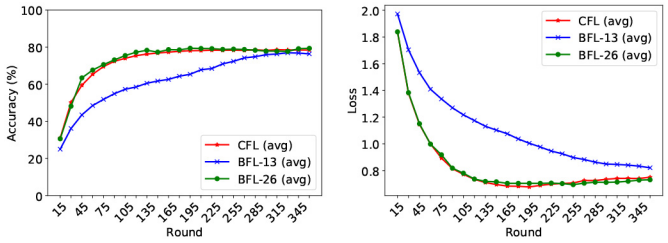
To evaluate our framework's effectiveness, we have implemented prototypes of CFL on our server machines. Corresponding to different consensus mechanisms designed for CFL, two prototypes are implemented, respectively, namely, CFL-HstCon and CFL-DefCon. Since CFL-DefCon is more efficient than CFL-HstCon, our experiments are mainly conducted on CFL-DefCon. For the rest of this section, unless otherwise stated, we refer to CFL-DefCon as CFL for short.

Each machine plays as a node/device in our experiments, which is consisted of two eight-core Intel Xeon E5-2670 CPUs, 64-GB memory, and 8-TB hard disks, with CentOS 7.2 as the operating system. The nodes are located in two racks, each of which represents a hospital and contains 13 nodes. Nodes in a rack are connected via the gigabit network, while the nodes across racks are connected via the simulated 100-Mb network. To simulate the long distance between two groups/hospitals, an extra latency of 200 ms is added to the cross-rack network. The blockchain system deployed is a permissioned blockchain, with PBFT as the consensus algorithm [13]. We take image recognization as the learning task, which is a common requirement in the scenario of IoMT. In our experiments, we run AlexNet [17] on the CIFAR-10 data



(a) (b)

Fig. 7. Comparison of model performance. (a) Accuracy by the rounds. (b) Loss by the rounds.



(a) (b)

Fig. 8. Comparison of the convergence speed. (a) Accuracy by the rounds. (b) Loss by the rounds.

set.[1] The evaluation of CFL is conducted from three aspects, including model performance, convergence speed, and system latency.

A. Model Performance

In this section, we compare the model performance between BFL and CFL. The CIFAR-10 data set is divided into two shards, each of which is fed into a rack. In a rack, the data set shard is distributed among 13 nodes at random. As for BFL, we establish clusters of different sizes, namely, BFL-13 and BFL-26. The former comprises 13 nodes in a rack, while the latter covers all the 26 nodes in two racks. In contrast, CFL is established by connecting two systems of BFL-13, each of which is run in a rack. In this way, the size of the data set in CFL is the same as that of BFL-26, while it is about twice as large as that of BFL-13. Each group of experiments is repeated five times to decrease experimental errors.

Fig. 7 shows the experimental results, including a subfigure for the accuracy and a subfigure for the loss. Each subfigure plots the average value of five experiments, with the round number as the x-axis and the testing result as the y-axis. Compared with BFL-13, it is easy to find that the model performance can be improved by CFL by a substantial margin. More specifically, the accuracy increases from 39.3% to 75.8%, while the loss decreases from 1.65 to 0.73. The main reason for the improvement is that the data in different racks can be utilized by CFL to acquire a better model. Besides, by comparing CFL with BFL-26, we can find that their model performance is close to each other. That is to say, CFL is comparable to BFL of the same size.

B. Convergence Speed

To show if the CFL framework can accelerate the model convergence, we have the data set in a single BFL be the same as the data set in CFL. We feed the entire CIFAR-10 data set to CFL, BFL-13, and BFL-26 systems separately. In each system, the data set is also distributed among all the nodes randomly. We also repeat each group of experiments five times, whose average results are shown in Fig. 8.

As can be seen from Fig. 8, after a long training period, all of CFL, BFL-13, and BFL-26 can bring the model to a similar accuracy and a close loss. The reason is that either CFL or BFL possesses the same data set. However, it takes BFL-13 over 315 rounds to achieve the model convergence, while the time taken by CFL is about 150 rounds, which is similar to BFL-26. To sum up, CFL can speed up the model convergence, compared with the BFL of a small size. The reason is that CFL aggregates the efforts from multiple racks/hospitals, which equivalently boosts the computing power for the model. Although BFL-26 also converges the model quickly from the view of rounds, it takes a long time to finish each round, which reduces the overall system efficiency.

C. System Latency

This section compares the overall system latency between different models. The system latency is defined as the elapsed time for the specific number of training rounds to finish.

We compare the latency taken by BFL-26, CFL-HstCon, and CFL, as the learning process goes on. Fig. 9 depicts the elapsed time in function of the rounds, which is divided into three parts: 1) training time; 2) consensus time; and 3) other time. The consensus time involves the consensus processes both in a chain and across the chains. By comparing bars of different systems, we can find that BFL across two racks (i.e., BFL-26) takes the highest latency, while CFL takes the lowest. In particular, when 80 rounds of training are finished, the latency taken by CFL is only 30.6% as large as that by BFL-26. The reason for BFL's high latency is its frequent communications between racks, which are required by the consensus algorithm. This can also be demonstrated by the consensus time of BFL-26 in Fig. 9. In contrast, CFL reduces the cross-rack communication by a substantial margin, which decreases the system latency. Besides, by comparing the system latency between CFL-HstCon and CFL, we can find that CFL also outperforms CFL-HstCon largely, which mainly optimizes the consensus time. This bears out the effectiveness of the DefCon protocol.

VI. Related Work

There are already several studies to combine FL and blockchain technologies. However, almost all of these works only take the system integration or algorithm optimization into consideration, which ignores the problem of data sparsity. On the other hand, the existing cross-chain technology cannot be directly adopted to connect different FL clusters, on account of its inefficiency or centralization problem. In this section, we summarize these works and talk about the gap between them and ours.

A. Blockchain-Based FL

Due to its decentralization and traceability, blockchain technology has been widely studied to make complements to FL.

[1] https://www.cs.toronto.edu/kriz/cifar.html

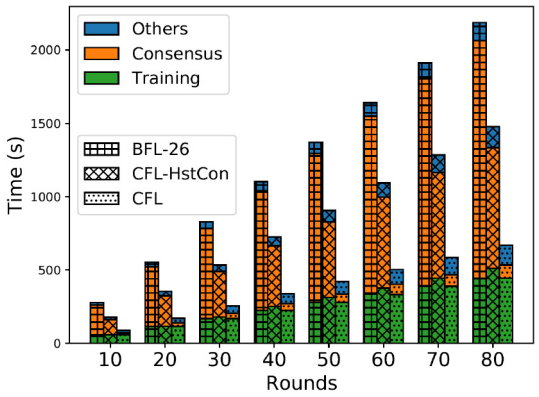


Fig. 9. Comparison of the system latency.

The application scenarios where BFL is applied vary from Internet of Vehicles (IoV) [18], through Industrial Internet of Things (IIoT) [19] to mobile-edge computing (MEC) [20], whose design objectives include robustness improvement [21] and the privacy protection [19].

In particular, Pokhrel and Choi [18], Bao *et al.* [21], and Kim *et al.* [22] proposed an architecture to enable FL in a totally distributed manner, respectively. By introducing the compensation mechanism, Kim *et al.* further encouraged devices with more data to contribute more to the global model. However, all of them ignore the fundamental problem that all the devices in a cluster may possess a small number of data samples. Lu *et al.* [19] unified the FL tasks and consensus computing tasks to implement the privacy-preserved data sharing in IIoT. However, they only consider the algorithm optimization, without thinking about the possible problem of data sparsity in an FL cluster.

To sum up, all the existing works make a hypothesis that the data in a cluster are abundant enough to build a good model. Unfortunately, this hypothesis cannot always be satisfied. Moreover, existing works try to collect all the nodes into a cluster and organize them as a single blockchain. This can bring higher communication overhead since the nodes may scatter over a large area.

B. Cross-Chain Technology

To exchange data among different blockchain systems, various cross-chain technologies are proposed [23]. These technologies can be divided into three categories: 1) notary; 2) sidechain/relay; and 3) hash-locking [24].

The notary scheme relies on a group of notaries to coordinate the actions in different chains. However, it brings the additional problem of centralization. In particular, if the notaries fail to work or behave maliciously, the state in different chains would be inconsistent. The sidechain/relay scheme enables one chain to monitor the actions on another chain and takes the corresponding actions. Each action in a chain must be agreed upon by the consensus before the other chains accept it. This reduces the efficiency of the interchain operations largely since the consensus usually brings a long latency. The hash-locking scheme supports cross-chain atomic operations without relying on any third-party entities. However, it has a limited usage scenario, which can particularly facilitate the assets exchange. In conclusion, the existing cross-chain technologies cannot be adopted in our cross-cluster FL directly, due to the problems of either centralization or inefficiency.

VII. Conclusion and Discussion

BFL technology is expected to open up massive possibilities for the IoMT scenario. However, the existing BFL schemes suffer from the problems of data sparsity and poor efficiency. In this article, we put forward to divide the nodes scattered over a large area into multiple small BFL clusters and propose CFL to connect these clusters. The aggregated updates are exchanged between clusters to enrich the data samples for each cluster. Since the size of aggregated updates is small, the communication overhead is reduced and the system efficiency is improved largely.

From another point of view, although BFL and CFL can improve the training results by exchanging model updates, they may bring heavy burdens of computation and communication on these devices. In contrast, most IoMT devices usually possess constrained resources, which can hardly bear these burdens. To deal with this problem, the technology of edge computing [25], [26] could be a good option. In other words, maybe we can attempt to combine CFL and edge computing technology in our future work.

References

[1] G. J. Joyia, R. M. Liaqat, A. Farooq, and S. Rehman, "Internet of Medical Things (IoMT): Applications, benefits and future challenges in healthcare domain," *J. Commun.*, vol. 12, no. 4, pp. 240–247, Apr. 2017.

[2] I. V. Pustokhina, D. A. Pustokhin, D. Gupta, A. Khanna, K. Shankar, and G. N. Nguyen, "An effective training scheme for deep neural network in edge computing enabled Internet of Medical Things (IoMT) systems," *IEEE Access*, vol. 8, pp. 107112–107123, 2020.

[3] F. Alsubaei, A. Abuhussein, and S. Shiva, "Security and privacy in the Internet of Medical Things: Taxonomy and risk assessment," in *Proc. 42nd IEEE Conf. Local Comput. Netw. Workshops (LCN Workshops)*, Nov. 2017, pp. 112–120.

[4] B. McMahan, E. Moore, D. Ramage, and B. A. Arcas, "Federated learning of deep networks using model averaging," Feb. 2016. [Online]. Available: arXiv:1602.05629.

[5] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData)*, Jun. 2017, pp. 557–564.

[6] M. Seliem and K. Elgazzar, "BIoMT: Blockchain for the Internet of Medical Things," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Jun. 2019, pp. 1–4.

[7] Y. Zhao *et al.*, "Privacy-preserving blockchain-based federated learning for IoT devices," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1817–1829, Feb. 2021.

[8] S. Banou *et al.*, "Beamforming galvanic coupling signals for IoMT implant-to-relay communication," *IEEE Sensors J.*, vol. 19, no. 19, pp. 8487–8501, Oct. 2019.

[9] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 2512–2520.

[10] S. Awan, F. Li, B. Luo, and M. Liu, "Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Nov. 2019, pp. 2561–2563.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Nov. 2019. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[12] L. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. 41st Int. Convent. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, May 2018, pp. 1545–1550.

[13] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Oper. Syst. Design Implement. (OSDI)*, Feb. 1999, pp. 173–186.
[14] C. Mohan, B. Lindsay, and R. Obermarck, "Transaction management in the R* distributed database management system," *ACM Trans. Database Syst.*, vol. 11, no. 4, pp. 378–396, Dec. 1986.
[15] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 119–129.
[16] Q. Xia, Z. Tao, Z. Hao, and Q. Li, "FABA: An algorithm for fast aggregation against Byzantine attacks in distributed neural networks," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2019, pp. 4824–4830.
[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1097–1105.
[18] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734–4746, Aug. 2020.
[19] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.
[20] M. H. U. Rehman, K. Salah, E. Damiani, and D. Svetinovic, "Towards blockchain-based reputation-aware federated learning," in *Proc. Conf. Comput. Commun. Workshops (INFOCOM)*, Aug. 2020, pp. 183–188.
[21] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "Flchain: A blockchain for auditable federated learning with trust and incentive," in *Proc. 5th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2019, pp. 151–159.
[22] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2019.
[23] H. Jin, X. Dai, and J. Xiao, "Towards a novel architecture for enabling interoperability amongst multiple blockchains," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1203–1211.
[24] V. Buterin, "Chain interoperability, R3 research," Sep. 2016. [Online]. Available: https://allquantor.at/blockchainbib/pdf/vitalik2016chain.pdf
[25] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: A decentralized computation offloading algorithm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 2, pp. 411–425, Feb. 2021.
[26] Z. Ning *et al.*, "Mobile edge computing enabled 5G health monitoring for Internet of Medical Things: A decentralized game theoretic approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 463–478, Feb. 2021.

Hai Jin (Fellow, IEEE) received the Ph.D. degree in computer engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1994.

He is a Cheung Kung Scholars Chair Professor of Computer Science and Engineering with HUST. In 1996, he was awarded a German Academic Exchange Service Fellowship to visit the Technical University of Chemnitz, Chemnitz, Germany. He worked with The University of Hong Kong, Hong Kong, from 1998 to 2000, and as a Visiting Scholar with the University of Southern California, Los Angeles, CA, USA, from 1999 to 2000. He has coauthored 15 books and published over 700 research papers. His research interests include computer architecture, virtualization technology, cluster computing and cloud computing, data storage, and network security.

Dr. Jin was awarded the Excellent Youth Award from the National Science Foundation of China in 2001. He is a Fellow of CCF and a member of ACM.

Xiaohai Dai (Student Member, IEEE) received the M.S. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2017, where he is currently pursuing the Ph.D. degree.

His current research interests include blockchain and distributed system.

Dr. Dai received several awards include the Outstanding Creative Award in 2018 FISCO BCOS Blockchain Application Contest and the Top Ten in FinTechathon 2019.

Jiang Xiao (Member, IEEE) received the B.Sc. degree from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2009, and the Ph.D. degree from Hong Kong University of Science and Technology, Hong Kong, in 2014.

She is currently an Associate Professor with the School of Computer Science and Technology, HUST. She has been engaged in research on blockchain, distributed computing, wireless indoor localization, and smart sensing. She has directed and participated in many research and development projects and grants from funding agencies, such as the National Natural Science Foundation of China (NSFC), Hong Kong Research Grant Council, Hong Kong Innovation and Technology Commission, and industries, such as Huawei, Tencent, and Intel, and been invited by NSFC in reviewing research projects.

Dr. Xiao received several awards include the CCF-Intel Young Faculty Research Program 2017, the Hubei Downlight Program 2018, the ACM Wuhan Rising Star Award 2019, and the Best Paper Awards from IEEE ICPADS/GLOBECOM/GPC.

Baochun Li (Fellow, IEEE) received the B.E. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 1995, and the M.S. and Ph.D. degrees from the Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 1997 and 2000, respectively.

Since 2000, he has been with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, where he is currently a Professor. He has coauthored more than 290 research papers, with a total of more than 13 000 citations, an H-index of 59, and an i10-index of 189, according to Google Scholar Citations. His research interests include large-scale distributed systems, cloud computing, peer-to-peer networks, applications of network coding, and wireless networks.

Dr. Li was the recipient of the IEEE Communications Society Leonard G. Abraham Award in the field of communications systems in 2000. He is a member of ACM.

Huichuwu Li (Student Member, IEEE) received the B.S. degree from the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan.

His current research interests include wireless communications, indoor localization, scene analysis, and smart sensing.

Yan Zhang (Fellow, IEEE) received the Ph.D. degree in electrical and electronics engineering from Nanyang Technological University, Singapore, in 2006.

He is currently a Full Professor with the Department of Informatics, University of Oslo, Oslo, Norway. His research interests include next-generation wireless networks leading to 5G Beyond, green and secure cyber–physical systems (e.g., smart grid and transport).

Prof. Zhang is an Editor of several IEEE publications, including the *IEEE Communications Magazine*, *IEEE Network Magazine*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE INTERNET OF THINGS, IEEE SYSTEMS JOURNAL, and *IEEE Vehicular Technology Magazine*. He serves as the Chair positions in a number of conferences, including the IEEE Global Communications Conference 2017, IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications 2016, and IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids 2015. He is the IEEE Vehicular Technology Society Distinguished Lecturer. He is a Fellow of the Institution of Engineering and Technology. He was the Chair of the IEEE Communications Society Technical Committee on Green Communications and Computing. He is recognized as a Highly Cited Researcher in 2018 according to Web of Science.