Received February 16, 2020, accepted March 4, 2020, date of publication March 12, 2020, date of current version April 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980298

TrafficChain: A Blockchain-Based Secure and Privacy-Preserving Traffic Map

QIANLONG WANG[1], TIANXI JI[1], YIFAN GUO[1], (Student Member, IEEE), LIXING YU[1], (Student Member, IEEE), XUHUI CHEN[2], AND PAN LI[1], (Member, IEEE)

[1]Department Electrical, Computer, and Systems Engineering, Case Western Reserve University, Cleveland, OH 44106, USA
[2]College of Aeronautics and Engineering, Kent State University, Kent, OH 44240, USA

Corresponding author: Pan Li (pxl288@case.edu)

ABSTRACT Intelligent Connected Vehicles (ICVs) can provide smart, safe, and efficient transportation services and have attracted intensive attention recently. Obtaining timely and accurate traffic information is one of the most important problems in transportation systems, which would allow people to select fast routes and avoid congestions, thus saving their travel time on the road. Currently, the most popular ways to obtain traffic information is to inquire navigation agents, e.g., Apple map, and Google map. However, these navigation agents are essentially centralized systems, which are vulnerable to service congestions, a single point of failure, and attacks. Furthermore, users' privacy gets compromised as the agents can know their home and work addresses and hence their identities, track them in real-time, etc. In this paper, we propose TrafficChain, a secure and privacy-preserving decentralized traffic information collection system on the blockchain, by taking advantage of fog/edge computing infrastructure. In particular, we employ a two-layer blockchain architecture in TrafficChain to improve system efficiency, design a privacy-preserving scheme to protect users' identities and travel traces, and devise LSTM based deep learning mechanisms that can defend against Byzantine attacks and Sybil attacks in our system. Furthermore, an incentive mechanism is designed to motivate users to participate in the system. Simulation results show that TrafficChain works very efficiently and is resilient to both Byzantine attacks and Sybil attacks.

INDEX TERMS Blockchain, intelligent connected vehicles (ICVs), fog/edge computing, Byzantine attacks, Sybil attacks, LSTM.

I. INTRODUCTION

Intelligent Connected Vehicles (ICVs) aim to provide smart, safe, and efficient transportation services by exploiting multiple modern technologies, including communications, computing, data mining, deep learning. Intel has predicted that vehicles will produce 4,000 GB of data every day by 2025 [1]. Properly utilizing and mining the data provided by smart vehicles, ICVs would be able to support various services such as dynamic routing [2], traffic incident detection [3], autonomous driving [4].

Despite having been studied for years, the current transportation infrastructure and systems are still simple and quite traditional. Many problems are still open and extensive efforts are required to fulfill the mission of ICVs. In particular, real-time traffic status is crucial information for vehicles to plan their routes to avoid congestion or road incidents, like accidents and road closure, and can greatly save people's travel time. Currently, people usually use popular navigation agents like Apple map and Google map to help plan travel routes. A driver can send his/her current location and the destination to the service provider through smartphones or navigation devices to acquire navigation guide. By collecting and mining the real-time travel data from a large number of users' smartphones, the traffic map service providers like Apple and Google are able to provide the traffic status on the roads. However, such navigation agents raise several major concerns. First, current navigation systems are centralized and vulnerable to congestion and a single point of failure. Second, current navigation systems like Apple map or Google map collect real-time information from users and has already compromised users' privacy. For example, they can know users' home and work addresses and hence their identities, track them in real-time, etc. Third, users have no control over

The associate editor coordinating the review of this manuscript and approving it for publication was Junhui Zhao.

the security of current navigation systems. If compromised, the navigation systems can not only deliberately return malicious route plans to users, but also track users and cause serious security concerns.

The emergence of blockchain provides a promising solution to the aforementioned issues. Blockchain is a novel data storage technology, which is built upon decentralized peer-to-peer networks [5]. On a blockchain, each participant is allowed to view the content in all blocks. When a new block is created, the transactions or information that belongs to the corresponding time slot is stored in that block, and all the participant can verify the content in the block. With different mechanisms such as Proof of Work (PoW) and smart contract, blockchain has successfully provided security and privacy for many applications, particularly for cryptocurrencies. However, how to exploit blockchain technology to design secure and privacy-aware applications for ICVs is still a challenging problem due to different security and privacy requirements in such systems. In the literature, there have been a few works on blockchain based ICV applications. For example, Li *et al.* [6] propose a privacy-preserving incentive announcement network called CreditCoin, which motivates the vehicles to share the data including the traffic status. Michelin *et al.* [7] develop a blockchain based smart transportation architecture named SpeedyChain, that allow smart vehicles to share their data. Hîrtan *et al.* [8] describe an architecture of the car navigation system in which the personal data is protected. Nevertheless, most previous works introduce Road Side Infrastructure (RSI), security managers, or other third-party authorities to authorize vehicles or verify communication messages, which essentially share the same concerns with the current centralized systems.

In this paper, we exploit blockchain to devise a decentralized real-time city-wide traffic information collection system called TrafficChain, which is both secure against malicious attacks on the system and able to protect vehicles' private information. In particular, TrafficChain is featured with a two-layer blockchain architecture as shown in Fig. 1, which includes local chains, one for each road segment, and a global chain. The computing nodes (i.e., "miners") in the system can be either computing nodes owned by individuals (stationary like computers at home and vehicles parked on the streets, or mobile like moving vehicles) that are willing to participate in this system, or edge (or fog) routers that are deployed by "edge service providers (ESP)" providing computing, communication, and storage services for the city, e.g., at the cellular base stations, on the streets, or on top of tall buildings. For each local chain, the "local miners" are the nearby miners who would like to participate. Each block on a local chain is broadcasted to the local miners that are on this particular local chain and all the miners on the global chain. For the global chain, any miner in the city that has enough computing capabilities can be a "global miner". Each block on the global chain contains the aggregated report on the traffic status of each of the road segments and is broadcasted to all the global miners. Whenever a vehicle needs to find a route to some

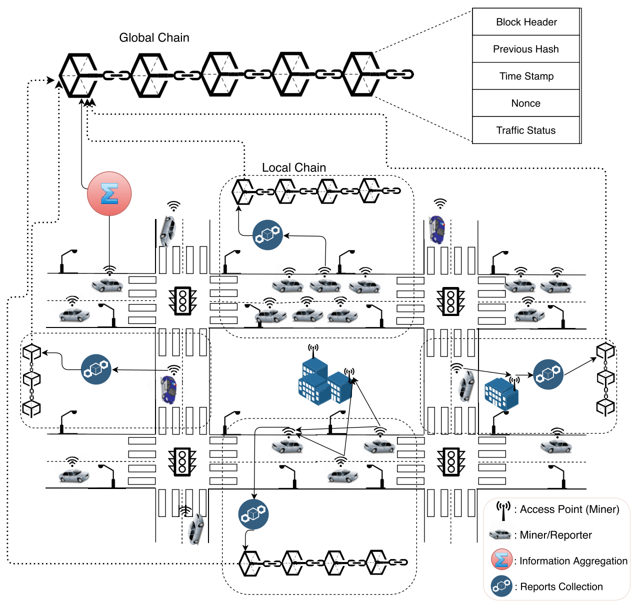

FIGURE 1. A blockchain based secure and privacy-preserving ICV system.

location, it can retrieve the necessary traffic status from the global chain, for example, by inquiring the nearby global miners. In such an architecture, the traffic status collection can be more efficient and the communication overhead can be significantly reduced due to avoiding letting every vehicle broadcast its reports to the whole network.

In addition to a new architecture design, we develop effective algorithms to protect the security of TrafficChain and the privacy of vehicles in the system. We consider Byzantine and Sybil attacks, which are the most popular and challenging attacks to deal with in blockchain based systems. By exploiting a deep learning model called Long Short-Term Memory (LSTM), we design novel LSTM based schemes that can not only defend against Byzantine and Sybil attacks but also make predictions on the forthcoming traffic status in a city. Besides, the SHA256 hash function and Elliptic Curve Digital Signature Algorithm (ECDSA) are employed for generating addresses for each user, where they can generate a large number of addresses for broadcasting reports in different time periods. Thus, the privacy of the participants can be protected.

Our main contributions in this paper are summarized as follows:

- We propose TrafficChain, a blockchain based decentralized real-time traffic information collection system. It has a novel two-layer blockchain architecture that can reduce the network communication overhead and enhance the block update speed, making the system more efficient.
- We introduce the Byzantine attack and Sybil attack on TrafficChain and propose novel LSTM based methods to defend against them. Furthermore, the proposed LSTM based schemes are able to predict the forthcoming traffic status in a city.

- We employ SHA256 and ECDSA to protect users' privacy. With digital signatures and verification schemes, an incentive mechanism is further designed to motivate users to report traffic status in TrafficChain.
- We implement TrafficChain on Ethereum to demonstrate its efficiency and resilience to both Byzantine and Sybil attacks.

The remainder of this paper is organized as follows. We introduce the related work in Section II and formulate our problem including the threat model in Section III, respectively. In Section IV, we detail the design of TrafficChain. We conduct simulations to validate TrafficChain's efficiency and resilience to Byzantine and Sybil attacks in Section V. Finally, we conclude the paper in Section VI.

II. RELATED WORK

A big chunk of work exists addressing traffic information collection or prediction in transportation systems. Jabari and Liu [9] propose a stochastic traffic flow model for estimating traffic flow. He and Liu [10] propose a prediction-correction model to describe the traffic equilibration process after an unexpected network disruption. Wang *et al.* [11] present a NeverStop system, which utilizes genetic algorithms and fuzzy control methods to control the traffic lights at the intersection automatically. Gisdakis *et al.* [12] leverage state-of-the-art cryptographic schemes and readily available telecommunication infrastructure and present a comprehensive solution to smartphone-based traffic estimation that is proven to be secure and privacy-preserving. Brown *et al.* [13] introduce Haze, a system that collects traffic statistics from user reports while protecting the users' privacy. Zhu *et al.* [14] propose a secure and privacy-preserving traffic flow analysis scheme for ICVs, called PTFA, where the traffic information is obtained and aggregated by a traffic regional center through vehicular ad hoc networks (VANETs). Lin *et al.* [15] identify security and privacy requirements in VANET communications and propose a group signature and identity (ID)-based signature techniques. Rabieh *et al.* [16] propose privacy-preserving route reporting schemes for traffic management in both infrastructures supported and self-organizing VANETs. Similarly, Zhang *et al.* [17] propose a privacy-preserving route reporting scheme that only an authenticated vehicle can use the route reporting service provided by the traffic management center. Note that most previous systems assume a trusted central management server, which may not always exist and can be under attacks. A few recent works have employed the blockchain technology to build decentralized ICVs. Rajbhandari [18] examine various blockchain applications in ICVs, and demonstrate that blockchain is applicable and beneficial to the current transportation system. Saranti *et al.* [19] depict a future of transportation system that combines autonomous vehicle and blockchain. Yuan and Wang [20] propose a blockchain based transportation system called La'zooz, which employs a consensus algorithm called proof-of-movement, to generate tokens for ridesharing and other transportation services. Rivera *et al.* [21] conduct a review for existing research on how to utilize digital identity on the blockchain for ICVs. Pedrosa and Pau [22] propose an Ethereum based system to support energy recharges for autonomous electric vehicles. Although these works propose various blockchain based ICV applications, security and privacy in the system are not studied.

Some works address user privacy in ICV applications built on blockchain. Li *et al.* [6] propose CreditCoin, a privacy-preserving incentive based announcement network on blockchain where users are able to send announcements anonymously in the non-fully trusted environment. Michelin *et al.* [7] propose a blockchain based smart transportation architecture named SpeedyChain, which ensures reliable Vehicle-to-Infrastructure communication and maintains vehicle privacy by employing periodically changeable keys. Singh and Kim [23] propose intelligent vehicles (IV) trust point (IVTP) by using blockchain, aiming at protecting the privacy and security in the communications among vehicles, where every message is signed by the private key of the user. Although a number of works attempt to protect the privacy in ICVs, most of them only focus on the anonymity of the broadcasting messages. Due to the exposure of driving routes for each user, attackers would be able to infer users' identities based on their route information. In our work, we consider the users' driving routes as private information and propose our privacy-preserving method.

Besides, several works are concerned about security issues in ICVs. Su *et al.* [24] propose an energy blockchain for secure electric vehicles (EV) charging in a smart community, which is to optimally schedule the charging behaviors of EVs with distinct energy consumption preferences. They propose a reputation based Byzantine fault tolerance consensus algorithm. Huang *et al.* [25] propose a secure decentralized charging pile management system on the blockchain, called lightning network and smart contract (LNSC), which can resist impersonating attack caused by key leaking. Sharma *et al.* [26] propose a secure and reliable vehicle network architecture based on blockchain, which considers the DoS attack, like jamming on the cloud. Singh and Kim [27] propose a trusted environment based intelligent vehicle framework, where the blockchain technology provides the trust environment between the vehicles with the based on proof of driving. Lei *et al.* [28] focus on a critical technique for network security, secure key management scheme, and develop a blockchain based secure key management framework in vehicular communication systems, which can reduce the key transfer time during vehicles handover. Chen *et al.* [29] target on the Byzantine attacks in a blockchain based distributed systems and propose an l-nearest method to mitigate the attacks. Xie *et al.* [30] propose three aggregation rules to resist Byzantine attacks in a distributed system, which are respectively based on geometric median, marginal median, and beyond median. Otte *et al.* [31] propose TrustChain,

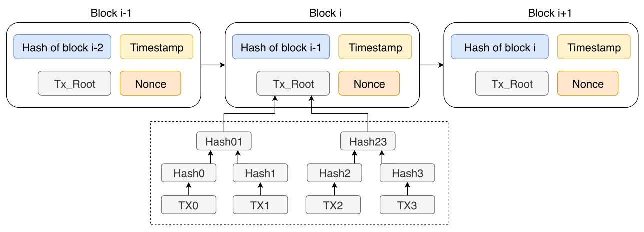

FIGURE 2. **The typical blockchain structure.**

a Sybil-resistant scalable blockchain, which can determine the trustworthiness of users in order to resist Sybil attacks in a distributed system. Although cyber-physical attacks like Byzantine and Sybil attacks have been widely studied, many of the methods have high computational complexity and may potentially leak sensitive or private information. Besides, these kinds of attacks have not been fully studied in ICVs. In our TrafficChain, we propose a novel deep learning based secure aggregation scheme that is resilient to both Byzantine and Sybil attacks.

III. PROBLEM FORMULATION

In this section, we first introduce some preliminaries about blockchain. Then, we model the decentralized traffic status collection problem, and describe the corresponding threat models in the system.

A. BLOCKCHAIN

Blockchain is a distributed, decentralized, public ledger. Originally devised for financial applications such as cryptocurrencies, the technology has now found use in many other applications such as smart health, smart business, smart city. One of the main features of blockchain is that it provides a decentralized solution to various traditional systems, where all the system information, e.g., transactions in digital currency systems, is stored in blocks and shared among all the participants. With communications through a peer-to-peer network, blockchain is capable of eliminating the need of an authorized third-party, e.g., bank, and hence has no concern about a single point of failure. Furthermore, with the design of different consensus protocols, e.g. proof of work (PoW) and proof of stake (PoS), blockchain is resilient to dishonest participants.

As shown in Fig. 2, blocks on a blockchain are linked chronologically, each of which contains four parts, i.e., the previous block's hash code, time stamp, transaction records, and a nonce. Among them, transaction records are represented by a Merkle tree. The nonce needs to be found by the miners so that the current block's hash fulfills certain system requirement. The miner who finds this nonce first, through PoW, wins the authority to write the new transactions stored in its memory pool into this current block and append it to the chain.

B. DECENTRALIZED TRAFFIC STATUS GENERATION

We define city traffic status as the passing time cost for each road segment. Denote the traffic status in the time slot i by $\mathbf{t}_i = [t_i^1, t_i^2, \ldots, t_i^N]$, where N is the total number of road segments, and t_i^j is the passing time cost for the jth road segment in the time slot i. Suppose that in the time slot i there are h_i^j reports about t_i^j, which are denoted by $\mathbf{t}_i^j = \{t_i^j(k), 1 \leq k \leq h_i^j\}$. To obtain the estimated passing time cost t_i, we design an aggregation function which is represented by $A(\cdot)$. Thus, we have

$$t_i^j = A(t_i^j(1), \ldots, t_i^j(h_i^j)), \quad j \in [1, N]. \tag{1}$$

A naive aggregation method is to calculate the average of all reports, which we consider as a benchmark, i.e.,

$$\bar{t}_i^j = \frac{1}{h_i^j} \sum_{k=1}^{h_i^j} t_i^j(k), \quad j \in [1, N]. \tag{2}$$

Note that the benchmark may be ineffective, particularly when there are malicious reporters in the system.

C. THREAT MODEL

We consider the following privacy and security threats in the system.

1) PRIVACY

The privacy in the system refers to two types of sensitive information: each user's identity and driving route. A user's identity includes both the driver's identity and the vehicle's identity, both of which need to be protected. Each user's driving route includes the road segments that the user passes through. It is very sensitive information since attackers can infer the user's identity and many details about this user's daily life (such as work location, home address, frequently visited places), and also track the user. Therefore, it is critical to protect each user's privacy in the system.

2) SECURITY

We are primarily concerned with Byzantine attacks and Sybil attacks in the system, which are described below.

a: BYZANTINE ATTACKS

Byzantine attackers report bogus reports in order to disrupt the normal report aggregation process in the system. For example, a Byzantine attacker can report a very high passing time for a certain road segment, which may lead the aggregated passing time to be much higher than it really is, thus forcing other users to use other roads while letting itself go through the road very fast. On the other hand, a Byzantine attacker can report a very low passing time for a road segment, which may mislead other users to use the road segment that is actually already crowded, thus exaggerating the traffic congestion.

b: SYBIL ATTACKS

Sybil attackers can create many different identities in the system in order to disrupt normal system operations. In particular, a Sybil traffic data reporter can use different identities to submit a number of bogus reports for the same road segment to improve the chance that the aggregated passing time for that road segment does get deviated from its true value. Besides, a Sybil miner can take advantage of many different identities to obtain more opportunities of writing blocks on the blockchain.

IV. TrafficChain: REAL-TIME TRAFFIC STATUS COLLECTION ON BLOCKCHAIN

A. SYSTEM OVERVIEW

In this section, we introduce the proposed real-time traffic status collection system on blockchain, i.e., TrafficChain, in detail. Different from most traditional ICV applications, TrafficChain has a decentralized architecture, which meanwhile is secure and privacy-preserving. The objective of TrafficChain is to securely collect and store the real-time traffic status in a city on the blockchain, which includes the estimated passing time cost for each road segment in the city and can be retrieved by each service demanding vehicle. The computing nodes (i.e., "miners") in the system can be either computing nodes owned by individuals (stationary like computers at home and vehicles parked on the streets, or mobile like moving vehicles), that are willing to participate in this system, or edge (or fog) routers that are deployed by "edge service providers (ESP)" providing computing, communication, and storage services for the city, e.g., at the cellular base stations, on the streets, or on top of tall buildings.

In particular, we propose a two-layer architecture for TrafficChain, which includes local chains, one for each road segment, and a global chain. For each local chain, the "local miners" are the nearby miners who would like to participate in the traffic status collection for the corresponding road segment. Each block on a local chain includes all the reports for the corresponding road segment and is only multicasted to these nearby local miners that are on this particular local chain. For the global chain, all the miners in the city can be the "global miners" for this global chain. Each block on the global chain contains the aggregated report on the traffic status of each road segment and is broadcasted to all the global miners. Whenever a vehicle needs to find a route to some location, it can retrieve the necessary traffic status from the global chain, for example, by inquiring the nearby global miners.

Besides, we aim to address the following critical issues in TrafficChain, including preservation of vehicles' privacy, resilience towards the aforementioned two types of attacks, and system anomaly detection. In the following, we first introduce the architecture of the proposed TrafficChain, and then elaborate on the privacy-preserving scheme and the security assuring algorithms.

B. THE ARCHITECTURE OF TrafficChain

The essential information that TrafficChain collects and stores is the passing time cost for each road segment. In general, each vehicle will report the passing time cost for the road segment that it just passed down, while the miner in

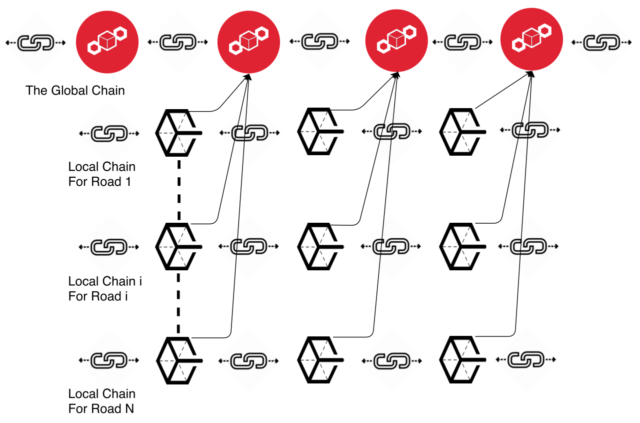

FIGURE 3. The architecture of TrafficChain.

TrafficChain will aggregate the reports and generate the estimated time cost for each road segment. However, if every vehicle simply broadcasts their reports to all the miners in the city, e.g., through a 5G cellular network or an edge network covering the whole city, the communication network will suffer from very high communication overhead, while every miner will be overloaded with a very high computational burden. Considering this practical issue, we propose a two-layer blockchain architecture for TrafficChain, which includes local chains and a global chain as shown in Fig. 3. There is a local chain for each road segment. All the vehicles passing down a road segment will report their individual passing time cost to the corresponding local chain. The local miners compete through Proof-of-Work (PoW) to determine who gets to write the next block containing all the reports from the vehicles. The winning local miner will broadcast the new block to all the miners on the local chain and those on the global chain. After receiving the blocks from the local chains, the global miners employ PoW as the consensus protocol to determine who "mines" the next block containing the traffic status of the city and all the reports from the local miners. The traffic status is obtained based on the local reports, by employing our proposed LSTM based aggregation scheme that will be introduced later.

1) BLOCK STRUCTURES

The structures for the blocks on the local chains and on the global chain are shown in Fig. 4 and Fig. 5, respectively.

In a local chain block, the block header stores the previous block's hash value, timestamp, a Merkle tree root, and a nonce. In the block body, the data is stored in a Merkle tree. For a local chain j, the data includes aggregated time cost, aggregated reports quantity, all the traffic reports for the corresponding road segment j broadcasted by the passing by vehicles, which are the passing time costs on the road segment j and will be detailed later.

Different from local chains, the global chain contains the global traffic information for the whole city. In a global chain block, the block header has the same structure as that of a

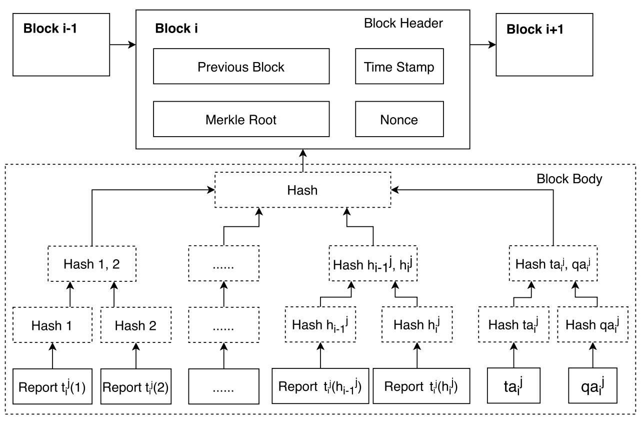

FIGURE 4. **The structure of the blocks on local chain *j*, which is corresponding to road segment *j*.**

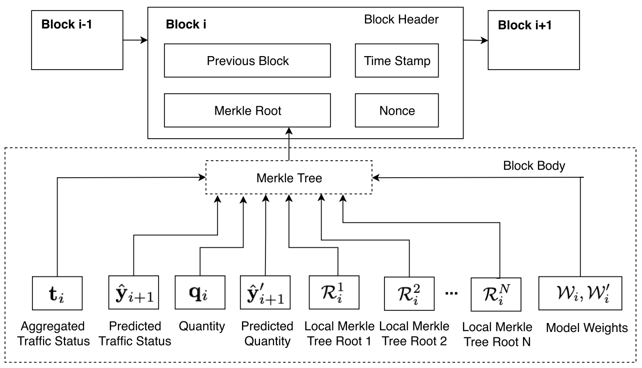

FIGURE 5. **The structure of the blocks on the global chain.**

local chain block. In the block body, the first element is the city traffic status vector $\mathbf{t}_i$ representing the aggregated time cost for each road in the time slot i, which collected from local miners who aggregate local traffic reports. The second element is $\hat{\mathbf{y}}_{i+1}$, i.e., the predicted traffic status in the next time slot by our proposed time cost estimation LSTM (TE-LSTM), which will be used by the local miners to aggregate local traffic reports in the next time slot. The third element is $\mathbf{q}_i$ representing the aggregated traffic report quantity for each road segment. The fourth element is $\hat{\mathbf{y}}'_{i+1}$, which is the predicted traffic report quantity for the next time slot. The next elements are the local Merkel trees' roots. Note that every global miner has all the local chains already. Thus, having only the local Merkel trees' roots in the global block is enough and can greatly reduce the size of the global block. The rest two elements are the parameters from our proposed LSTM based aggregation methods, i.e., model weights, which are used for defending against Byzantine attacks and Sybil attacks. All the above elements are stored in a Merkel tree, whose root is included in the global block header.

The blocks in both local chains and the global chain are chained up chronologically. Each block records the data collected in a time slot, the length of which can be set as needed.

2) TRAFFIC REPORTERS

In TrafficChain, the traffic reporters typically refer to the vehicles driving in the streets, which are encouraged to report the recorded time cost for the road segments that it has passed down. In the meantime, traffic reporters' privacy as defined in Sec. 4.2.1, i.e., their identities and driving routes, need to be protected.

Towards this goal, each user first generates its own blockchain address, e.g., the SHA256 and RIPEMD160 hash functions on its ECDSA public key like that on the bitcoin blockchain [32]. Note that each user can generate an unlimited number of addresses on a given private key, for example, similar to that in sequential or hierarchical deterministic wallets, and hence use different addresses for reports of different road segments to protect its identity.

The traffic reports are generated as follows. Consider a user k on road segment j in the time slot i. It will first generate a message tuple:

$$m_i^j(k) =< road_j, t_i^j(k), timestamp >, \tag{3}$$

where $road_j$ is the ID of the road segment j, $t_i^j(k)$ is the reported passing time cost, $timestamp$ represents the time when the report is generated. Next, user k signs the message with a signature $< \mathrm{r}_i^j(k), \mathrm{s}_i^j(k) >$. Particularly, all users in the system share the same curve parameters $(CURVE, G, n)$, where $CURVE$ is an elliptic curve equation, G is an elliptic curve base point, and n is a prime and the order of G. To generate a signature, the detail process is as follows. User k first generates a private key integer $a \in [1, n-1]$, and a public key with elliptic curve point multiplication, i.e., $b = a \times G$. The signature $< \mathrm{r}_i^j(k), \mathrm{s}_i^j(k) >$ is generated by

$$\mathrm{r}_i^j(k) = x_i^j \bmod n, \tag{4}$$

$$\mathrm{s}_i^j(k) = R^{-1}(H(m) + \mathrm{r}_i^j a), \tag{5}$$

respectively. Here, $H(\cdot)$ is the hash function SHA256, R is a secure per message random integer on $[1, n-1]$, x_i^j is from a curve point $(x_i^j, y_i^j) = R \times G$. At last, user k broadcasts the following traffic report to the local miners for the road segment j in the time slot i:

$$\mathcal{R}_i^j(k) =< b^k, \mathrm{r}_i^j(k), \mathrm{s}_i^j(k), m_i^j(k) > \tag{6}$$

where b^k is the address of user k. The process for user k to generate a traffic status report for road segment j in the time slot i is summarized in Algorithm 1.

Note that the public and private keys are utilized for protecting user identities and signing the message, while the messages are not encrypted. Thus, attacks like the Dolev-Yao model [33] are not concerns of our system. Besides, to motivate the users to submit traffic status reports, we design an incentive mechanism for the regular nodes that broadcast traffic status. The details of the incentive mechanism will be detailed later.

Algorithm 1 Traffic Report Generation

```
Input: $CURVE$, $G$, $n$, and current time $timestamp$.
    Private key $a \leftarrow 1 \leq a \leq n-1$; public key $b \leftarrow a \times G$.
    TrafficChain address $\leftarrow b$.
    Message $m_i^j(k) \leftarrow < road_j, t_i^j(k), timestamp >$.
    Signature $< \mathrm{r}_i^j(k), \mathrm{s}_i^j(k) > \leftarrow$ Calculate equations (4) & (5).

    Report $\mathcal{R}_i^j(k) \leftarrow < b^k, \mathrm{r}_i^j(k), \mathrm{s}_i^j(k), m_i^j(k) >$.
Output: $\mathcal{R}_i^j(k)$.
```

3) COMPUTING NODES

The computing nodes or miners compete through PoW to win the authority for creating new blocks. As mentioned before, the winning local miner will broadcast the new block to all the miners on the local chain and those on the global chain. After receiving the blocks from the local chains, the global miners also compete through PoW to get to write new blocks on the global chain. The winning global miner employs our proposed LSTM based aggregation algorithms to aggregate all the local reports, which are resilient to Byzantine and Sybil attacks and will be discussed next.

C. SECURE TRAFFIC REPORT AGGREGATION ON TrafficChain

Recall that the passing time cost reports about each road segment need to be aggregated as shown in Eq. (1). How to ensure that the reports aggregation is correct and resilient against both Byzantine attackers and Sybil attackers is a very challenging and critical problem in TrafficChain. In this section, we design a secure aggregation method by utilizing a deep learning model called LSTM.

1) LSTM BASED SECURE REPORT AGGREGATION

Deep learning techniques, such as Artificial Neural Network (ANN), Recurrent Neural Network (RNN), LSTM, have been widely employed in various applications due to their strong capability in data pattern and correlation mining [34], [35]. In TrafficChain, it is obvious that the road passing time cost for and the number of vehicles on each road segment are highly related to the history data on the same road segment and the data on other road segments, which essentially indicates the spatiotemporal correlation among the traffic status data. Therefore, we employ the LSTM model to predict the traffic status and the number of vehicles on all the road segments, which are then used to defend against Byzantine and Sybil attacks. In the following, we describe a Byzantine attack resilient algorithm and a Sybil attack resilient algorithm, respectively, in detail.

a: A BYZANTINE ATTACK RESILIENT ALGORITHM

First, we develop a time cost estimation LSTM (TE-LSTM) neural network for global miners to predict the traffic status on all the road segments. In particular, the TE-LSTM network is trained with a data batch denoted by $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_Q\}$,

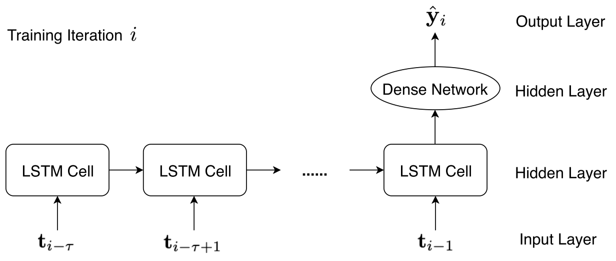

FIGURE 6. The structure of the TE-LSTM network.

where $\mathbf{X}_i \in \mathbf{X}$ is a data sample. For each data sample, we have $\mathbf{X}_i = \{\mathbf{ta}_{i-\tau}, \mathbf{ta}_{i-\tau+1}, \ldots, \mathbf{ta}_{i-1}\}(1 \leq i \leq Q)$. Note that $\mathbf{ta} = [ta_i^1, ta_i^2, \ldots, ta_i^N]$ is the aggregated traffic status on all the road segments in the time slot i (ta_i^j represents the aggregated time cost for road segment j in the time slot i and will be introduced later) collected from local miners, τ is the lookback window size representing the amount of temporal information used for each prediction. The structure of the TE-LSTM network is shown in Fig. 6. For each data sample $\mathbf{X}_i$, we expect the output of the network $\hat{\mathbf{y}}_i$ to be close to $\mathbf{y}_i = \mathbf{ta}_i$. Particularly, we denote by $G(\cdot)$ the classification function of the proposed TE-LSTM network, and $\mathcal{W}$ the model parameter set that needs to be optimized during the training process. The objective function of TE-LSTM, denoted by $J(\mathcal{W})$, is:

$$J(\mathcal{W}) = \frac{1}{Q}\sum_{i=1}^{Q}\phi(G(\mathcal{W}; \mathbf{X}_i), \mathbf{y}_i) + \frac{\lambda}{2}||\mathcal{W}||_2^2 \tag{7}$$

where $\phi(\cdot)$ is the loss function, $||\mathcal{W}||_2^2$ is the regularization term, and $\lambda > 0$ is the regularization coefficient.

Note that all the global miners share the same TE-LSTM network that can predict the passing time costs on all the road segments in the city. The TE-LSTM network can be trained and initialized by a certain winning global miner using the history traffic status stored on the global chain, and published in the new global block. The winning global miner also predicts the passing time costs on all the road segments in the next time slot based on the TE-LSTM model and writes them in the new global block. In the next time slot, when the winning local miner aggregates the passing time cost reports for the corresponding road segment, it employs the predicted value by the winning global miner in the previous time slot as a reference value.

Second, by taking advantage of the predicted time cost by TE-LSTM, we propose our Byzantine resilient aggregation method based on an existing scheme Krum [36], which is called Mean-Around-Krum. Recall that in the time slot i for road segment j, local miners have the traffic reports by vehicles denoted by $\mathbf{t}_i^j = \{t_i^j(1), \ldots, t_i^j(h_i^j)\}$, where h_i^j is the number of reports that include both honest and bogus reports. The aggregated time cost ta_i^j is calculated as:

$$ta_i^j = (1 - \frac{\mu}{|t_i^{j*} - \hat{y}_i^j|})t_i^{j*} + (\frac{\mu}{|t_i^{j*} - \hat{y}_i^j|})\hat{y}_i^j, \tag{8}$$

$$t_i^{j*} = \frac{1}{m} \sum_{k \to k^*} t_i^j(k), \tag{9}$$

$$k^* = \operatorname{argmin}_{I \in [1, h_i^j]} \sum_{p \to I} |t_i^j(p) - t_i^j(I)|. \tag{10}$$

Here, $\hat{y}_i^j$ is the estimated time cost given by TE-LSTM from the last time slot $i-1$, t_i^{j*} is the aggregated value from Mean-Around-Krum method, $k \to j$ means the m closest reported costs to t_j, μ is a control parameter. Particularly, Eq. (10) is from the Krum method, which is to find the index of the report that the m closest neighbors have the smallest total distance from. Our final aggregated time cost ta_i^j is a weighted sum of the cost t_i^{j*} given by Mean-Around-Krum and the cost $\hat{y}_i^j$ estimated by TE-LSTM. As t_i^{j*} gets close to $\hat{y}_i^j$, it means that the system is under a regular pattern, and the weight of $\hat{y}_i^j$ is larger. Otherwise, it shows that system may be subject to an abnormal pattern because of sudden changes, e.g., traffic accidents or attacks, and hence the weight of $\hat{y}_i^j$ decreases.

Recall that the winning local miner broadcasts the new local block to all the miners on the local chain and those on the global chain. When a winning global miner gets the authority to write a new global block, it will have the aggregated passing time cost on all the road segments in the current time slot, and utilize this new traffic status data to update the current TE-LSTM model. Specifically, at the end of the ith time slot, the winning global miner can have a new training sample $\mathbf{X}_i = \{\mathbf{ta}_{i-\tau}, \ldots, \mathbf{ta}_{i-1}\}$, $\mathbf{y}_i = \mathbf{ta}_i$. Then, it employs the Stochastic Gradient Descent (SGD) algorithm to minimize the objective function and in turn update the model parameter set $\mathcal{W}$, where the gradient is calculated as

$$\nabla J(\mathcal{W}_{i-1}) = \frac{d}{d\mathcal{W}_{i-1}} \phi(G(\mathcal{W}_{i-1}; \mathbf{X}_i), \mathbf{y}_i) + \lambda \mathcal{W}_{i-1} \tag{11}$$

Note that the loss function ϕ is considered to be convex and differentiable, e.g., the mean square loss function. Subsequently, the winning global miner updates the model parameter set as follows:

$$\mathcal{W}_i = \mathcal{W}_{i-1} + \eta \nabla J(\mathcal{W}_{i-1}) \tag{12}$$

where η is a control parameter.

The winning global miner finally writes the updated TE-LSTM model $\mathcal{W}_i$, aggregated time cost $\mathbf{ta}_i$, predicted time cost for the next time slot $\hat{\mathbf{y}}_{i+1}$ obtained through the new model $\mathcal{W}_i$ and input $\mathbf{X}_i$, and $\nabla J(\mathcal{W}_{i-1})$ into the new global block as shown in Fig. 5.

b: A SYBIL ATTACK RESILIENT ALGORITHM

Next, we build another LSTM network for quantity estimation of traffic reports for each road segment, called QE-LSTM. Similar to the passing time cost, the quantity of traffic reports for each road segment is also correlated to one another in a spatiotemporal manner. Therefore, we follow the same way as we design TE-LSTM to build QE-LSTM. Particularly, the ith data sample is $\mathbf{X}'_i = \{\mathbf{qa}_{i-\tau}, \ldots, \mathbf{qa}_{i-1}\}$, where $\mathbf{qa}_i = [qa_i^1, \ldots, qa_i^N]$ is the vector of the aggregated quantities of traffic reports on all the road segments in the time slot i and stored in the ith global block. Similar to Eq. (8) (note that in this case qa_i^{j*} denotes the aggregated number of reports while $y_i^{j'}$ is the predicted number of reports by QE-LSTM), (11), and (12) for TE-LSTM, in the time slot i, with the sample $\mathbf{X}'_i$ and its label $\mathbf{y}'_i = \mathbf{qa}_i$, the winning global miner can calculate the gradients $\nabla J'(\mathcal{W}'_{i-1})$, update the QE-LSTM model parameter set $\mathcal{W}'_i$, and predict the quantity of traffic reports in the next time slot $\hat{\mathbf{y}}'_i$. Moreover, the winning global miner compares the predicted quantity of reports $\hat{\mathbf{y}}'_i$ from the last time slot with the finally aggregated quantity of reports $\mathbf{qa}_i$. If $|qa_i^j - \hat{y}'^j_i|/\hat{y}'^j_i \leq \epsilon$, the global winning miner decides that there are no Sybil attacks on this road segment, and proceeds as mentioned before. Otherwise, the global winning miner takes the predicted quantity of traffic reports $\hat{y}'^j_i$ as the real quantity, the predicted passing time cost $\hat{y}_i^j$ as the real passing time cost. It then writes the updated QE-LSTM model $\mathcal{W}'_i$, aggregated traffic report quantity $\mathbf{qa}_i$, predicted report quantity for the next time slot $\hat{\mathbf{y}}'_i$ obtained through the new model $\mathcal{W}'_i$ and input $\mathbf{X}'_i$, and $\nabla J(\mathcal{W}'_{i-1})$ into the new local block and broadcasts this new block to all the global miners.

The whole process for defending against Byzantine attacks and Sybil attacks and update the local and global chains is described in Algorithm 2.

Algorithm 2 Defending Against Byzantine and Sybil Attacks

```
Input: At the ith time slot, obtained data sample $\mathbf{X}_i$, $\mathbf{X}'_i$, weights $\mathcal{W}_{i-1}$, $\mathcal{W}'_{i-1}$ from the global chain, threshold $\epsilon$ and $l$.
1:  Winning Miner on Local Chain $j, j \in [1, N]$:
2:      $ta_i^j \leftarrow$ aggregate reports $\mathcal{R}_i^j$ on road segment $j$.
3:      $qa_i^j \leftarrow$ length of $\mathbf{t}_i^j$.
4:      $NewLocalBlock \leftarrow ta_i^j, qa_i^j, \mathcal{R}_i^j$.
5:  Winning Miner on Global Chain:
6:      for each road $j$ do
7:          $\mathbf{ta}_i^j, \mathbf{qa}_i^j \leftarrow$ obtained from local chain $j$.
8:          if $|\mathbf{qa}_i^j - \hat{\mathbf{y}}'^j_{i-1}|/|\hat{\mathbf{y}}'^j_{i-1}| \leq \epsilon$ then
9:              $\mathbf{y}_i^j \leftarrow \mathbf{ta}_i^j$.
10:             $\mathbf{y}'^j_i \leftarrow \mathbf{qa}_i^j$.
11:         else
12:             $\mathbf{y}_i^j \leftarrow \hat{\mathbf{y}}_{i-1}^j$.
13:             $\mathbf{y}'^j_i \leftarrow \hat{\mathbf{y}}'^j_{i-1}$.
14:         end if
15:     end for
16:     $\nabla J(\mathcal{W}_{i-1}), \mathcal{W}_i \leftarrow$ update the TE-LSTM with $\mathbf{X}_i$, $\mathbf{y}_i$, $\mathcal{W}_{i-1}$.
17:     $\nabla J'(\mathcal{W}'_{i-1}), \mathcal{W}'_i \leftarrow$ update the QE-LSTM with $\mathbf{X}'_i$, $\mathbf{y}'_i$, $\mathcal{W}'_{i-1}$.
18:     $NewGlobalBlock \leftarrow \mathbf{ta}_i, \hat{\mathbf{y}}_{i+1}, \mathbf{qa}_i, \hat{\mathbf{y}}'_{i+1}, \mathcal{W}_i, \mathcal{W}'_i$.
Output: New blocks on local and global chains.
```

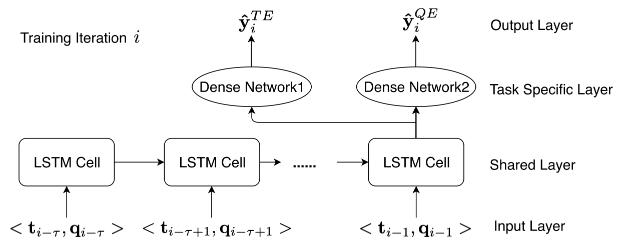
Training Iteration i
$\hat{\mathbf{y}}_i^{TE}$
$\hat{\mathbf{y}}_i^{QE}$
Output Layer
Dense Network1
Dense Network2
Task Specific Layer
LSTM Cell
LSTM Cell
LSTM Cell
Shared Layer
$< \mathbf{t}_{i-\tau}, \mathbf{q}_{i-\tau} >$
$< \mathbf{t}_{i-\tau+1}, \mathbf{q}_{i-\tau+1} >$
$< \mathbf{t}_{i-1}, \mathbf{q}_{i-1} >$
Input Layer

FIGURE 7. **The structure of the MT-LSTM network.**

2) MULTI-TASK LEARNING BASED SECURE REPORT AGGREGATION

Since the passing time costs and the number of vehicles on all the road segments are highly correlated as we mentioned above, we further extend our traffic report algorithm by utilizing Multi-Task Learning (MTL), which can potentially enhance the estimation of both passing time costs and vehicle quantities. In particular, the structure of our Multi-task LSTM network (MT-LSTM) is shown in Fig. 7. In the time slot i, the input sample for training and updating the network is $\mathbf{X_i} = \{\mathbf{x}_{i-\tau}, \ldots, \mathbf{x}_{i-1}\}$, where $\mathbf{x}_i =< \mathbf{t}_i, \mathbf{q}_i >$, and $< \cdot, \cdot >$ is to concatenate the two vectors. The labels for input $\mathbf{X}_i$ are the target values for two specific tasks, i.e., $\mathbf{y}_i^{TE} = \mathbf{t}_i$ for time cost estimation and $\mathbf{y}_i^{QE} = \mathbf{q}_i$ for quantity estimation as defined before. The predictions of the network are $\hat{\mathbf{y}}_i^{TE}$ and $\hat{\mathbf{y}}_i^{QE}$ for the two tasks, respectively. The two tasks are trained simultaneously, where the shared LSTM layer captures the correlation among the two tasks and the task-specific layer includes an individual fully connected layer for each task. In so doing, the MT-LSTM fulfills three advantages. First, MT-LSTM helps prevent overfitting since common representations are learned [37]. Second, MT-LSTM further improves the accuracy of predicting passing time costs and report quantities by learning more information from mining the correlations among them. Third, MTL structure helps simplify the network and makes it light-weighted.

D. AN INCENTIVE MECHANISM

In order to motivate users to participate in the system and submit traffic status reports, we also propose an incentive mechanism. In particular, we reward users whose reports are deemed valid when aggregated by the winning local miners. In other words, reports $t_i^j(k)$ in Eq. (9) are considered valid, which contribute to the final aggregated $t_{a_i}^j$. The reporter can get rewards from TrafficChain operator, e.g., the city or Department of Transportation. In addition, since each report in TrafficChain has an ECDSA based signature, TrafficChain can easily check the authenticity of the reports.

E. SECURITY ANALYSIS

We first discuss the system resilience to Byzantine attacks. Under Byzantine attacks, attackers submit bogus reports containing misleading passing time on road segments. Recall the naive aggregation method in Eq. (2). A few bogus reports would severely deviate the estimated passing time cost $\overline{t}_i^j$ from its true value. In the following theorem, we prove that the impact of Byzantine attacks is limited in our Mean-Around-Krum aggregation method. We drop the subscripts and superscripts to simplify the notations.

Theorem 1: Consider the reports for a road segment where $2f < m \leq h$. f and h denote the number of bogus reports, and that of honest reports, respectively. Let $\mathbf{t}^{(b)} = \{t_1^{(b)}, \ldots, t_f^{(b)}\}$ be all the Byzantine reports, where $t_l^{(b)} \leq t_{l+1}^{(b)}, l \in [1, f-1]$, $\mathbf{t}^{(h)} = \{t_1^{(h)}, \ldots, t_h^{(h)}\}$ be all the honest reports, where $t_l^{(h)} \leq t_{l+1}^{(h)}, l \in [1, h-1]$, and $\{t_{min}, t_{max}\} = \{t_1^{(h)}, t_h^{(h)}\}$. The output of Mean-Around-Krum t^* in the worst case is bounded by

$$t_{min} - \frac{f}{m}(t_{max} - t_{min}) < t^* < t_{max} + \frac{f}{m}(t_{max} - t_{min}).$$

Proof: To analyze the bounds of the output of Mean-Around-Krum t_i^{j*} in the worst case, we consider the scenario where all the Byzantine reports are included in the aggregation. So, Eq. (9) can be rewritten into

$$t^* = \frac{1}{m}(\sum_i^f t_i^{(b)} + \sum_j^{m-f} t_j^{(h)}).$$

Besides, the ground truth is $\frac{1}{h}\sum_{j=1}^h t_j^{(h)} \in [t_{min}, t_{max}]$, while the second term in t^* is $\sum_i^{m-f} t_j^{(h)} \in [(m-f)t_{min}, (m-f)t_{max}]$. Since we have $m > 2f$, although all bogus reports are included in the aggregation, $t(k^*)$ is still located in $[t_{min}, t_{max}]$ according to Eq. (10). The proof of this is given in Lemma 1 in Appendix.

The objective of attackers is to steer the aggregated value t_i^{j*} as small or large as possible. Therefore, we discuss the attack in two worst cases.

Case I: All the bogus reports aim to steer the passing time smaller, where $t_f^{(b)} < t_{min} \leq t(k^*)$. From Eq. (10), we get

$$\begin{aligned} &|t_1^{(b)} - t(k^*)| + \cdots + |t_f^{(b)} - t(k^*)| \\ &\quad + |t_1^{(h)} - t(k^*)| + \cdots + |t_{m-f}^{(h)} - t(k^*)| \\ &< |t_{m-f+1}^{(h)} - t(k^*)| + \cdots + |t_m^{(h)} - t(k^*)| \\ &\quad + |t_1^{(h)} - t(k^*)| + \cdots + |t_{m-f}^{(h)} - t(k^*)|. \end{aligned}$$

Due to $t_f^{(b)} < t(k^*) < t_{m-f+1}^{(h)}$, the above inequality can be rewritten into

$$\begin{aligned} &ft(k^*) - \sum_{l=1}^{f} t_l^{(b)} \\ &\quad < \sum_{l=m-f+1}^{m} t_l^{(h)} - ft(k^*) \\ &\quad \Rightarrow \sum_{l=1}^{f} t_l^{(b)} > 2ft(k^*) - \sum_{l=m-f+1}^{m} t_l^{(h)} > 2ft_{min} - ft_{max} \\ &\quad \Rightarrow \sum_{l=1}^{f} t_l^{(b)} + \sum_{l=1}^{m-f} t_l^{(h)} > 2ft_{min} - ft_{max} + (m-f)t_{min} \\ &\quad = (m+f)t_{min} - ft_{max} \\ &\quad \Rightarrow t^* > t_{min} - \frac{f}{m}(t_{max} - t_{min}) \end{aligned} \tag{13}$$

Case II: All the bogus reports aim to steer the passing time greater, where $t_1^{(b)} > t_{max} \geq t(k^*)$. Similarly, we get

$$\begin{aligned}
&|t_1^{(b)} - t(k^*)| + \cdots + |t_f^{(b)} - t(k^*)| \\
&\quad + |t_{h-m+f+1}^{(h)} - t(k^*)| + \cdots + |t_h^{(h)} - t(k^*)| \\
&< |t_{h-m+1}^{(h)} - t(k^*)| + \cdots + |t_{h-m+f}^{(h)} - t(k^*)| \\
&\quad + |t_{h-m+f+1}^{(h)} - t(k^*)| + \cdots + |t_h^{(h)} - t(k^*)|.
\end{aligned}$$

Because of $t_1^{(b)} > t(k^*) > t_{h-m+f}^{(h)}$, we have

$$\begin{aligned}
&\sum_{l=1}^{f} t_l^{(b)} - ft(k^*) \\
&\quad < ft(k^*) - \sum_{l=h-m+1}^{h-m+f} t_l^{(h)} \\
&\Rightarrow \sum_{l=1}^{f} t_l^{(b)} < 2ft(k^*) - \sum_{l=h-m+1}^{h-m+f} t_l^{(h)} < 2ft_{max} - ft_{min} \\
&\Rightarrow \sum_{l=1}^{f} t_l^{(b)} + \sum_{l=1}^{m-f} t_l^{(h)} < 2ft_{max} - ft_{min} + (m-f)t_{max} \\
&= (m+f)t_{max} - ft_{min} \\
&\Rightarrow t^* < t_{max} + \frac{f}{m}(t_{max} - t_{min}) \qquad (14)
\end{aligned}$$

Thus, Theorem 1 follows. □

Note that the attackers need to know $\mathbf{t}^{(h)}$ in order to impose the worst impact on the aggregation results, which can be challenging for attackers. Otherwise, the malicious reports $\mathbf{t}^{(b)}$ can be either too far away from or too close to $\mathbf{t}^{(h)}$. In the first case, the malicious reports may be ignored in the aggregation process due to the proposed Mean-Around-Krum scheme, while in the second case, the effect of attacks is mitigated to a large extent.

Next, we analyze Sybil attacks. Since every user submits its traffic reports with different addresses so as to mask its real identity and driving routes, Sybil attackers can take advantage of it to submit many bogus reports for each segment. To defend against Sybil attacks, our proposed QE-LSTM network detects attacks by comparing the collected number of traffic reports with the predicted one. Thus, if Sybil attacks are detected for a road segment, we discard all the reports and take the reference passing time, i.e., the predicted passing time by TE-LSTM network, as the aggregated passing time for this road segment. In so doing, the system performance is bounded by the prediction error of TE-LSTM.

Note that both accidents and attacks can result in longer than expected or predicted passing time on road segments. In fact, there are key differences between these two scenarios. Particularly, in the case of accidents, all honest users, and hence most users, will report longer than predicted passing time. However, in the case of attacks, only a few reports by attackers will include longer than expected passing time since extensive Sybil attacks will be detected. Thus, when accidents happen, the proposed TE-LSTM can correctly aggregate traffic reports and adapt to system status changes.

TABLE 1. Main attributes of data in CEbTS.

Attributes	Description	Type
Time	Date and time	D&T
Segment ID	Unique arbitrary number to represent each segment.	#
Street	Street name of the traffic segment.	T
Direction	Traffic flow direction for the segment.	T
Speed	Estimated speed in miles per hour.	#
Start&End Latitudes	Latitudes of the start and end of the segment.	#
Start&End Longitudes	Longitudes of the start and end of the segment.	#
Bus Count	Number of buses passed in the time slot.	#

V. PERFORMANCE EVALUATION

In this section, we conduct simulations to evaluate the performance of our TrafficChain system and focus on its information accuracy and attack resistance. In particular, we first show the performance of TE-LSTM network on predicting traffic statuses and compare with other machine learning methods. Then, we launch Byzantine attacks and Sybil attacks in the system, and test the resilience of TrafficChain to them respectively. Meanwhile, we compare with other existing Byzantine resilient aggregation methods. At last, we further analyze and discuss the simulation results.

A. DATASET

The dataset we use for simulations is a public dataset of Chicago city [38]. Specifically, we use the dataset of Chicago Traffic Tracker-Congestion Estimates by Traffic Segments (CEbTS). The city of Chicago divides the streets into segments, where each segment is typically a half-mile long in one direction of traffic. CEbTS gives the estimated speed for 1250 road segments covering 300 miles of streets, which is obtained by continuously monitoring and analyzing GPS traces of Chicago Transit Authority (CTA) buses. Speed data is generated every 10 minutes for the period of the year 2018, leading to a total of 50,457,500 data samples for 1250 road segments over 40,366 time slots. The main attributes of data in CEbTS are shown in Table 1. '#, T, D&T' represents number, text, and date & time data types, respectively. It should be noticed that a GPS trace is required for estimating speed on a road segment. Therefore, the speed information is missing in the dataset when there is no bus on a road segment, e.g., in some time slots of off-peak hours. In this case, the speed is marked as '−1' to reflect data unavailability. Besides, the speed '0' reflects full or partial street closure due to an unexpected event. For the longitudes and latitudes of start and end points of the segment, the ending latitude and longitude for a segment will be the same as the starting latitude and longitude for the segment next to it.

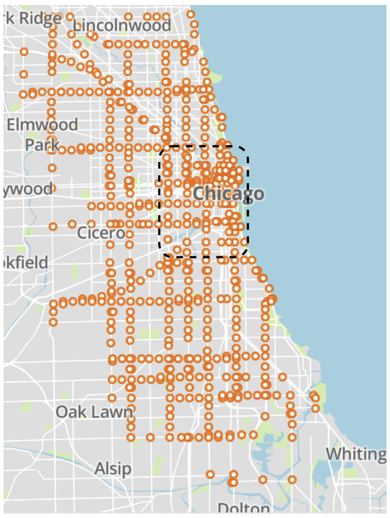

FIGURE 8. The data in CEbTS. Each point is a road segment, and there are 1250 points in total. The dashed line rectangle shows our area of interest, which contains a total of 419 road segments.

TABLE 2. RMSE of TE-LSTM, RNN, ARIMA, logistic regression, and linear regression.

ID	TE-LSTM	RNN	ARIMA	Logistic	Linear
#1	19.80	22.45	20.20	23.79	29.57
#25	18.79	19.92	19.24	23.62	28.44
#50	19.47	21.81	21.24	24.75	30.25
#100	13.82	16.23	13.97	21.70	27.44
Overall	18.58	19.93	19.92	23.57	29.14

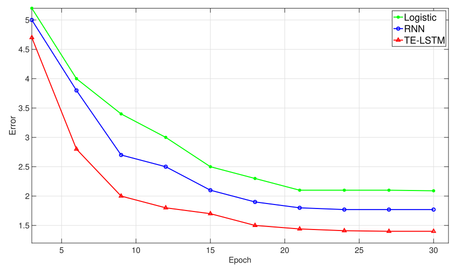

FIGURE 9. Average training loss of TE-LSTM, RNN, and logistic regression. It shows the training speed of our task-specific LSTM, and comparing models, i.e., RNN and Logistic Regression. The horizontal axis is the training epochs, and the vertical one is normalized MSE error with the unit of $10^{(-3)}$.

B. DATA PRE-PROCESSING

As shown in Fig. 8, we select the rectangular area in the City of Chicago as our area of interest, which contains 419 road segments. The CEbTS dataset provides estimated traffic speeds on the road segments, while our TrafficChain system collects the time cost for each road segment. Since each road segment is set to half a mile long, we calculate the time cost in seconds for road segment i as $t_i = \frac{0.5}{v_i} \times 3600$, where v_i is the estimated speed (mph) on the road segment i. Moreover, we set an upper limit of 600 seconds to the time cost for a road segment, which refers to street closure.

As mentioned before, the CEbTS dataset has missing data marked as '-1'. We fill the missing data with the most recent estimated traffic speed. This is because most missing data happen during off-peak hours when the traffic speed usually remains stable. Besides, recall that there are 40,366 time slots in total. We only use the data in the first select 10,000 (24.77%) time slots as training samples, and use the data in the rest time slots as testing samples.

C. PERFORMANCE OF TE-LSTM

1) NETWORK SETUP

We first evaluate the performance of TE-LSTM in traffic reports aggregation. We implement it in a multi-task architecture, with the prediction for time cost on each road segment being a task. The hard parameter sharing structure is adopted [37]. The lookback window size is set to 24, which is to analyze 4 hours' history data for making prediction in the next time slot of ten minutes. So, the size of each training sample has a dimension of 24×419, and the label is 1×419. We set the output of the parameter sharing layer to 128 nodes. Each task-specific layer maps the previous output to the particular road time cost with size $128 \rightarrow 1$. Normalization is also performed for the added at the input of the network. We choose Mean Square Error (MSE) as the loss function.

2) SIMULATION RESULTS

Table 2 presents the Root of Mean Square Error (RMSE) of TE-LSTM and other popular time series data prediction methods, including RNN, ARIMA, logistic regression, and linear regression. Since our TE-LSTM network is implemented as a multi-task structure, we show the testing results for some particular road segments, i.e., segments 1, 25, 50, 100, as well as the overall results for all the road segments. We can see that our TE-LSTM network outperforms all the other methods both in individual prediction for each road segment and in overall prediction.

Note that LSTM is usually difficult to be well trained when the dimension of the training samples gets huge. However, as the multi-task structure is employed, the parameter sharing layer can be pre-trained and only the task-specific layer parameters need to be updated, whose size is relatively small. We show the training speed on task-specific layers with the pre-trained parameter sharing layer in Fig. 9. We can observe that TE-LSTM can converge fast.

D. ATTACK RESILIENCE

Next, we test the performance of TrafficChain in terms of resilience to Byzantine and Sybil attacks.

1) RESILIENCE TO BYZANTINE ATTACKS

We launch Byzantine attacks in the system by adding Gaussian noises to the traffic reports. Particularly, we select three of the busiest road segments for simulations, i.e., 889, 922, and 1295, respectively. We set the number of the reports equal to the number of buses on a road segment. A certain

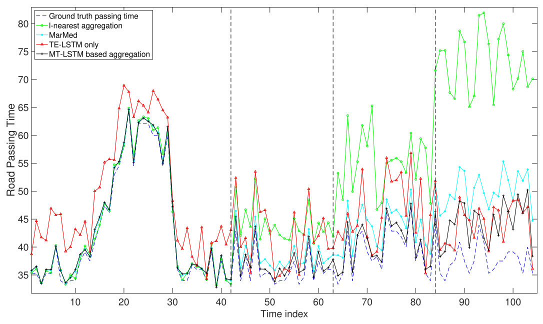

FIGURE 10. Resilience to Byzantine attacks on road segment #889. The horizontal axis represents time slots index in a day. The vertical axis represents the aggregated time cost. The dashed line is the ground truth of time cost by CEbTS. The vertical dashed lines separate data into 40%, 20%, 20%, 20%. $\sigma^{(H)} = 0.2$, $\sigma^{(M)} = 50$. In these four parts of data, 0%, 30%, 50%, and 80% of the reports are attacked, respectively.

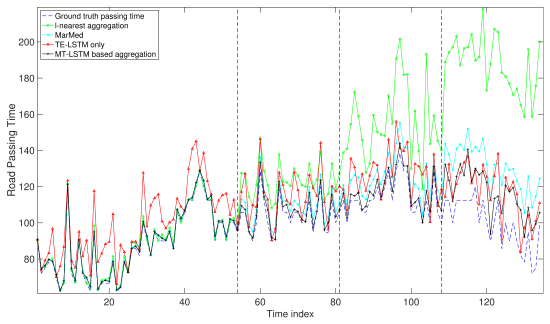

FIGURE 11. Resilience to Byzantine attacks on road segment #922. Setup is the same as in Fig. 10, except for $\sigma^{(M)} = 100$.

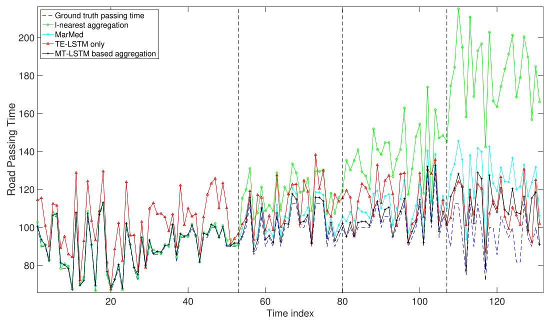

FIGURE 12. Resilience to Byzantine attacks on road segment #1295. $\sigma^{(M)} = 120$. Different from the setup in Fig. 10 and Fig. 11, in four parts of data separated by dashed lines, 0%, 20%, 40%, and 60% of the reports are attacked, respectively.

ratio of reports are added with Gaussian noises $|\mathcal{N}(0, \sigma^{(M)})|$. To simulate a practical scenario, we also add Gaussian noises $\mathcal{N}(0, \sigma^{(H)})$ to honest reports too. For each of the three road segments, we use the data on a particular day to test our MT-LSTM aggregation method. We compare our proposed MT-LSTM aggregation with two state-of-the-art schemes, i.e., l-nearest [29] and marginal median (MarMed) [30], as well as with using TE-LSTM only without the proposed Mean-Around-Krum scheme.

In particular, for road segment #889, we divide the time slots in a day into four parts as shown in Fig. 10. In the first 40% of time slots, no attack is launched, which means all the reports are submitted honestly. In the first 40% to 60% time slots, we add malicious Gaussian noises to 30% of the reports. In the first 60% to 80% time slots, we add malicious Gaussian noises to 50% of the reports. In the rest time slots, 80% of the reports are attacked. The results are shown in Fig. 10. In the first 40% time lots, all four schemes can obtain aggregated time cost fairly close to the ground truth. Among them, TE-LSTM returns the worst performance caused by prediction errors, which indicates the effectiveness of Mean-Around-Krum. When it comes to the second part, since 30% of the reports are attacked, the regular l-nearest method has the worst performance among all the schemes because it selects l-nearest reports to the mean of all reports to aggregate, which gets higher due to the attacks. Our MT-LSTM aggregation slightly outperforms MarMed. Then, when 50% of the reports are attacked, both MarMed and regular l-nearest methods include a number of attacked reports for aggregation and hence achieve degraded results, while our MT-LSTM aggregation scheme still obtains aggregated time cost close to the ground truth. In the last data part where 80% reports are attacked, our MT-LSTM aggregation inevitably includes some attacked reports for aggregation, but its performance is still better than the other two aggregation methods. The simulation results on the other two road segments are shown in Fig. 11 and Fig. 12, where $\sigma^{(M)}$ is set to 100 and 120, respectively. We can observe that the proposed MT-LSTM based aggregation can always achieve the best results close to the ground truth.

2) RESILIENCE TO SYBIL ATTACKS

Recall that Sybil attackers create many fake identities to submit malicious traffic reports. Our QE-LSTM detects Sybil attacks by comparing the number of collected traffic reports with the predicted one. Similar to TE-LSTM, we add a task-specific layer for predicting the number of reports for each road segment. The RMSE of the QE-LSTM is used to estimate whether Sybil attacks exist. In particular, let $e^{(q)}$ be the RMSE of QE-LSTM. If $|x' - \hat{x}'| \leq e^{(q)}$, where x' and $\hat{x}'$ are the total number of collected reports and the quantity prediction from QE-LSTM, respectively, there is no Sybil attack, and Mean-Around-Krum is employed to finally aggregate the time cost. Otherwise, it is determined that there are Sybil attacks. In this case, only the outputs of QE-LSTM and TE-LSTM are used for generating the final result and updating both LSTMs.

We first show the performance of QE-LSTM and MT-LSTM on the prediction of the number of traffic reports, and then evaluate their resilience to both Byzantine and Sybil attacks. Table 3 shows the prediction results of our QE-LSTM and MT-LSTM, compared with other prediction methods, including RNN, logistic regression, and linear regression. Particularly, bus counts information in CEbTS is used as the ground truth of the number of reports. We can see that MT-LSTM achieves better results than QE-LSTM since it

TABLE 3. RMSE results of MT-LSTM and QE-LSTM on predicting the quantity of passing vehicles compared with RNN, logistic and linear regression. The RMSE of three particular roads with segment IDs #889, #922, #1295, and the overall average RMSE on all roads are given. Bus counts information is used as the ground truth.

ID	MT-LSTM	QE-LSTM	RNN	Logistic	Linear
#889	2.20	2.54	3.05	3.63	4.2
#922	1.76	2.42	2.88	3.78	4.59
#1295	2.43	2.49	2.59	3.16	4.21
Overall	2.21	2.60	2.89	3.67	4.48

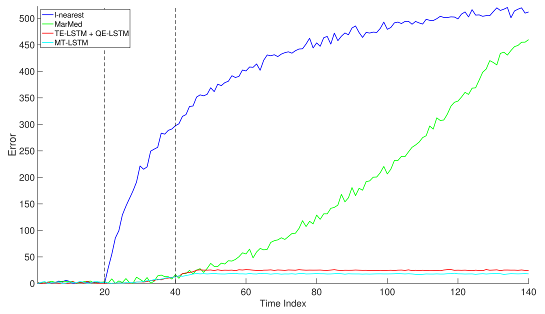

FIGURE 13. The system performance under both Byzantine and Sybil attacks. The vertical axis means the average error of all segments. From the first vertical dash line (20th time slot), the attacks are added, where each time slot is added by one more arbitrary report than the previous time slot. From the second dash line, the LSTM agent notices the attack and the information from reports are eliminated.

can learn more information, and that both MT-LSTM and QE-LSTM outperform the other state-of-the-art methods.

Fig. 13 shows the system performance under both Byzantine and Sybil attacks. The testing data includes a whole day period. In the first 20 time slots, there is no attack. After that, in each time slot, one more malicious report is added which reports the time cost of 600s. Four aggregation methods are tested, including regular l-nearest, MarMed, QE-LSTM, plus TE-LSTM, MT-LSTM. As the time index increases, there are more and more malicious reports, the regular l-nearest aggregation method is severely affected. MarMed obtains competitive performance when only a limited amount of Sybil attacks are launched. However, as there are over 20 malicious reports, i.e., when malicious reports are more than honest reports, MarMed gets very poor results. In this case, QE-LSTM and MT-LSTM will detect Sybil attacks due to $|x' - \hat{x}'| > e^{(q)}$, and solely use the prediction results of TE-LSTM as the time cost. Although the accuracy of time cost decreases under Sybil attacks, both QE-LSTM + TE-LSTM and MT-LSTM can obtain significantly better results than the other schemes.

E. TrafficChain UPDATE EFFICIENCY

Since, in practice, there can be a huge number of vehicles in a city participating in the system, the computing load in the system can be heavy. To address this problem, we have developed a two-layer blockchain architecture to enhance system efficiency. In this section, we test the time consumption of reports collection and model update by comparing TrafficChain with a normal single blockchain. In particular,

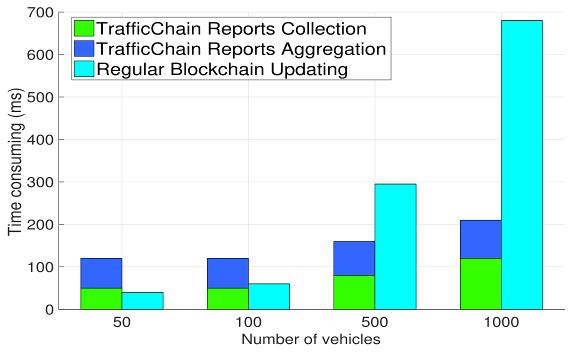

FIGURE 14. Time consumption for building a new global block. Green and blue bars are the time needed for reports collection and LSTM based aggregation processes in TrafficChain, respectively. The light blue bar is the updating process for a regular single blockchain.

the single blockchain collects reports from all vehicles in the system, and aggregate them with the regular l-nearest method. In TrafficChain, local miners first aggregate the traffic reports for their corresponding road segments and then submit reports to global miners for the model update. Since traffic reports aggregation is conducted by local miners in a distributed fashion, the reports collection process is much faster. Since the number of local chains is much smaller than the number of all the vehicles in the system, the model update process is much faster.

1) BLOCK UPDATE EFFICIENCY

Fig. 14 shows the results of time consumption of building a global block on TrafficChain and on a regular blockchain. The simulation is based on a Mobile Adhoc Network MANET in NetSim emulator [39] supported by a PC with 16GB of RAM and Intel i7 at 2.8Ghz. From the results, we can see that when there are fewer than 100 vehicles in the system, TrafficChain spends more time than a regular blockchain to collect reports and build a new global block. However, when there are more than 100 vehicles, the time needed by TrafficChain to build a new block remains relatively stable since the number of local chains remains the same, while the time needed by a regular blockchain to build a new block increases significantly.

2) REPORT AGGREGATION EFFICIENCY

Since the proposed deep neural network can be pre-trained offline, the time cost for running our proposed scheme mainly comes from two parts. One part is due to the online updating of the trained deep neural network, which is to calculate the gradients in the SGD algorithm as in Eq. (11). The other is due to the Mean-Around-Krum algorithm. Comparing with the second part, the time cost of online updating is negligible. The time complexity of Mean-Around-Krum depends on sorting the reports. As we use quicksort as the sorting algorithm, the time complexity of Mean-Around-Krum is lower and upper bounded by $O(n)$ and $O(nlog(n))$, respectively, when sorting n reports. On a PC with 16G RAM and Intel i7 CPU, our aggregation method only takes 6s to aggregate the reports for 2800 road segments in each time slot.

VI. CONCLUSION

In this paper, we have developed TrafficChain, a secure and privacy-preserving decentralized traffic information collection system. In particular, we have designed a two-layer blockchain architecture for efficient communication and block updating in TrafficChain. Besides, a privacy-preserving scheme has been devised to protect users' identities and driving routes. Moreover, we have considered two critical kinds of attacks, i.e., Byzantine and Sybil attacks, in TrafficChain, and developed novel deep learning based schemes to defend against them. Simulation results show that TrafficChain is both resilient to those attacks and efficient in generating new blocks.

APPENDIX

Lemma 1: Under the same conditions as in Theorem 1, in the worst case that all Byzantine reports are included in the aggreggation, we have $t(k^*) \in [t_{min}, t_{max}]$.

Proof: Note that $t_{min} = t_1^{(h)}$ and $t_{max} = t_h^{(h)}$. We first consider the worst case that the Byzantine attackers aim to maliciously make $t(k^*)$ very small, where $t_f^{(b)} < t_{min}$. Thus, the report set of "$p \to I$" in Eq. (10) can be written as $\mathbf{t} = \{t(1), t(2), \ldots, t(m)\} = \{t_1^{(b)}, \ldots, t_f^{(b)}, t_1^{(h)}, \ldots, t_{(m-f)}^{(h)}\}$.

Define $KR(I) = \sum_{p \to I} |t(p) - t(I)|$ and $\Delta_{ij} = |t(i) - t(j)|$. Then, we have

$$\begin{aligned} KR(s) &= \Delta_{1s} + \cdots + \Delta_{(s-1)s} \\ &\quad + \Delta_{(s+1)s} + \cdots + \Delta_{ms}, \\ KR(s+1) &= \Delta_{1(s+1)} + \cdots + \Delta_{s(s+1)} \\ &\quad + \Delta_{(s+2)(s+1)} + \cdots + \Delta_{m(s+1)}. \end{aligned}$$

Note that $\Delta_{i(s+1)} = \Delta_{is} + \Delta_{s(s+1)}$ for $1 \le i < s$ and $\Delta_{i(s+1)} = \Delta_{is} - \Delta_{s(s+1)}$ for $s+1 < i \le m$. Thus, we can obtain

$$\begin{aligned} KR(s+1) &= \Delta_{1s} + \Delta_{2s} + \cdots + \Delta_{(s-1)s} \\ &\quad + (s-1)\Delta_{s(s+1)} + \Delta_{s(s+1)} + \Delta_{(s+2)s} \\ &\quad + \cdots + \Delta_{ms} - (m-s-1)\Delta_{s(s+1)} \\ &= KR(s) + (2s-m)\Delta_{s(s+1)}. \end{aligned}$$

Due to $m \ge 2f$, we have $KR(s+1) < KR(s)$ for $1 \le s \le f$, which means $KR(f+1) < KR(f)$. Consequently, we get $t(k^*) \ge t_{min}$.

Similarly, in the worst case that Byzantine attackers aim to maliciously make $t(k^*)$ very large, where $t_1^{(b)} > t_{max}$, we can prove that $t(k^*) \le t_{max}$.

Therefore, Lemma 1 follows. □

REFERENCES

[1] Intel. (2018). *5G: Communications Key to Autonomous Driving*. https://iq.intel.com/5g-communications-key-autonomous-driving

[2] U. Ritzinger, J. Puchinger, and R. F. Hartl, "A survey on dynamic and stochastic vehicle routing problems," *Int. J. Prod. Res.*, vol. 54, no. 1, pp. 215–231, Jan. 2016.

[3] T. Yamamoto, K. Oku, and K. Kawagoe, "A real-time detection and localization system of unexpected events using crowd-sourcing," in *Proc. 13th Int. Conf. Mobile Ubiquitous Syst., Comput. Netw. Services (MOBIQUITOUS)*, 2016, pp. 130–135.

[4] H. G. Seif and X. Hu, "Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry," *Engineering*, vol. 2, no. 2, pp. 159–162, Jun. 2016.

[5] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proc. 2nd Int. Conf. Internet-Things Design Implement. (IoTDI)*, 2017, pp. 173–178.

[6] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018.

[7] R. A. Michelin, A. Dorri, M. Steger, R. C. Lunardi, S. S. Kanhere, R. Jurdak, and A. F. Zorzo, "SpeedyChain: A framework for decoupling data from blockchain for smart cities," in *Proc. 15th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, 2018, pp. 145–154.

[8] L.-A. Hirtan and C. Dobre, "Blockchain privacy-preservation in intelligent transportation systems," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE)*, Oct. 2018, pp. 177–184.

[9] S. E. Jabari and H. X. Liu, "A stochastic model of traffic flow: Gaussian approximation and estimation," *Transp. Res. B, Methodol.*, vol. 47, pp. 15–41, Jan. 2013.

[10] X. He and H. X. Liu, "Modeling the day-to-day traffic evolution process after an unexpected network disruption," *Transp. Res. B, Methodol.*, vol. 46, no. 1, pp. 50–71, Jan. 2012.

[11] C. Wang, X. Li, X. Zhou, A. Wang, and N. Nedjah, "Soft computing in big data intelligent transportation systems," *Appl. Soft Comput.*, vol. 38, pp. 1099–1108, Jan. 2016.

[12] S. Gisdakis, V. Manolopoulos, S. Tao, A. Rusu, and P. Papadimitratos, "Secure and privacy-preserving smartphone-based traffic information systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1428–1438, Jun. 2015.

[13] J. W. Brown, O. Ohrimenko, and R. Tamassia, "Haze: Privacy-preserving real-time traffic statistics," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2013, pp. 540–543.

[14] H. Zhu, X. He, X. Liu, and H. Li, "PTFA: A secure and privacy-preserving traffic flow analysis scheme for intelligent transportation system," *Int. J. Embedded Syst.*, vol. 8, no. 1, pp. 78–86, 2016.

[15] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.

[16] K. Rabieh, M. M. E. A. Mahmoud, and M. Younis, "Privacy-preserving route reporting schemes for traffic management systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2703–2713, Mar. 2017.

[17] Y. Zhang, Q. Pei, F. Dai, and L. Zhang, "Efficient secure and privacy-preserving route reporting scheme for VANETs," *J. Phys., Conf. Ser.*, vol. 910, nol. 1, Oct. 2017, Art. no. 012070.

[18] R. Rajbhandari, "Exploring blockchain-technology behind bitcoin and implications for transforming transportation," Texas A&M Transp. Inst., College Station, TX, USA, Tech. Rep., 2018.

[19] P. G. Saranti, D. Chondrogianni, and S. Karatzas, "Autonomous vehicles and blockchain technology are shaping the future of transportation," in *Proc. 4th Conf. Sustain. Urban Mobility*. Cham, Switzerland: Springer, 2018, pp. 797–803.

[20] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2663–2668.

[21] R. Rivera, J. G. Robledo, V. M. Larios, and J. M. Avalos, "How digital identity on blockchain can contribute in a smart city environment," in *Proc. Int. Smart Cities Conf. (ISC)*, Sep. 2017, pp. 1–4.

[22] A. R. Pedrosa and G. Pau, "ChargeltUp: On blockchain-based technologies for autonomous vehicles," in *Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst.*, 2018, pp. 87–92.

[23] M. Singh and S. Kim, "Intelligent vehicle-trust point: Reward based intelligent vehicle communication using blockchain," 2017, *arXiv:1707.07442*. [Online]. Available: http://arxiv.org/abs/1707.07442

[24] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, and N. Zhang, "A secure charging scheme for electric vehicles with smart communities in energy blockchain," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4601–4613, Jun. 2019.

[25] X. Huang, C. Xu, P. Wang, and H. Liu, "LNSC: A security model for electric vehicle and charging pile management based on blockchain ecosystem," *IEEE Access*, vol. 6, pp. 13565–13574, 2018.

[26] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-VN: A distributed blockchain based vehicular network architecture in smart city," *J. Inf. Process. Syst.*, vol. 13, no. 1, p. 84, 2017.

[27] M. Singh and S. Kim, "Blockchain based intelligent vehicle data sharing framework," 2017, *arXiv:1708.09721*. [Online]. Available: http://arxiv.org/abs/1708.09721

[28] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1832–1843, Dec. 2017.

[29] X. Chen, J. Ji, C. Luo, W. Liao, and P. Li, "When machine learning meets blockchain: A decentralized, privacy-preserving and secure design," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1178–1187.

[30] C. Xie, O. Koyejo, and I. Gupta, "Generalized Byzantine-tolerant SGD," 2018, *arXiv:1802.10116*. [Online]. Available: http://arxiv.org/abs/1802.10116

[31] P. Otte, M. de Vos, and J. Pouwelse, "TrustChain: A Sybil-resistant scalable blockchain," *Future Gener. Comput. Syst.*, to be published.

[32] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.

[33] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[34] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake prediction based on spatio-temporal data mining: An LSTM network approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 1, pp. 148–158, Jan. 2020.

[35] J. Ji, C. Luo, X. Chen, L. Yu, and P. Li, "Cross-domain sentiment classification via a bifurcated-LSTM," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2018, pp. 681–693.

[36] P. Blanchard, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 119–129.

[37] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*. [Online]. Available: http://arxiv.org/abs/1706.05098

[38] City of Chicago. (2019). *Chicago Data Portal*. [Online]. Available: https://data.cityofchicago.org/

[39] *Mobile Adhoc Networks (Manet)*. Accessed: 2017. [Online]. Available: https://www.tetcos.com/manet.html

QIANLONG WANG received the B.E. degree in electrical engineering from Wuhan University, China, in 2013, and the M.E. degree in electrical and computer engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2015, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Case Western Reserve University. His research interests include robust and secure deep-learning networks, security, and privacy in distributed and decentralized systems, e.g., cyber physical system (CPS) and the Internet of Things (IoT), and related applications, e.g., smart transportation, smart city, and smart healthcare.

TIANXI JI received the B.E. degree in telecommunication engineering from the Nanjing University of Posts and Telecommunications, China, in 2014, and the M.S. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Case Western Reserve University. His research interests include big data, edge computing, and security and privacy.

YIFAN GUO (Student Member, IEEE) received the B.S. degree in information and computing sciences from the Beijing University of Posts and Telecommunications, China, in 2013, and the M.S. degree in computer science from Northwestern University, Evanston, IL, USA, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Case Western Reserve University. His research interests include artificial intelligence, machine learning, and big data.

LIXING YU (Student Member, IEEE) received the B.S. degree in electrical engineering from the Beijing Institute of Technology, China, in 2012, and the M.S. degree in electrical engineering from George Washington University, Washington, DC, USA, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Case Western Reserve University. His research interests include deep learning and data mining.

XUHUI (TRACY) CHEN received the B.E. degree in information engineering from Xidian University, Xi'an, China, in 2012, the M.S. degree in electrical and computer engineering from Mississippi State University, in 2015, and the Ph.D. degree in computer engineering from Case Western Reserve University, in 2019. He joined the College of Aeronautics and Engineering, Kent State University, as an Assistant Professor, in August 2019. His research interests include cybersecurity (security and privacy in distributed machine learning, blockchain technology), artificial intelligence, with applications in the cyber-physical systems (UAV, Robot), and bioinformatics.

PAN LI (Member, IEEE) received the B.E. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2009. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Case Western Reserve University. His research interests include network science and economics, energy systems, security and privacy, and big data. He is a member of the ACM. He received the NSF CAREER Award, in 2012. He has served as the General Chair for MobiQuitous 2018, the IEEE Smart Data 2017, and the Technical Program Committee (TPC) Co-Chair for Wireless Networks Symposium, the IEEE ICNC 2018, Social Networks and Big Data Symposium, the IEEE ICCC 2017, Ad-hoc, Mesh, Machine-to-Machine and Sensor Networks Track, IEE VTC 2014, Physical Layer Track, Wireless Communications Symposium, WTS 2014, and Wireless Networking Symposium, the IEEE ICC 2013. He has also served as an Editor for the IEEE Transactions on Wireless Communications, the IEEE Wireless Communications Letters, the IEEE Journal on Selected Areas in Communications–Cognitive Radio Series, and the IEEE Communications Surveys and Tutorials, and a Feature Editor for the IEEE Wireless Communications.

• • •