Future Generation Computer Systems 117 (2021) 328–337

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Privacy-preserving blockchain-based federated learning for traffic flow prediction

Yuanhang Qi [a,b], M. Shamim Hossain [c,*], Jiangtian Nie [d,e], Xuandi Li [d]

[a] School of Computer Science, University of Electronic Science and Technology of China, Zhongshan Institute, China
[b] School of Computer Science and Engineering, University of Electronic Science and Technology of China, China
[c] Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[d] Energy Research Institute, Nanyang Technological University, Singapore
[e] School of Computer Science and Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 2 October 2020
Received in revised form 25 November 2020
Accepted 6 December 2020
Available online 10 December 2020

Keywords:
Federated learning
Blockchain
Local differential privacy
Traffic flow prediction
Intelligent transportation systems

ABSTRACT

As accurate and timely traffic flow information is extremely important for traffic management, traffic flow prediction has become a vital component of intelligent transportation systems. However, existing traffic flow prediction methods based on centralized machine learning need to gather raw data for model training, which involves serious privacy exposure risks. To address these problems, federated learning that shares model updates without exchanging raw data, has recently been introduced as an efficient solution for achieving privacy protection. However, the existing federated learning frameworks are based on a centralized model coordinator that still suffers from severe security challenges, such as a single point of failure. Thereby, a consortium blockchain-based federated learning framework is proposed to enable decentralized, reliable, and secure federated learning without a centralized model coordinator. In the proposed framework, the model updates from distributed vehicles are verified by miners to prevent unreliable model updates and are then stored on the blockchain. In addition, to further protect model privacy on the blockchain, a differential privacy method with a noise-adding mechanism is applied for the blockchain-based federated learning framework. Numerical results illustrate that the proposed schemes can effectively prevent data poisoning attacks and improve the privacy protection of model updates for secure and privacy-preserving traffic flow prediction.

1. Introduction

With the rapid development and deployment of intelligent transportation systems (ITS), a huge amount of traffic information is gathered by a traffic flow prediction (TFP) module, which provides real-time traffic analysis to estimate future states based on the historical traffic data collected from various sensors, including radars, mobile phones, cameras, etc. As an essential component of ITS, TFP is helpful in supporting traffic control departments and traffic policies to allocate road resources, alleviate traffic congestion, and improve traffic efficiency, ultimately achieving transportation management and guaranteeing driving safety [1]. At present, deep learning has been widely applied for TFP to learn generic traffic flow features and improve prediction performance. However, deep learning algorithms generally require vehicles to transmit raw data containing sensitive information (e.g., location) to a central server to train the proposed deep learning models in a centralized fashion [2–5]. If the central server is compromised, the entire TFP system suffers from a typical single-point-of-failure attack, and this may easily result in severe privacy loss for the vehicles.

To resolve the problem, we propose a federated learning (FL)-based TFP method. As introduced by [6], FL is a new machine learning paradigm that encourages participants to collaboratively train a globally shared model released by the central server. In each iteration of the global model training, each participant receives an initial global model, performs local model training, and submits local model updates (i.e., gradient parameters) to the central server without uploading the raw data. After that, the central server aggregates all the local model updates to obtain a new global model, and then releases the model. The above process of global model training continues iteratively until convergence. The application of FL can effectively remove privacy threats for vehicles by performing TFP tasks locally. However, conventional FL depends greatly on the central server, which collects and manages the local model updates, thus the security vulnerability to a single point of failure cannot be avoided [7]. Furthermore, if malicious vehicles upload false or low-quality

* Corresponding author.
E-mail addresses: qiyuanhang@zsc.edu.cn (Y. Qi), mshossain@ksu.edu.sa (M.S. Hossain), jnie001@e.ntu.edu.sg (J. Nie), cindy.li@ntu.edu.sg (X. Li).

https://doi.org/10.1016/j.future.2020.12.003
0167-739X/

local model updates to the central server, the convergence performance of the algorithm degrades, and this further influences the results of the TFP system, ultimately affecting the traffic control decisions of the traffic management department or the traffic police. Therefore, a considerate FL framework is strictly required to ensure the accuracy and reliability of the algorithm [8,9], such that the application performance of FL in the TFP system is enhanced to promote traffic management and policy enforcement for ITS.

To cope with these challenges, we focus on blockchain and further study a combination of blockchain and FL to enhance the entire TFP system with security and privacy protection. As a decentralized and distributed public ledger technology, blockchain adopts consensus mechanisms to synchronize the changes in P2P networks, without dependence on any centralized entities or third parties. Due to the remarkable features of decentralization, unalterability, traceability, and anonymity, blockchain has been widely applied in a variety of domains [10,11]. We were motivated to utilize the technology for secure and efficient implementation of FL in TFP by verifying local model updates and removing unreliable ones from malicious vehicles. At the same time, we considered the decentralized blockchain instead of the central server to record and manage all the local model updates and tackle the single-point-of-failure problem.

In this study, we integrate FL and blockchain into the TFP system for high-level privacy protection. More specifically, we introduce the application of FL for TFP by applying a lightweight yet efficient neural network model from [12] called GRU to process time series traffic data and construct a consortium blockchain for the decentralized FL-based TFP system. A consortium blockchain is exploited to employ a limited number of pre-selected miners to maintain the distributed ledger. Compared with the public and private blockchain technologies, consortium blockchain is widely accepted due to its moderate management overhead, remarkable network scalability, and lower consensus latency. We investigate potential security threats from the viewpoints of the central server and vehicles participating in FL, respectively. To protect against these security threats, we implement the entire FL procedure within a consortium blockchain-based framework and particularly describe the consensus process by using delegated Practical Byzantine Fault Tolerance (dBFT). Furthermore, we leverage the local differential privacy technology with a noise-adding mechanism to strengthen location privacy protection for participating vehicles that need to share their locations with other vehicles during vehicular communications. Finally, extensive numerical experiments based on real datasets are performed to demonstrate the effectiveness and efficiency of our proposed scheme, which is helpful to traffic management departments and traffic police in making correct traffic control decisions.

The main contributions of this paper are summarized as follows:

- We revise the GRU neural network to successfully present an FL-based TFP method. In this method, the participating vehicles use their data to perform local model training and share local model updates instead of directly transmitting individual data, thereby preserving privacy.
- We exploit the consortium blockchain to promote the application of FL for TFP in a decentralized manner. The local model updates submitted by participating vehicles are validated, with the central server replaced by a set of trusted consensus nodes to manage all the local model updates. Consequently, the security risks of FL on the central server and participants' sides are prevented.
- We apply the local differential privacy technique to provide location privacy protection for participating vehicles. When a participating vehicle uploads its local model update, considerable Gaussian noise is added to disturb the location information and finally prevent lawbreakers from collecting information from participants using a membership inference attack.

The remainder of this paper is organized as follows: Section 2 presents the related literature. Section 3 describes the FL based on GRU for TFP. Section 4 introduces the consortium blockchain for FL-based TFP and the consensus algorithm design for the consortium blockchain is proposed in Section 5. In Section 6, a local differential privacy technique is applied for privacy-preserving vehicular communications in TFP. A performance evaluation based on real datasets is chosen as the study case to demonstrate the proposed framework in Section 7. Finally, some conclusions and future works are presented in Section 8.

2. Related work

2.1. TFP for the internet of vehicles

TFP is an essential component of an ITS, which directly affects the planning, management, and control of traffic. Considering the real-time prediction of TFP, the seasonal autoregressive integrated moving average (SARIMA) and generalized autoregressive conditional heteroscedasticity (GARCH) were adopted in [13], and were implemented using adaptive Kalman filters. Unlike the work in [13], the authors in [14] not only applied SARIMA + GARCH to predict the regular part, but also adopted nonlinear methods to predict the irregular part, which effectively improved the TFP results. However, the above methods are based on the establishment of accurate models for TFP, which are rarely implemented in real life.

With the development of artificial intelligence [15,16], machine learning that trains data without an accurate model to obtain approximate model is successfully applied in TFP. As one popular machine learning algorithm, the long short-term memory (LSTM) neural network works effectively in TFP. With the additional feature extraction capability of autoencoders, an autoencoder LSTM was proposed in [17]. Furthermore, the authors in [18] proposed a hybrid LSTM and divided the short-term TFP into two stages. First, five kinds of LSTM were used to predict traffic flow with different time lags. After that, no negative constraint theory or population extremal optimization were adopted to aggregate the five models obtained in the first stage, and thus obtain the final TFP results. In addition to LSTM, some researchers have adopted the regression neural network and the generative adversarial capsule network to implement TFP in recent years. Using past and future traffic information simultaneously, the authors in [19] proposed a bi-directional regression neural network to predict traffic flow. To address regional epitaxial TFP, the authors in [20] proposed a generative adversarial capsule network based on inflow and outflow information in the central area.

2.2. Privacy and security protection for the internet of vehicles

In ITS research, many works depend on training raw data, which include considerable private information [21,22]. With the increasing attention to privacy problems, direct raw data exchange is not permitted in many cases [23,24]. Thus, FL, as a machine learning method that shares only model updates without exchanging raw data, has been used to train private data. After being it was proposed by Google in 2016 [25], FL quickly gained the attention of researchers. The authors in [26] proposed a privacy-enhanced FL. Furthermore, to solve the degradation in FL accuracy caused by the imbalance of distributed training

data in mobile systems, the authors in [27] proposed a self-balancing FL model. Similarly, FL has been used in power systems, cloud computing, language modeling, social media, and other fields [28,29]. To address the high communication cost between clients and the cloud server, a highly efficient FL was proposed in [30]. In addition, to enhance the convergence efficiency of FL, a momentum gradient descent was adopted in the local update process of FL in [31]. Nevertheless, FL in the above papers posed the following security threats: (1) the centralized mechanism was vulnerable to single-point-of-failure attacks; (2) malicious vehicles could upload false or low-quality local model updates to the central server.

To overcome these challenges, researchers use decentralized blockchain technology. Blockchain technology, with a decentralized service and consensus mechanism, verifies the data from participants and prevents the upload of unreliable and unsafe data. By combining blockchain with FL, the authors in [32] introduced a crowdsourcing framework that reduced overhead and improved security during crowdsourcing implementation. By further considering communication, consensus delays, and computation cost in Fog Computing, the authors in [33] analyzed the latency performance of blockchain-enabled FL and derived an optimal block generation rate. In addition, some works adopted a differential privacy method with a noise-adding mechanism to further enhance location privacy protection for participating vehicles, which prevented membership inference attacks. For the industrial Internet of Things to adversary attack, the authors in [34] introduced a framework that integrated differential privacy, FL, Ethereum blockchain, and smart contracts to enhance the privacy and credibility of data. Unlike the work in [34], a blockchain FL framework with differential privacy that prevented poisoning and membership inference attacks was used to enhance the security of 5G [35]. In this secure FL framework for 5G networks [36], the smart contracts of blockchain technology were used to prevent malicious or unreliable participants from transmitting false data, while differential privacy technology was used to prevent membership inference attacks.

Although the above studies employed many privacy-preserving methods, none of the methods can be directly applied to TFP. In this study, a consortium blockchain-based FL framework with differential privacy is explored to protect the privacy and security of information shared in TFP.

3. FL for TFP

In this section, we introduce the application of FL based on the GRU neural network for TFP.

3.1. Overview of FL

FL is a privacy-preserving distributed machine learning framework that enables a set of participants (e.g., vehicles and devices) to collaboratively train a global model based on their local data instead of gathering the local data for centralized model training. The individual data of the participants may contain sensitive information but remains in local storage, thereby protecting user privacy through FL. In each iteration of global model training, every participant performs the local model training task with an initialized global model, and afterward determines the updates for the local model in terms of gradients or weights, and submits them to a central server for parameter aggregation. Next, we focus on the FL procedure in a specific iteration t ($t \in \{1, 2, \ldots, T\}$). To summarize, the procedure consists of three phases as follows.

- *Phase 1, Initialization:* The participants register with the central server to participate in this procedure. The central server selects appropriate participants with strong computing and communication capabilities to accomplish the local model training tasks. Then the central server sends an initialized global model ω_0 to each participant.
- *Phase 2, Training:* Participant k ($k \in \{1, 2, \ldots, K\}$) uses the local data to train the local model by letting $\omega_t^k \leftarrow \omega_t$, and obtains the updated local model ω_{t+1}^k. In other words, the participant tries to train the local model by tackling an optimization problem of a loss function as follows.

$$\arg\min_{\omega \in \mathbb{R}} F_k(\omega), \;\; F_k(\omega) = \frac{1}{D_k} \sum_{i \in \mathcal{I}_k} f_i(\omega), \tag{1}$$

  where $x_i, y_i \in \mathbb{R}$ is a input–output vector pair of a local dataset $\mathcal{I}_k$ belonging to participant k, D_k is the size of $\mathcal{I}_k$, ω is the weights of the local model, and $f_i(\omega)$ represents a local loss function, $f_i(\omega) = \frac{1}{2}(x_i{}^T \omega - y_i)$.
- *Phase 3, Aggregation:* The participants upload the local model updates to the central server, which uses a typical aggregation algorithm, e.g., FedSGD and FedAVG, to update the global model ω_{t+1},

$$\omega_{t+1} \leftarrow \omega_t - \frac{1}{K} \sum_{k=1}^{K} F_k(\omega), \tag{2}$$

Finally, the central server sends the new global model ω_{t+1} to each participant for the next iteration of global model training.

3.2. FL-based TFP method using GRU

In this paper, we aim to achieve accurate and timely TFP through a lightweight FL scheme. Next, we apply the GRU neural network as a lightweight technique to predict future traffic and realize the application of FL in the TFP system. The GRU neural network was developed in 2014 and has become an advanced sequence prediction model to process a variety of time series data, including traffic flow and electricity data.

We use $X = \{x_1, x_2, \ldots, x_n\}$ to represent the time-series input data, and $Y = \{y_1, y_2, \ldots, y_n\}$ represent the time-series output data. Let $H = \{h_1, h_2, \ldots, h_n\}$ be the hidden state of the cells. At the tth ($1 \le t \le n$) time step, the value of update gate z_t is calculated by

$$z_t = \sigma(W^{(z)} x_t + U^{(z)} h_{t-1}), \tag{3}$$

where $W^{(z)}$ is the weight matrix, h_{t-1} holds the cell state of the previous time step $t-1$, and x_t is the input vector of the tth time step. Similarly, the reset gate r_t is calculated by

$$r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1}). \tag{4}$$

The candidate activation $h_t{}'$ is denoted as

$$h_t{}' = \tanh(W x_t + r_t \odot U h_{t-1}), \tag{5}$$

where $r_t \odot U h_{t-1}$ indicates the Hadamard product of r_t and $U h_{t-1}$. Then, at the current time step t, the final memory is calculated by

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t{}'. \tag{6}$$

To integrate the above GRU neural network into the FL algorithm, we improve the original GRU neural network of a vehicle v in the TFP system as follows.

$$z_v^t = \sigma(W^{(z_v)} + U^{(z_v)} h_v^{t-1}), \tag{7}$$

$$r_v^t = \sigma(W^{(r_v)} + U^{(r_v)} h_v^{t-1}), \tag{8}$$

Algorithm 1: An algorithm utilizing the GRU neural network for the application of FL in the TFP system.

Input: A set of vehicles $\mathcal{V} = \{v_1, v_2, \cdots, v_k\}$, on-Vehicle Machine Learning (oVML) models $\mathcal{M} = \{M_1, M_2, \cdots, M_k\}$, the global batch size B, the number of local epochs E, the learning rate α and the local optimization function $\nabla\mathcal{L}(\cdot;\cdot)$.

Output: ω.

Initialize ω^v;
foreach *Iteration* $t = 1, 2, \cdots$ **do**
  $\{V_k\} \leftarrow$ select the vehicles from $\mathcal{V}$ participate in this iteration of the global model training;
  Local RSUs broadcast global model ω^v to the vehicles in $\{V_k\}$;
  foreach *vehicle* $v \in \{V_k\}$ ***in parallel*** **do**
    Initialize $\omega_t^v = \omega^v$;
    ω_{t+1}^v ($\omega_{t+1}^v \leftarrow$ LocalUpdate(v, ω_t^v);
  $\omega_{t+1} \leftarrow \frac{1}{|\{V_k\}|} \sum_{v \in V_k} \omega_{t+1}^v$;
LocalUpdate(v, ω_t^v): // Run on oVML models $\mathcal{M}$, i.e., GRU algorithms ;
$\mathcal{B} \leftarrow$ (split $\mathcal{S}_o$ into B);
for *each local epoch i from 1 to E* **do**
  if *batch* $b \in \mathcal{B}$ **then**
    $\omega \leftarrow \omega - \alpha \cdot \nabla\mathcal{L}(\omega; b)$;
return ω to $\{V_k\}$.

$$h_v^{t\prime} = \tanh(Wx_v^t + r_v^t \odot Uh_v^{t-1}), \tag{9}$$

$$h_v^t = z_v^t \odot h_v^{t-1} + (1 - z_v^t) \odot h_v^{t\prime}. \tag{10}$$

where $X = \{x_v^1, x_v^2, \ldots, x_v^n\}$ denotes the vehicle's time-series input data, $Y = \{y_v^1, y_v^2, \ldots, y_v^n\}$ denotes the vehicle's time-series output data, and $H = \{h_v^1, h_v^2, \ldots, h_v^n\}$ denotes the hidden state of the cells. Based on the improvements, we present an algorithm to utilize the GRU neural network for the application of FL in the TFP system. In the algorithm, Roadside Units (RSUs) act as local access points of participating vehicles. The objective function is shown as follows:

$$\arg\min_{\omega \in \mathbb{R}^d} J(\omega) := \sum_{k=1}^{K} \frac{\sum_{i \in D_v} f_i(\omega) + \lambda h(\omega)}{D}. \tag{11}$$

The pseudocode of this algorithm is introduced in Algorithm 1.

4. Decentralized TFP system with FL

4.1. Threat model

In traditional FL, both the participants and the central server have a high degree of autonomy. The failure of the participants or the central server has a negative impact on FL performance. Next, we discuss security threats from two main perspectives: server vulnerabilities and participant vulnerabilities. First, we consider the security risks for the malicious central server, which are summarized as follows.

- **Single-Point-of-Failure Attack**: The central server is a vital coordinator of FL that collects and aggregates all the local model updates. If the central server suffers from a typical single-point-of-failure attack, the execution of model update aggregation will fail, and a new global model will not be distributed to the participants for subsequent local model training. This means that the entire FL algorithm terminates.
- **Membership Inference Attack**: Because the central server holds knowledge about the local model updates from all the participants, the central server has an opportunity to record the parameter information for a participant's local model. The central server can further utilize the parameter information of the local model to perform membership inference attacks to steal the local dataset of a participant. In this case, the participants cannot know whether their parameter information is recorded by the central server. This seriously violates the principle of FL.

Second, malicious participants may launch poisoning attacks to undermine the accuracy of the global model training. Malicious participants can launch poisoning attacks in a variety of ways, which are summarized as follows.

- **Byzantine Attack**: In the Byzantine attack, a malicious participant may arbitrarily provide the central server with false or low-quality local model updates instead of valid ones, which would poison the server-side global model training [37]. As a result, the accuracy of the global model would sharply decrease.
- **Label Flipping Attack**: In the execution of a GRU neural network, malicious participants may modify the labels (i.e., the target for oVML model learning) of local datasets to provide low-quality model updates [38], thereby affecting the entire learning phase. For example, a vehicle can modify the traffic flow data at time t, which will affect the oVML prediction of traffic flow at time $(t + 1)$.
- **Data Poisoning Attack**: Participants have all the permissions of the local training dataset, so malicious participants may add extreme noise that destroys the quality of the local datasets to generate unreliable local model updates [39]. This is very easy for malicious participants to do without the central server being aware.

The attacks described above are related to local model updates. Thus, we are motivated to detect malicious local model updates to provide security and privacy guarantees for the application of FL in the TFP system. In this paper, we employ blockchain technologies to perform model update verification tasks without the use of third parties.

4.2. Consortium blockchain for FL-based TFP

As shown in Fig. 1, a dedicated consortium blockchain is established to achieve a decentralized TFP system with FL. To facilitate the system functionality, we first select a certain number of RSUs as authorized miners. To this end, the hardware configuration of these RSUs is upgraded so they have powerful computation, storage, and communication capabilities to validate the local model updates submitted by distributed vehicles, and to identify low-accuracy and unreliable updates. Through elaborately designed consensus algorithms, they generate a new block with records of the qualified local model updates. With the implementation of the consortium blockchain, an iteration of global model training in the application of FL for TFP includes the following phases:

- **Local Model Training**: Each vehicle executes the GRU neural network by using its local dataset to obtain the local model updates and sends these updates to the closest miner.
- **Model Update Verification**: To obtain certain token rewards, the miners are employed to receive and verify the collected local model updates. When a miner uses an effective method to successfully filter the false and low-quality local model updates, the miner generates a new data block to store all the qualified local model updates.

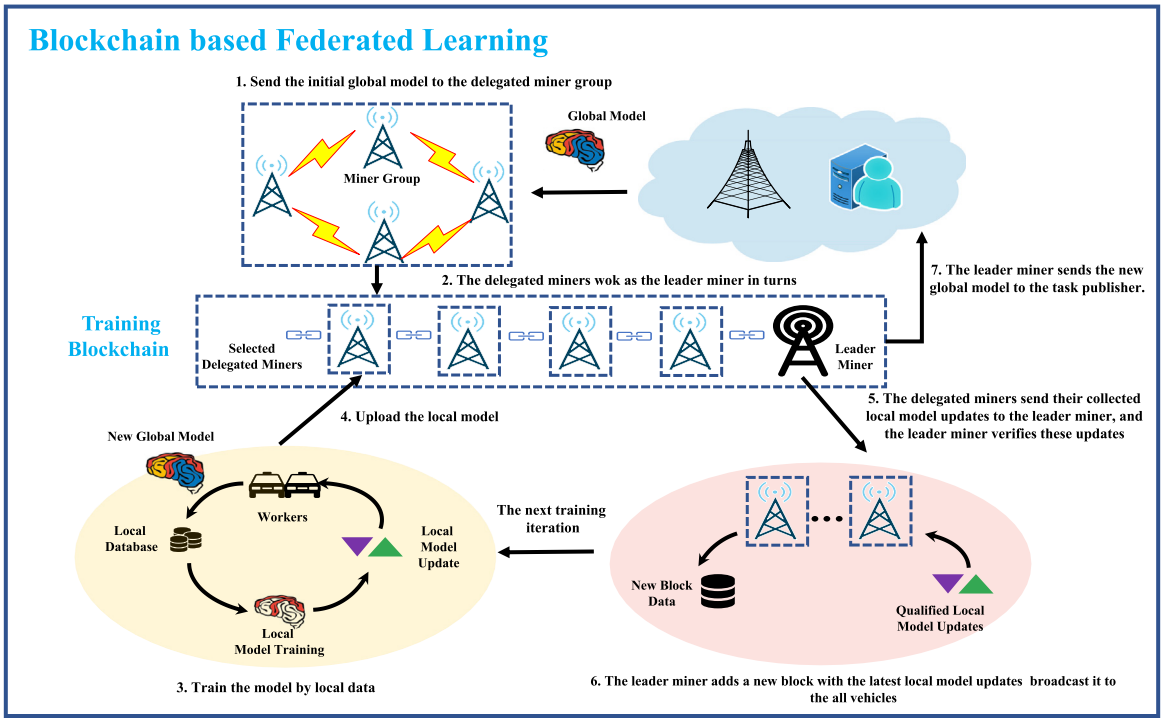

Fig. 1. Privacy-preserving blockchain based federated learning framework in IoV.

- **Model Update Aggregation**: Finally, the new block consisting of the latest local model updates is recorded on the consortium blockchain, which is also responsible for notifying all the participants to download the latest block data. Each participant further computes the global model update by knowing the local model updates of the other participants.

5. Consensus algorithm design for the consortium blockchain

Furthermore, as shown in Fig. 2, we design an efficient consensus algorithm to defend against security attacks launched by a malicious entity, e.g., a vehicle or miner [40]. Each miner is instructed to download a uniform testing dataset and use it to judge whether the local model updates uploaded by a vehicle are qualified. According to the specific requirements for algorithm accuracy, each miner adopts predefined filtering strategies to filter malicious entities with failures or poisoning attacks. To identify and remedy the adverse effects of the malicious entities on the consortium blockchain, we analyze and filter the malicious entities using the consensus algorithm, i.e., preventing them from disturbing normal system operations. We take advantage of the flexibility of the consensus algorithm to defend against potential security attacks.

More specifically, a consensus algorithm named dBFT is used to carry out the consensus process among the miners. Only the qualified local model updates are aggregated to generate the latest global model. In this way, we remove low-quality local model updates and ultimately achieve reliable TFP.

Unlike the traditional consensus algorithms, such as the computation-intensive proof-of-work, communication-complex Byzantine fault tolerance, and unfair proof-of-stake, dBFT evolved from the practical Byzantine fault tolerance, and enables the flexible changing of miners without a fixed miner group. Compared with the delegated proof-of-stake algorithm, dBFT shows better performance for the finality of block data. Inspired by this, we employ dBFT as the consensus algorithm to establish an efficient and flexible consortium blockchain for the application of FL in TFP. We revise the consensus process to integrate model update verification into the consensus algorithm, thus defending against any potential security attack and ensuring reliable FL. More details about the consensus process are given as follows.

- ***Step 1: Initialization:*** A set of RSUs with powerful computation and communication capabilities are selected as miner candidates. A global trust authority, e.g., a government department, authenticates the identification of these miner candidates. Only legitimate candidates can be chosen as delegated miners, and the trusted authority receives their certificates for information encryption and digital signatures. The delegates not only generate or verify block data but also execute the model update verification tasks based on a given testing dataset. To prevent malicious delegates, the delegates must submit a deposit to a specific account, as required by the trusted authority. All the delegates are supervised by the other delegates. If a delegate executes malicious behaviors during the consensus process, the deposit will be confiscated and the delegate is also put on a blacklist and removed from the delegate groups [41]. Moreover, the elliptic curve digital signature algorithm and asymmetric cryptography for protocol initialization are applied in the consortium blockchain as a security guarantee.
- ***Step 2: Leader selection:*** Similar to delegated proof-of-stake, the delegates working as miners generate and verify the block data in dBFT (as shown in Fig. 1). A delegated miner (i.e., a delegate) is randomly chosen as the leader miner of the current consensus round from the delegates, and the rest of the delegates act as followers during the consensus process. We consider that there are N delegates with a maximal number of g malicious delegates in the consortium blockchain. To ensure blockchain security, we assume that $N > 3g + 1$ is satisfied in the consortium blockchain, and the leader miner changes after the current consensus round. Over N time slots, each delegate is randomly chosen as the leader miner and given a turn to perform block

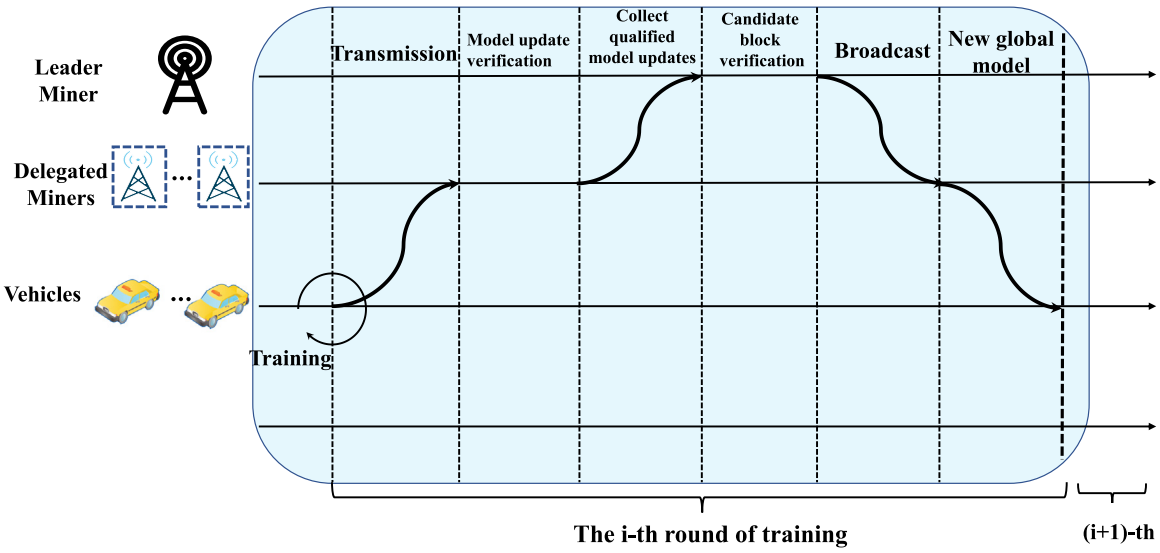

Fig. 2. Consensus process for blockchain-based federated learning.

generation, block broadcasting, data verification, and block management.

- ***Step 3: Local model training:*** To create an accurate and timely traffic flow model, the application of FL is proposed to train a global model by scheduling mobile vehicles to execute Algorithm 1. Vehicles collaboratively train an initialized global model released by a task publisher in the TFP system, and iteratively generate their own local model updates. The vehicles then upload these local model updates with their own digital signatures to the miner group (i.e., the delegate group) with the help of current vehicular communication technologies.
- ***Step 4: Model update verification***: After receiving the local model updates, the miners first verify the legality of the senders by authenticating their digital signatures and estimating the quality of the local model updates based on a testing dataset [42]. To achieve a reliable model update, the testing dataset provided by the task publisher can be a small and shared dataset with iid data samples for all the miners [35,41]. This dataset is treated as a reliable and public dataset to evaluate the quality of the local model updates. With this unique and universal dataset, only qualified local model updates with a higher training accuracy than a given threshold are accepted as transactions sent to the current leader miners. This threshold is determined by the task publisher according to application requirements. With the help of the testing dataset, low-quality local model updates are identified and removed from the model update aggregation. After a certain period of time, the leader miner collects qualified local model updates, generates a new block with its digital signature, and broadcasts this block to the other miners acting as followers. The followers receive and verify the block data (including the signature of the leader miner, transaction data, and signatures of all the transactions) from the leader miner.
- ***Step 5: Candidate block verification***: The process of candidate block verification includes the following steps:
- Preparation stage: The follower miners generate verification results for the block data and send the verification results with their digital signatures to each other for mutual confirmation. When miners receive at least $2g$ verification results for the block data, the preparation stage is complete.
- Commit stage: Each follower compares its own verification results with the results received from other followers and replies with an acknowledgment message that includes its digital signature to all the other miners to indicate whether it agrees with the verification results. If more than $2g + 1$ miners agree with the block data, the followers send the commit result with a digital signature to the leader miner to demonstrate that most of the miners agree on the truthfulness of the block data.
- Reply stage: After receiving the feedback from the different follower miners, the leader miner verifies the information and checks whether more than $2/3$ of the miners have reached the same conclusion about the block data. If so, the block data is recorded into the consortium blockchain.
- ***Step 6: Global model training***: In FL, all the participants download the new block data from the consortium blockchain, and calculate the weighted average of all the qualified local model updates as their new global models. The participants utilize the new global model as the initialized model for the next iteration of global model training. When the global model training is complete, the vehicles that continuously submitted qualified local model updates are noted and awarded with a certain size of monetary reward.

In the above consensus process, false or low-quality local model updates are not integrated into the global model training, which improves the FL performance and reduces poisoning attacks. Therefore, the proposed consortium blockchain with a dBFT-based consensus algorithm ensures the secure, reliable, and privacy-preserving characteristics of the FL-based TFP method for ITS.

6. Local differential privacy technique for privacy-preserving vehicular communications in TFP

Unlike the original differential privacy mechanism, the LDP mechanism focuses on protecting the privacy of participants during the data collection process. LDP is an effective countermeasure for privacy protection, and it is commonly applied in distributed machine learning with the help of local disturbances [15, 43]. Let x be the input vector, and a perturbed vector v^* for x is output through a randomized algorithm $\mathcal{M}$. Therefore, (ε, δ)-LDP can be defined as follows:

Definition 1 ((ε, δ)-*LDP [44]*)**.** A randomized algorithm $\mathcal{M} : \mathbb{X} \to \mathbb{V}$ with domain $\mathbb{X}$ and range $\mathbb{V} \subseteq \mathbb{X}$ satisfies (ε, δ)-LDP if and only if for any two inputs $x, x' \in \mathbb{X}$ and output $v^* \in \mathbb{V}$:

$$\Pr[\mathcal{M}(x) = v^*] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(x') = v^*] + \delta, \tag{12}$$

where $\Pr[\cdot]$ is the conditional probability density function that depends on $\mathcal{M}$, ε represents privacy budget that controls the trade-off between privacy and utility, and δ is a sufficiently small positive real number. It also implies a higher privacy budget ε means a lower privacy protection. The above definition can be changed to ε-LDP when $\delta = 0$.

According to Definition 1, randomized perturbations should be added to the data by the participants. To protect the privacy of the vehicles' location information during the TFP tasks, we add well-designed noise to disturb the location information by applying a Gaussian mechanism. We first define the Gaussian mechanism as follows:

Definition 2 (*Gaussian Mechanism [35]*)**.** Suppose a participant of FL wants to generate a function $f(x)$ of an input x subject to (ε, δ)-LDP. Thus, we have:

$$M(x) \triangleq f(x) + \mathcal{N}(0, \sigma^2 S^2). \tag{13}$$

It is assumed that $S = \| f(x) - f(x') \|_2 \leq \Delta_f$, (i.e., the sensitivity S of the function $f(x)$ is bounded by Δ_f), $\forall x, x'$, and then for any $\delta \in (0, 1)$, if and only if $\varepsilon = \frac{\Delta_f}{\sigma}\sqrt{2 \log \frac{1.25}{\delta}}$, Gaussian mechanism satisfies (ε, δ)-LDP.

In this paper, we assume that the latitude and longitude coordinates of the vehicle's location are (a, b). When the oVML of a vehicle needs to be updated, the vehicle may share its location information with other vehicles, which violates location privacy. Therefore, we apply the LDP technique to the above procedure to support privacy-preserving interactions among vehicles, as follows.

$$\Omega_{(a_i,b_i)|1 \leqslant i \leqslant k} = (a_i, b_i) + \mathcal{N}(0, \sigma^2 S^2), \tag{14}$$

where $\Omega_{(a_i,b_i)|1 \leqslant i \leqslant k}$ is the modified location information with the help of the disturbance, and $\mathcal{N}(0, \sigma^2 S^2)$ represents the Gaussian noise mechanism.

7. Experiments

In this section, we use the PySyft [45] framework to design an FL framework, and we use the consortium blockchain to build a blockchain verification platform. We use the Caltrans performance measurement system [46] dataset collected by California highways to verify the performance of the proposed framework. Because the Caltrans performance measurement system dataset contains traffic flow data collected in California during the first three months of 2013, we used the first two months of data for training and the last month of data for testing. We use the traffic flow data from the previous two months as input to predict the traffic flow of the following month. During the simulation, we set the number of vehicles $\mathcal{V} = 10$, the learning rate $\alpha = 0.001$, the training round $T = 500$, the mini-batch $m = 128$, and local training epochs $\mathcal{B} = 5$ [47,48]. Furthermore, Mean Absolute Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE) are adopted to indicate the prediction accuracy, which are calculated as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_p|, \tag{15}$$

Table 1
Performance comparison of MAE, MSE, and RMSE, For federated learning-based GRU model, LSTM model, SAE model, And SVM model.

Metrics	MAE	MSE	RMSE
Federated learning-based GRU Model	7.96	101.49	11.04
SAE model [2]	8.26	99.82	11.60
LSTM model [49]	8.28	107.16	11.45
SVM model [50]	8.68	115.52	13.24

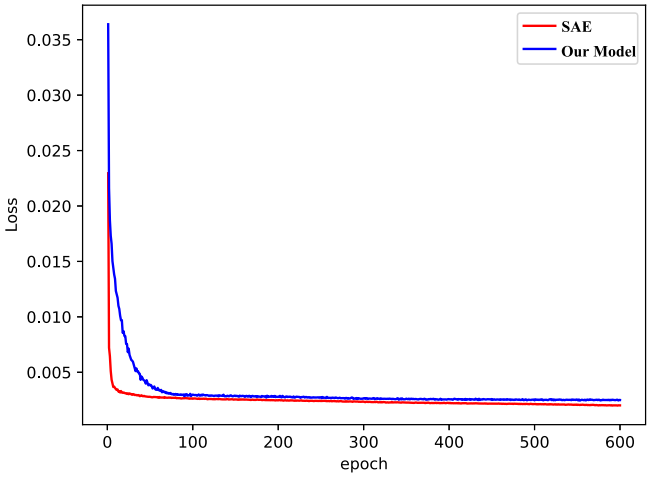

Fig. 3. Loss of SAE model and our model.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_p)^2, \tag{16}$$

$$\text{RMSE} = [\frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_p|)^2]^{\frac{1}{2}}, \tag{17}$$

where y_i indicates the observed traffic flow, and $\hat{y}_p$ indicates the predicted traffic flow.

7.1. System performance

As shown in Table 1, we compared the performance of the proposed model, i.e., the FL-based GRU model, with that of the stacked autoencoder (SAE), LSTM, and SVM models with the same simulation configuration. SAE, LSTM, and SVM models are advanced centralized TFP models, but the FL-based GRU model is an FL model. In this experiment, all models were evaluated on the PeMS dataset, and the experimental results are shown in Table 1. Without loss of generality, we estimated the traffic flow 5 min in the future to evaluate all models. According to the experimental results, the proposed model achieved state-of-the-art results. Specifically, the MAE of the FL-based GRU model is 10.04% lower than that of the SVM model (i.e., the worst-case model). The experimental results show that the proposed model not only accurately predicted traffic flow, but also protected the privacy of the vehicle data.

As shown in Fig. 3, we compared the loss of the SAE model with that of our model. We can explore the accuracy and convergence speed of the two models by comparing their losses. From the experimental results, we find that the convergence rate of the SAE model is faster than the convergence rate of the proposed model. The reason for this is that the SAE model uses a centralized learning paradigm, and thus it does not need to complete a model aggregation step, whereas our model needs to complete

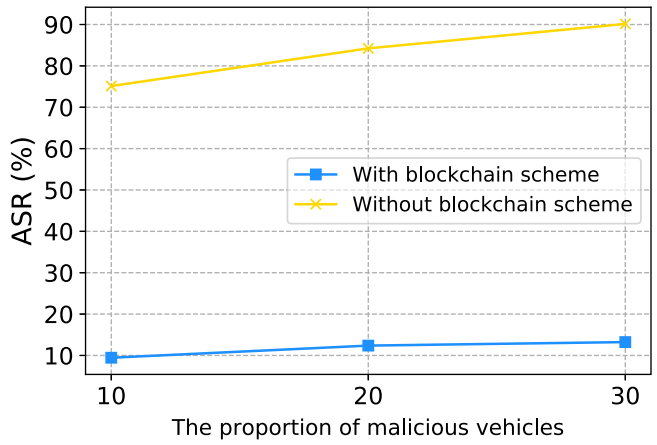

Fig. 4. Compare the impact of different proportions of malicious vehicles on the successful attack rate.

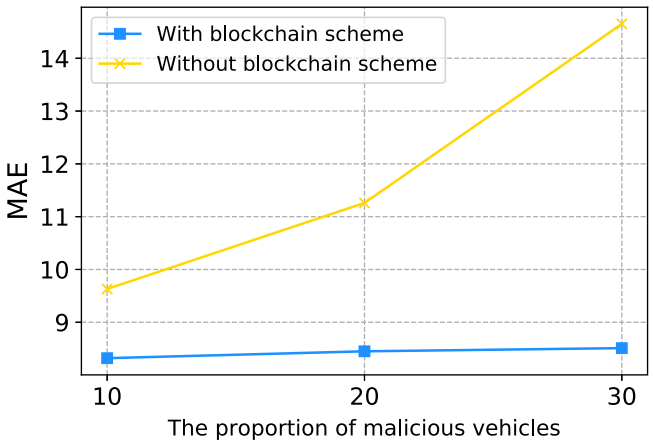

Fig. 5. Compare the impact of different proportions of malicious vehicles on model accuracy.

this step. In short, model aggregation reduces the convergence speed of the model. Second, the final convergence values for the two models are approximately the same. Such results show that the performance of the proposed model is comparable to the centralized learning model.

7.2. Performance of secure federated model training

In this section, we explore the defense capabilities of the proposed framework against poisoning attacks. First, we give the definition of attack success rate (ASR) as follows: a malicious attacker launches a malicious attack and successfully destroys the performance of the framework or steals privacy. Second, we assume that attackers initiate data or poisoning attacks on the proposed framework and that the proportion of attackers is $p \in \{10\%, 20\%, 30\%\}$. As shown in Fig. 4, the proposed framework is robust to data and poisoning attacks. The reason for this is that blockchain technology can verify the quality of each vehicle's model update, and it can filter out malicious model updates. Specifically, as the number of attackers increases, a framework without a blockchain solution is less able to resist malicious attacks. The above experimental results show that the proposed framework can resist poisoning attacks.

In Fig. 5, we can see that the model using the blockchain solution is robust against malicious vehicle attacks. This is because the accuracy of the model using the blockchain solution will not decrease as the number of malicious vehicles increases. On the contrary, the performance of models that do not use the blockchain solution greatly decreases as the proportion of malicious vehicles increases.

7.3. Performance of secure federated data sharing

In this section, we explore the privacy protection capabilities of local differential privacy technique for location information sharing in the federated learning framework. As mentioned in the previous section, we still use ASR to indicate the ability to protect privacy. First, we set privacy parameter $\delta \in \{e^{-1}, e^{-3}, e^{-5}\}$. Note that the smaller the privacy parameter, the more noise is added, i.e., the stronger the privacy protection capability. As shown in Fig. 6, there is always a higher possibility of privacy leakage in the framework without the LDP scheme. It is willing that the attacker can steal the location information of the vehicle through physical attacks in the process of vehicle location sharing. On the contrary, the probability of successfully attacking the framework with LDP scheme is very low. Furthermore, the smaller the privacy parameter, the more secure the framework of the LDP scheme.

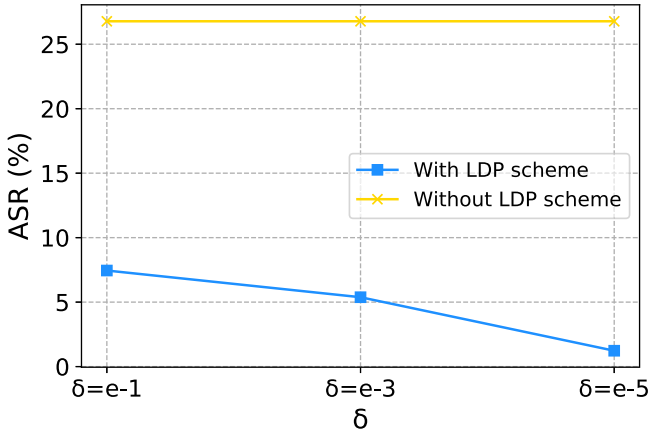

Fig. 6. Compare the impact of different privacy parameters on the success rate of attacks.

7.4. Performance of secure block verification

To evaluate the security performance of the dBFT consensus algorithm in the proposed system, we model the security performance of a miner group (i.e., delegate group) as a random sampling problem in terms of two possible outcomes: well-behaved delegates and malicious delegates [51]. Similar to [51], the security probability of a delegate group, that a block data is verified correctly and truly by the delegates when the number of malicious delegates g is less than $(N-1)/3$, is expressed as $PM = \sum_{z=0}^{g} \binom{N}{z} p_m^z (1-p_m)^{N-z}$. Here, p_m is the probability of becoming a malicious delegate, which takes a value between 0.1 and 0.3. As shown in Fig. 7, the probability of a consensus process using dBFT decreases with the increasing probability of becoming a malicious delegate (i.e., p_m). We can observe that the security probability increases when the size of the delegate group grows with any level of probability that malicious delegates exist. The reason for this is that a larger size of delegate group results in a more secure consensus process, due to the increased number of well-behaved delegates participating in block verification. Therefore, the proposed dBFT consensus algorithm can provide reliable and secure block verification when enough delegates participate in the consensus.

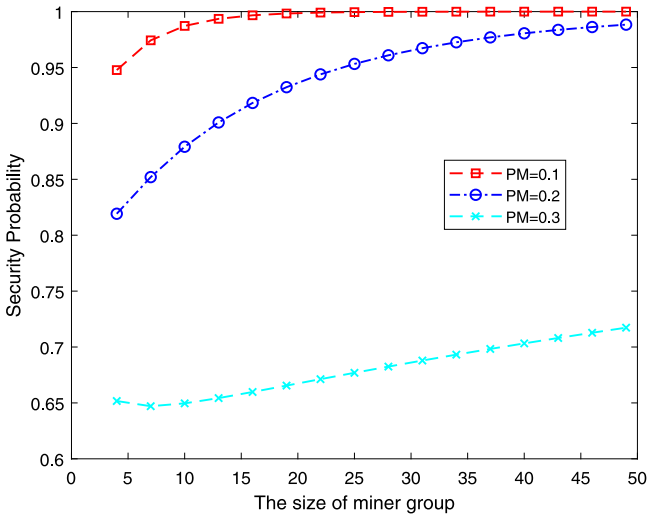

Fig. 7. Security probability with respect to different compromised probability of miners.

8. Conclusions

In this study, we designed a blockchain-based secure FL framework for urban traffic flow management. Specifically, we introduced an FL framework to protect the privacy of the vehicle data. We also applied the GRU model to the FL framework to achieve accurate TFP. In order to prevent malicious attackers from destroying the urban traffic flow management system, we leveraged the blockchain to implement a decentralized FL framework that could defend against poisoning attacks. Finally, we used local differential privacy technology to protect privacy in vehicle location sharing. The described framework may be useful to traffic management departments and traffic police for managing traffic.

In the future, we will improve the accuracy and efficiency of the GRU model. This is because the communication overhead for FL is expensive, which affects the performance of the framework. We may work towards designing efficient algorithms and frameworks for communication.

CRediT authorship contribution statement

Yuanhang Qi: Conceptualization, Writing - original draft, Methodology. **M. Shamim Hossain:** Supervision, Methodology. **Jiangtian Nie:** Writing - review & editing, Validation. **Xuandi Li:** Experimental results preparation, Software.

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.

Acknowledgments

The authors extend their appreciation to the Researchers Supporting Project number (RSP-2020/32), King Saud University, Riyadh, Saudi Arabia for funding this work. This research was also funded by Key Project in Higher Education of Guangdong Province, China under grant No. 2020ZDZX3030 and the Young Innovation Talents Project in Higher Education of Guangdong Province, China under Grant No. 2018KQNCX333.

References

[1] Z. Cao, S. Jiang, J. Zhang, H. Guo, A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion, IEEE Trans. Intell. Transp. Syst. 18 (2016) 1958–1973.
[2] Y. Lv, Y. Duan, W. Kang, Z. Li, F.Y. Wang, Traffic flow prediction with big data: a deep learning approach, IEEE Trans. Intell. Transp. Syst. 16 (2014) 865–873.
[3] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. Pham, S.K. Padannayil, K. Simran, A visualized botnet detection system based deep learning for the internet of things networks of smart cities, IEEE Trans. Ind. Appl. 56 (2020) 4436–4456.
[4] X. Yang, et al., Deep relative attributes, IEEE Trans. Multimed. 18 (2016) 1832–1842.
[5] M.S. Hossain, M. Al-Hammadi, G. Muhammad, Automatic fruit classification using deep learning for industrial applications, IEEE Trans. Ind. Inf. 15 (2019) 1027–1034.
[6] S. Messaoud, et al., Deep federated q-learning-based network slicing for industrial iot, IEEE Trans. Ind. Inf. (2020) 1, http://dx.doi.org/10.1109/TII.2020.3032165.
[7] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, K.S. Ng, Towards fair and privacy-preserving federated deep models, IEEE Trans. Parallel Distrib. Syst. 31 (2020) 2524–2541.
[8] M.S. Hossain, G. Muhammad, W. Abdul, B. Song, B. Gupta, Cloud-assisted secure video transmission and sharing framework for smart cities, Future Gener. Comput. Syst. 83 (2018) 596–606.
[9] A.A.A. El-Latif, B. Abd-El-Atty, M.S. Hossain, S. Elmougy, A. Ghoneim, Secure quantum steganography protocol for fog cloud internet of things, IEEE Access 6 (2018) 10332–10340.
[10] X. Huang, D. Ye, R. Yu, L. Shu, Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design, IEEE/CAA J. Autom. Sin. 7 (2020) 426–441.
[11] J. Kang, Z. Xiong, D. Niyato, S. Xie, J. Zhang, Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory, IEEE Internet Things J. 6 (2019) 10700–10714.
[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.
[13] J. Guo, W. Huang, B.M. Williams, Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification, Transp. Res. C 43 (2014) 50–64.
[14] R. Yao, W. Zhang, L. Zhang, Hybrid methods for short-term traffic flow prediction based on arima-garch model and wavelet neural network, J. Transp. Eng. Part A Syst. 146 (2020) 04020086.
[15] A. Azab, M. Alazab, M. Aiash, Machine learning based botnet identification traffic, in: 2016 IEEE Trustcom/BigDataSE/ISPA, 2016, pp. 1788–1794.
[16] M. Alazab, S. Venkatraman, Detecting malicious behaviour using supervised learning algorithms of the function calls, Int. J. Electron. Secur. Digit. Forensics 5 (2013) 90–109.
[17] W. Wei, H. Wu, H. Ma, An autoencoder and lstm-based traffic flow prediction method, Sensors 19 (2946) (2019).
[18] F. Zhao, G. Zeng, K. Lu, Enlstm-wpeo: Short-term traffic flow prediction by ensemble lstm, nnct weight integration, and population extremal optimization, IEEE Trans. Veh. Technol. 69 (2020) 101–113.
[19] S. Zhao, Q. Zhao, Y. Bai, S. Li, A traffic flow prediction method based on road crossing vector coding and a bidirectional recursive neural network, Electronics 8 (1006) (2019).
[20] J. Li, H. Li, G. Cui, Y. Kang, Y.Z. Yang Hu, Gacnet: A generative adversarial capsule network for regional epitaxial traffic flow prediction, Comput. Mater. Contin. 64 (2020) 925–940.
[21] Z. Cao, H. Guo, J. Zhang, D. Niyato, U. Fastenrath, Finding the shortest path in stochastic vehicle routing: a cardinality minimization approach, IEEE Trans. Intell. Transp. Syst. 17 (2016) 1688–1702.
[22] Y. Zhang, et al., Edge intelligence in the cognitive internet of things: Improving sensitivity and interactivity, IEEE Netw. 33 (2019) 58–64.
[23] Y. Qian, M. Chen, J. Chen, M.S. Hossain, A. Alamri, Secure enforcement in cognitive internet of vehicles, IEEE Internet Things J. 5 (2018) 1242–1250.
[24] Y. Qian, et al., Blockchain-based privacy-aware content caching in cognitive internet of vehicles, IEEE Netw. 34 (2020) 46–51.
[25] J. Konecný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, CoRR abs/1610.05492.
[26] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, IEEE Trans. Ind. Inf. 16 (2020) 6532–6542.
[27] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, L. Liang, Self-balancing federated learning with global imbalanced data in mobile systems, IEEE Trans. Parallel Distrib. Syst. 32 (2021) 59–71.

[28] K. Lin, et al., Green video transmission in the mobile cloud networks, IEEE Trans. Circuits Syst. Video Technol. 27 (2017) 159–169.
[29] M. Alhamid, et al., Towards context-sensitive collaborative media recommender system, Multimedia Tools Appl. 74 (2015) 11399–11428.
[30] C. Fang, Y. Guo, N. Wang, A. Ju, Highly efficient federated learning with strong privacy preservation in cloud computing, Comput. Secur. 96 (2020) 101889.
[31] W. Liu, L. Chen, Y. Chen, W. Zhang, Accelerating federated learning via momentum gradient descent, IEEE Trans. Parallel Distrib. Syst. 31 (2020) 1754–1766.
[32] Z. Li, J. Liu, J. Hao, H. Wang, M. Xian, CrowdSFL: A secure crowd computing framework based on blockchain and federated learning, Electronics 9 (773) (2020).
[33] Y. Qu, L. Gao, T.H. Luan, Y. Xiang, S. Yu, B. Li, G. Zheng, Decentralized privacy using blockchain-enabled federated learning in fog computing, IEEE Internet Things J. 7 (2020) 5171–5183.
[34] P.C.M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, M. Atiquzzaman, A trustworthy privacy preserving framework for machine learning in industrial iot systems, IEEE Trans. Ind. Inf. 16 (2020) 6092–6102.
[35] Y. Liu, J. Peng, J. Kang, A.M. Iliyasu, D. Niyato, A.A.A. El-Latif, A secure federated learning framework for 5g networks, IEEE Wirel. Commun. 27 (2020) 24–31.
[36] M.S. Hossain, G. Muhammad, A deep-tree-model-based radio resource distribution for 5g networks, IEEE Wirel. Commun. 27 (2020) 62–67.
[37] L. Zhang, G. Ding, Q. Wu, Y. Zou, Z. Han, J. Wang, Byzantine attack and defense in cognitive radio networks: A survey, IEEE Commun. Surv. Tutor. 17 (2015) 1342–1363.
[38] C. Fung, C.J.M. Yoon, I. Beschastnikh, Mitigating sybils in federated learning poisoning, 2018, arXiv:1808.04866.
[39] J. Steinhardt, P.W.W. Koh, P.S. Liang, Certified defenses for data poisoning attacks, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, pp. 3517–3529.
[40] M. Tang, M. Alazab, Y. Luo, Big data for cybersecurity: Vulnerability disclosure trends and dependencies, IEEE Trans. Big Data 5 (2019) 317–329.
[41] J. Kang, Z. Xiong, C. Jiang, Y. Liu, S. Guo, Y. Zhang, D. Niyato, C. Leung, C. Miao, Scalable and communication-efficient decentralized federated edge learning with multi-blockchain framework, 2020, arXiv preprint arXiv:2008.04743.
[42] C. Benzaid, K. Lounis, A. Al-Nemrat, N. Badache, M. Alazab, Fast authentication in wireless sensor networks, Future Gener. Comput. Syst. 55 (2016) 362–375.
[43] Y. Liu, J. Peng, J.J. Yu, Y. Wu, Ppgan: Privacy-preserving generative adversarial network, in: 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), 2019.
[44] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (2014) 211–407.
[45] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, J. Passerat-Palmbach, A generic framework for privacy preserving deep learning, 2018, arXiv preprint arXiv:1811.04017.
[46] C. Chao, Freeway performance measurement system (pems), 2003.
[47] Y. Liu, J.J.Q. Yu, J. Kang, D. Niyato, S. Zhang, Privacy-preserving traffic flow prediction: A federated learning approach, IEEE Internet Things J. 7 (2020) 7751–7763.
[48] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, M.S. Hossain, Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach, IEEE Internet Things J. (2020).
[49] X. Ma, Z. Tao, Y. Wang, H. Yu, Y. Wang, Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transp. Res. C 54 (2015) 187–197.
[50] M.A. Mohandes, T.O. Halawani, S. Rehman, A.A. Hussain, Support vector machines for wind speed prediction, Renew. Energy 29 (2004) 939–947.
[51] K. Lei, M. Du, J. Huang, T. Jin, Groupchain: Towards a scalable public blockchain in fog computing of iot services computing, IEEE Trans. Serv. Comput. 13 (2020) 252–262, http://dx.doi.org/10.1109/TSC.2019.2949801.

Yuanhang Qi received his Ph.D. degree from Guangdong University of Technology, Guangzhou, China, in 2018. He is currently a lecturer at University of Electronic Science and Technology of China, Zhongshan Institute. His current research interests include modeling and optimization of complex system, intelligent algorithm and intelligent optimization.

M. Shamim Hossain is a Professor at the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is also an adjunct professor at the School of Electrical Engineering and Computer Science, University of Ottawa, Canada. He received his Ph.D. in Electrical and Computer Engineering from the University of Ottawa, Canada. His research interests include cloud networking, smart environment (smart city, smart health), AI, deep learning, edge computing, Internet of Things (IoT), multimedia for health care, and multimedia big data. He has authored and coauthored more than 300 publications including refereed journals, conference papers, books, and book chapters. Recently, he co-edited a book on "Connected Health in Smart Cities", published by Springer. He has served as cochair, general chair, workshop chair, publication chair, and TPC for over 12 IEEE and ACM conferences and workshops. Currently, he is the cochair of the 3rd IEEE ICME workshop on Multimedia Services and Tools for smart-health (MUST-SH 2020). He is a recipient of a number of awards, including the Best Conference Paper Award and the 2016 ACM Transactions on Multimedia Computing, Communications and Applications (TOMM) Nicolas D. Georganas Best Paper Award. He is on the editorial board of the IEEE Transactions on Multimedia, IEEE Multimedia, IEEE Network, IEEE Wireless Communications, IEEE Access, Journal of Network and Computer Applications (Elsevier), and International Journal of Multimedia Tools and Applications (Springer). He also presently serves as a lead guest editor of and Multimedia systems Journal. He serves/served as a guest editor of IEEE Communications Magazine, IEEE Network, ACM Transactions on Internet Technology, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), IEEE Transactions on Information Technology in Biomedicine (currently JBHI), IEEE Transactions on Cloud Computing, Multimedia Systems, International Journal of Multimedia Tools and Applications (Springer), Cluster Computing (Springer), Future Generation Computer Systems (Elsevier). He is a senior member of both the IEEE, and ACM. He is an IEEE ComSoc Distinguished Lecturer (DL).

Jiangtian Nie received her B.Eng degree with honors in Electronics and Information Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2016. She is currently working towards the Ph.D. degree with ERI@N in the Interdisciplinary Graduate School, Nanyang Technological University, Singapore. Her research interests include incentive mechanism design in crowdsensing and game theory.

Xuandi Li received the B.Eng. degree from Beijing Institute of Technology, Zhuhai in 2013. She is currently a research engineer in Nanyang Technological University, Singapore. Her research interests mainly focus on blockchain, federated learning, and Internet of Things.