Vehicle Position Correction: A Vehicular Blockchain Networks-Based GPS Error Sharing Framework

Changle Li, *Senior Member, IEEE*, Yuchuan Fu, Fei Richard Yu, *Fellow, IEEE*,
Tom H. Luan, *Senior Member, IEEE*, and Yao Zhang

***Abstract*—The positioning accuracy of the existing vehicular Global Positioning System (GPS) is far from sufficient to support autonomous driving and ITS applications. To remedy that, leading methods such as ranging and cooperation have improved the positioning accuracy to varying degrees, but they are still full of challenges in practical applications. Especially for cooperative positioning, in addition to the performance of methods, cooperators may provide false data due to attacks or selfishness, which can seriously affect the positioning accuracy. By fully exploiting the characteristics of blockchain and edge computing, this paper proposes a vehicular blockchain-based secure and efficient GPS positioning error evolution sharing framework, which improves vehicle positioning accuracy from ensuring security and credibility of cooperators and data. First, by analyzing the GPS error, a bridge can be established between the sensor-rich vehicles and the common vehicles to achieve cooperation by sharing the positioning error evolution at a specific time and location. Particularly, the positioning error evolution is obtained by a deep neural network (DNN)-based prediction algorithm running on the edge server. We further propose to use blockchain technology for storage and sharing the evolution of positioning errors, mainly to guarantee the security of cooperative vehicles and mobile edge computing nodes (MECNs). In addition, the corresponding smart contracts are designed to automate and efficiently perform storage and sharing tasks as well as solve inconsistencies in time scales. Extensive simulations based on actual data indicate the accuracy and security of our proposal in terms of positioning error correction and data sharing.**

***Index Terms*—Positioning error evolution, edge computing, DNN, vehicular blockchain, smart contract.**

I. Introduction

POSITIONING information is very important for vehicles, especially for autonomous vehicles, which can be used to navigate in real-time with other data such as geographic data and vehicle state data. Vehicles with accurate position information can adjust driving conditions according to dynamic changes in the driving environment, thus enabling safe driving and improving the driving experience.

Global Positioning System (GPS) is one of the most widely used technologies. Although GPS signals are available most of the time, accurate positioning is still challenging for supporting autonomous driving and other applications where positioning accuracy is critical. For instants, in urban canyons, due to multipath interference, GPS accuracy in urban canyons ranges between 30-50 meters [1]. Even under normal circumstances, GPS positioning errors may lead to wrong navigation, waste of time, and traffic accidents. Fortunately, for some vehicles with multiple sensors (hereinafter referred to as sensor-rich vehicles), such as camera and Lidar, other sensing methods can be used to improve their positioning accuracy [2], [3]. However, on the one hand, for current traffic scenarios, most common vehicles still exist, and it is impossible to obtain accurate sensing information. On the other hand, Lidar sensors are expensive, which restricts their large-scale application. Therefore, how to improve GPS positioning accuracy is still worth studying.

A lot of work has been proposed to improve the GPS positioning accuracy of vehicles, from hybrid systems, infrastructure-based positioning, vehicle cooperative positioning (VCP), cooperative positioning (CP) systems, fusion systems, to mobile phone-based positioning, etc. Demetriou *et al.* [1] proposes a system called CoDriv for providing the accurate position of sensor-rich vehicles to a common vehicle. Angelis *et al.* [4] propose to use the Global Navigation Satellite System (GNSS) and cellular networks to form a hybrid scheme for user positioning in urban scenarios. Alam and Dempster [5] discuss modern and conventional CP systems. Yassin and Rachid [6] explain the principles behind positioning techniques used in satellite networks, mobile networks, and Wireless Local Area Networks. Some infrastructure-based positioning methods are proposed (e.g., anchor nodes [7], and magnetic positioning system [8]). Wang *et al.* [9] design a dedicated short-range communications (DSRC)-based vehicle cooperative positioning enhancement system. Li *et al.* [10] develop roadside equipment (RSE)-assisted lane-level positioning method using GPS data and received signal strength (RSS) data. Walter *et al.* [11] propose to use the sensors of Android smartphone for vehicle navigation when the GPS signal is lost.

Manuscript received February 6, 2019; revised May 31, 2019 and October 8, 2019; accepted December 16, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant U1801266, Grant 61571350, and Grant 61601344, in part by the Key Research and Development Program of Shaanxi under Contract 2018ZDXM-GY-038 and Contract 2018ZDCXL-GY-04-02, in part by the Youth Innovation Team of Shaanxi Universities, and in part by the Science and Technology Projects of Xi'an, China, under Grant 201809170CX11JC12. The Associate Editor for this article was S.-H. Kong. *(Corresponding author: Changle Li.)*

C. Li, Y. Fu, and Y. Zhang are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: clli@mail.xidian.edu.cn).

F. R. Yu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada.

T. H. Luan is with the School of Cyber Engineering, Xidian University, Xi'an 710071, China.

Digital Object Identifier 10.1109/TITS.2019.2961400

These efforts use different methods to improve vehicle positioning accuracy to degrees. However, the methods based on expensive sensors are still challenging for applying to common vehicles. The ranging based method also has errors in the ranging process [10]. Therefore, starting from the GPS positioning error characteristics, it is very meaningful to study a scheme that can adapt to the current scenario of the coexistence of common vehicles and sensor-rich vehicles to facilitate positioning error correction. On the one hand, the mobile edge computing nodes (MECNs) can store accurate map information, and sensor-rich vehicles can easily obtain accurate positioning. On the other hand, thanks to the computing power and storage capabilities of edge computing, sensor-rich vehicles participating in cooperative positioning can store the positioning error model in MECNs to provide a reference for common vehicles. However, due to attacks or selfish reasons, cooperative vehicles and MECNs cannot be fully trusted, which may provide false data. How to improve the security and reliability of cooperators and data is worth pondering. Indeed, blockchain is a very promising technology for secure storing and sharing data in many application scenarios, such as the Internet of Things (IoT) and vehicular networks [12], [13]. Applying blockchain technology for vehicle position information sharing can bring many benefits, such as security, automation, and transparency [14]. Therefore, positioning accuracy can be improved by avoiding malicious false data and motivating more users to participate in cooperative positioning.

This work proposes a vehicular blockchain-based framework for improving GPS positioning accuracy by ensuring the security and credibility of cooperative data. In particular, as information providers, sensor-rich vehicles provide positioning errors obtained by other methods to common vehicles (*i.e.,* data requesters). Specifically, the characteristics of GPS error are analyzed and the GPS positioning error evolution model is obtained by the deep neural networks (DNN). Then we put forward a GPS error evolution sharing framework that utilizes edge computing and blockchain technology. In order to solve the problem of inconsistent block generation time and information exchange time scale, we designed smart contracts that run the data storage and sharing process securely and efficiently. The main contributions of this paper are four-fold:

- *System Solution*: A vehicular blockchain-based GPS error evolution model sharing framework is proposed. The corresponding smart contracts for data storage and sharing are designed. Our proposal improves the accuracy of cooperative positioning by ensuring the security and reliability of cooperators and data.
- *AI-based Algorithm*: We propose a DNN-based error correction algorithm that runs on the edge server and can take advantage of the computing power and low latency characteristics of edge computing.
- *Collective Learning Strategy*: Cooperation between vehicles by sharing DNN models instead of positioning error data, which makes the results more adaptive to the current driving environment.
- *Validation*: The developed proposal is evaluated by extensive simulations. Simulation results validate the accuracy and security of our proposal in terms of positioning error prediction, error correction and data sharing process.

The remainder of the paper is organized as follows. The related work is presented in Section II. In Section III, we depict the system architecture and analyze the GPS error. Based on the system model, the deep learning algorithm for analyzing GPS error evolution is proposed in Section IV. In Section V, we present the vehicular blockchain structure and smart contracts. Extensive simulation results are discussed in Section VI, and conclusion is drawn in Section VII.

II. Related Work

In this section, we survey the existing works on the improvement of vehicle positioning accuracy and vehicular blockchain.

A. Positioning of Vehicles

There are two main types of methods for positioning of vehicles. One is a non-cooperative positioning method based on the vehicle's own body sensor such as radar, camera, and Lidar, and the other is a CP method [15], [16]. For non-cooperative positioning, in most cases, the positioning accuracy is relatively accurate. Chen *et al.* [17] propose a positioning algorithm based on machine vision that utilizes low-cost monocular cameras. In [18] and [19], Lidar is adopted to observe the surrounding environment and match the observation with a priori knew 3D point cloud map for estimating the position of the vehicle within the map. However, this sensor-based approach has the drawback of a line-of-sight characteristic and severely affected by the weather. The authors of [20] propose a sensor fusion-based vehicle positioning system by using GPS, camera and digital map, etc., which costs less. High definition (HD) map contains the accurate three-dimensional characterization of the road network (centimeter-level accuracy). However, the requirements of high accuracy and real-time also make the collecting and processing of large amounts of data become extremely challenging. At the same time, its huge amount of data consumption of communication and computing resources should not be underestimated.

For CP, it is through direct communication between several vehicles to improve their positioning accuracy [5]. The authors of [5] discuss modern and conventional CP systems in detail. One is based on the method of ranging, such as RSS, time of arrival (TOA), angle of arrival (AOA), and Time Difference-of-Arrival (TDOA), which estimate the relative position directly at the RF-signal level [16]. However, these measurements are challenged by some drawbacks varying from complexities of the time-synchronization, occupations of the high-bandwidth, to huge costs on the implementations. Kloeden *et al.* design a low-cost prototype system for vehicle self-localization using AOA technique [21], which can achieve lane positioning. In order to solve the problem of inaccurate positioning caused by nonline-of-sight (NLoS), a TOA-based positioning technology is proposed [22]. In addition to the ranging-based method, the other is to exchange positioning data, speed and other information between vehicles to improve positioning accuracy. Cooperative vehicle communication technology offers new opportunities for CP methods [9], [23]–[25]. The authors of [9] take use of DSRC for sharing motion states and physical measurements among vehicles to improve vehicle positioning accuracy. In [23] and [24], GPS position estimates and vehicle-to-vehicle (V2V) measurements are combined for positioning enhancement. The authors also explain the effect of noise on the GPS and received signal strength of V2V. Kakkavas *et al.* [25] investigate the performance of relative posting using V2V and the importance of AOA and TDOA for location estimation. A drawback of this type of method is that the exchange of information between vehicles is unreliable, which results in reduced awareness of the relative position of the own vehicle relative to the surrounding vehicle. Cooperators who are not fully trusted will also affect the accuracy of cooperative positioning. These inspire our work to improve positioning accuracy from the perspective of cooperators and cooperative data security.

There are also some methods based on smartphones, roadside unit (RSU), and other infrastructures. The former dramatically depends on the accuracy in real-time event data provided by smartphones. Both [11] and [26] use smartphones to improve GPS positioning accuracy. In [11], the sensor data of the smartphone is used to fuse the GPS data of the vehicle to realize the vehicle navigation when the GPS is not available. In order to solve the problem of insufficient accuracy of the current positioning technology, the authors [26] introduce a smartphone-based lane detection system, which uses the low-power inertial sensor data in the smartphone to provide an accurate estimate of the current lane position of vehicles. In [10], the RSE is fully utilized for lane-level positioning. Both GPS and RSS data are considered and high positioning accuracy is achieved by connected vehicle networks. A hybrid TDOA and AOA localization method in three-dimensional (3D) space are proposed in [27]. The principle of infrastructure-assistant methods are the same as the ranging method of CP, but the positioning is accurate when the vehicle needs to access multiple infrastructures at the same time. The focus of both [28] and [29] is on machine learning algorithms for the indoor positioning system. By contrast, the focus of this paper is on the use of vehicular blockchain technology to improve the security and reliability of cooperators and cooperative data for enhancing the positioning accuracy in outdoor vehicular environments. These work [28], [29] demonstrate the feasibility of applying machine learning algorithms for positioning.

B. Background of Blockchain Technology

Blockchain is a decentralized infrastructure widely used in emerging digital cryptocurrencies. It has the characteristics of decentralization, data not tampering, and trustworthiness [14]. According to the degree of centralization of the blockchain systems, it is mainly divided into three categories: public blockchain, consortium blockchain, and private blockchain [30]. The public blockchain allows any node to join the blockchain network and view information on the blockchain, such as Bitcoin [31] and Ethereum [32]. The consortium blockchain allows authorized nodes to join the network and can view information according to permissions, which is often used between organizations. In the private blockchain, all nodes in the network are in the hands of an organization or entity.

In the blockchain system, non-tamperable data structures, distributed network consensus mechanisms, and increasingly flexible smart contracts are representative innovations. The core problem of distributed networks is how to achieve consensus efficiently [14]. The consensus mechanisms of the current blockchain system are mainly the following: Practical Byzantine Fault Tolerance (PBFT), Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS) [33]–[35].

The smart contract of blockchain technology is the decentralized, information-sharing program code deployed on the blockchain. The parties who signed the contract agreed on the contract content and deployed it on the blockchain in the form of smart contracts, which can automatically execute the contract on behalf of the signatories without relying on any central organization [14], [36]. The operation of the smart contract is as follows. The smart contract encapsulates a number of predefined states, conversion rules, trigger conditions, and corresponding operations. After being signed by the parties, it is attached to the blockchain data in the form of program code, and is recorded after propagation and verification by the blockchain network. In the distributed ledger of each node, the blockchain can monitor the status of the entire smart contract in real-time and activate and execute the contract after confirming that certain trigger conditions are satisfied. Smart contracts encapsulate the complex behavior of nodes in a blockchain network, providing a convenient interface for building upper-layer applications based on blockchain technology.

C. Blockchain for Internet of Vehicles

At present, the blockchain technology has not been applied to vehicle positioning, so we investigate the state-of-the-art work in applying blockchain to the Internet of Vehicles.

Kang *et al.* [12] make use of blockchain technology to achieve secure data sharing in the Internet of Vehicles. In this paper, the vehicle can be a data provider or requester, using efficient contracts running on the blockchain for efficient data storage and sharing. In [37], a distributed and self-managed Vehicle Ad-hoc Networks (VANETs) architecture based on concept of Ethereum is proposed. Specifically, Ethereum's contract system is utilized to organize and self-manage various applications running on the RSU to provide different services to the vehicle. In order to protect the vehicle from security and privacy threats, a blockchain-based system is proposed to increase the security of the vehicular ecosystem [38]. Singh and Kim [39] use the blockchain technology applied in Bit-coin to propose an Intelligent Vehicle (IV) data sharing framework to build trust and reliability. In the trust environment based IV framework, the blockchain is used as the backbone of the IV data sharing. Yang *et al.* [40] propose a decentralized trust management system in vehicular networks based on blockchain technology to solve the problem that the vehicle is difficult to evaluate the credibility of the received message due to the untrusted environment.

TABLE I

MAIN NOTATIONS

Notation	Explanation
v_i	The ith vehicle in the system
$P_i =< p_i^e, p_i^n >$	The GPS position of vehicle i
$P_i' =< p_i'^e, p_i'^n >$	The position of vehicle i measured with landmark
$P_l =< p_l^e, p_l^n >$	The position of landmark l
d_{il}	Distance between vehicle i and landmark l
θ_{il}	Angle between vehicle i and landmark l
E_i	The GPS positioning error
E_s	The GPS systematic error
E_r	The GPS random error
$\parallel \Delta E_{ij} \parallel$	The relative error of vehicle i and j
$MECN_m$	The mth MECN in the system
a_i	vehicle's acceleration of vehicle i
dr_i	driving direction of vehicle i

In this system, the vehicle can use a Bayesian inference model to verify received messages from neighboring vehicles.

However, although these important works have made great efforts to improve the accuracy of vehicle positioning, there are still challenges in practical applications. In detail, in the current mixed traffic scene, GPS is still the main way of positioning most vehicles. It is meaningful to improve the positioning accuracy by analyzing the GPS positioning error characteristics to carry out corresponding research. In addition to the positioning method itself, data security is an important factor to consider in cooperative positioning methods because the cooperative vehicle and MECNs are not completely reliable. Therefore, this paper uses DNN and vehicular blockchain technology to improve the positioning accuracy of vehicles to take a further step in terms of accuracy and security.

III. SYSTEM MODEL

In this section, we describe the system architecture and the models that need to be used for positioning error correction. In addition, we analyze the possibility of sharing vehicle positioning error models for improving positioning accuracy. The main notices and abbreviations in this article are listed in Table I and Table II.

A. Positioning Scenarios and System Architecture

Generally speaking, there are two types of vehicles, *i.e.,* common vehicles and sensor-rich vehicles, driving on the road. Among them, common vehicles can only obtain position information through GPS. In addition to GPS, sensor-rich vehicles can also be assisted by other on-board sensors, such as camera and Lidar. Assume that the GPS receivers of the two types of vehicles are the same. All vehicles can access the MECNs to request service within the communication range. The MECNs have enough storage space and computing power, which store accurate location information of the infrastructure in the area (*e.g.,* landmark), as shown in Fig. 1.

TABLE II

MAIN ABBREVIATIONS

Abbreviation	Explanation
GPS	Global Position System
DNN	Deep Neural Network
RSU	Roadside Unit
MECN	Mobile Edge Computing Node
CP	Cooperative Positioning
VCP	Vehicle Cooperative Positioning
DSRC	Dedicated Short Range Communications
RSS	Received Signal Strength
IoT	Internet of Things
HD Map	High Definition Map
TOA	Time of Arrival
AOA	Angle of Arrival
TDOA	Time Difference of Arrival
NLoS	Nonline-of-Sight
V2V	Vehicleto-Vehicle
SCR	Smart Contract for Recording
PBFT	Practical Byzantine Fault Tolerance
SCS	Smart Contract for Sharing
IPFS	InterPlanetary File System
DPoS	Delegated Proof of Stake
PoW	Proof of Work
PoS	Proof of Stake

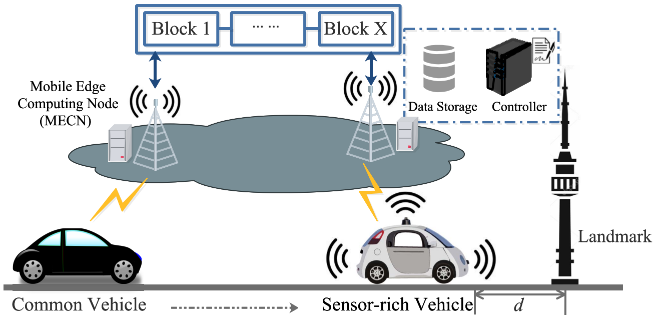


Fig. 1. System architecture of the GPS error sharing framework using vehicular blockchain.

When a sensor-rich vehicle approaches the landmark, the distance and angle between itself and the landmark can be obtained by on-board sensors. And the position of the landmark can be used to calculate the current position of the vehicle. For vehicle i, the GPS position is expressed as $P_i =< p_i^e, p_i^n >$, and the vehicle position $P_i' =< p_i'^e, p_i'^n >$ measured with landmark l can be expressed as:

$$\begin{cases} p_i'^e = p_l^e + d_{il} * \cos\theta_{il} \\ p_i'^n = p_l^n + d_{il} * \sin\theta_{il} \end{cases}, \tag{1}$$

where $P_l =< p_l^e, p_l^n >$ is the position of landmark l, d_{il} and θ_{il} are the distance and angle between vehicle i and landmark l, respectively.

Currently, on-board sensors have relative high range accuracy in most cases. Therefore, for vehicle i, we consider that P_i' is the accurate position information compared to P_i. The GPS positioning error E_i can be calculated as:

$$E_i = \|P_i - P_i'\|. \tag{2}$$

B. GPS Error Analysis

In practice, GPS measurements contain a variety of errors, which can be classified into three categories based on the source of the error: 1) errors associated with GPS satellites; 2) errors associated with signal propagation; 3) and receiver-related errors. Many studies and our previous work have shown that according to the nature of the error, these errors can be divided into two types: systematic error and random error [41], [42]. The GPS errors E can be decomposed into:

$$E = E_s + E_r, \tag{3}$$

where E_s and E_r are the systematic errors and random error, respectively. Particularly, the systematic error mainly includes satellite orbit error, satellite clock error, ionospheric delay, tropospheric delay, receiver clock error and receiver position error, providing the same error magnitude and direction for each GPS receiver within a certain range, about $50 - 200km$. The random error mainly includes multipath effect error and receiver noise. Usually, random error E_r is much smaller than the systematic error E_s [43], [44].

For vehicles i and j that are driving on the same road at similar times, their relative error $\| \Delta E_{ij} \|$ is satisfied:

$$\begin{aligned} \| \Delta E_{ij} \| &= \| (P_i - P_j) \| - \| (P_i' - P_j') \| \\ &\leq \| (P_i - P_j) - (P_i' - P_j') \| \\ &\leq \| E_i - E_j \|, \end{aligned} \tag{4}$$

where P_i, P_j, P_i', P_j', E_i and E_j are the GPS position, accurate position and GPS positioning error of vehicle i and vehicle j, respectively. According to (3) and (4), we can get:

$$\begin{aligned} \| \Delta E_{ij} \| &\leq \| (E_{si} + E_{ri}) - (E_{sj} + E_{rj}) \| \\ &\leq \| E_{si} - E_{sj} \| + \| E_{ri} - E_{rj} \|, \end{aligned} \tag{5}$$

where E_{si}, E_{sj}, E_{ri} and E_{rj} are systematic errors and random error of GPS positioning of vehicle i and vehicle j, respectively.

For the same type of GPS receiver, the positioning error radius is fixed. In addition, at similar moments, the satellite combinations that can be observed by each vehicle are basically the same. Therefore, when using the same satellite combination for positioning, the satellite position deviation caused by satellite clock difference, atmospheric delay, etc., is almost the same. Moreover, for two vehicles with similar positions, the difference in position between vehicles is negligible relative to the distance from the vehicle to the satellite. In this paper, vehicles for cooperative positioning and common vehicles are considered to be on the same road segment, that is, between two intersections, and generally less than 2 km. Therefore, in this case, the systematic error can be considered to be almost the same, that is, $E_{si} \approx E_{sj}$. Bring it into (5) to get:

$$\| \Delta E_{ij} \| \leq \| E_{ri} - E_{rj} \| . \tag{6}$$

Since random errors E_r are quite small compared to systematic errors E_s, $E_i \approx E_j$ can be considered in this case.

IV. Positioning Error Correction Algorithm

As outlined above, sensor-rich vehicles can utilize other sensors to correct their position through the location of the landmarks (*e.g.,* stop sign) and get the current GPS positioning error. However, due to the discontinuity of landmarks, accurate position information based on sensor ranging cannot be obtained without the landmark assistance. Therefore, the positioning error of these road sections can be compensated through prediction. In this section, we propose a deep learning-based algorithm for predicting error evolution that can be used to fill the error of these discontinuous road sections.

A. DNN for Positioning Error Prediction

DNN is highly fault-tolerant and self-learning, self-organizing and adaptive, capable of simulating complex nonlinear mapping. Many factors are affecting the accuracy of GPS positioning and the relationships are complicated. Therefore, the characteristics of the DNN and powerful nonlinear processing capabilities are well suited for predictive problems in traffic scenarios. Due to the powerful computing capacity of the edge server and improved data set, the DNN algorithm can run on the edge server for prediction when the landmark is unavailable.

In our work, a DNN algorithm is proposed to learn and compensate for the positioning error. When there is a landmark on the road, the positioning error obtained by the sensor-rich vehicles can be used as the training data of the neural network. When there is no landmark on the road, the positioning error can be predicted by using the GPS data of the vehicle and the vehicle motion state data.

1) Input Nodes: In this scenario, there are five input nodes, *i.e.*, vehicle's speed v_i, vehicle's acceleration a_i, driving direction dr_i, the GPS position p_i^e and p_i^n. Therefore, the input vector can be expressed as $[v_i \ a_i \ dr_i \ p_i^e \ p_i^n]$.

2) Output Nodes: There are two output nodes, which are the positioning error in the east direction ΔE_i^e and the positioning error in the north direction ΔE_i^n. The input vector can be expressed as $[\Delta E_i^e \ \Delta E_i^n]$.

3) Layers of Network: In general, a DNN is a neural network with multiple hidden layers between input and output [45]. When using the DNN for prediction, the more hidden layer, the higher the accuracy of the network, while the complexity and time-consuming will increase. There is currently no definitive formula to calculate the number of network layers, and it is necessary to continuously learn and train according to the data set and specific problems.

4) Hidden Layers Nodes: On the one hand, too few hidden layer nodes will make the network unable to have the necessary learning ability and information processing capability. Conversely, too many hidden layer nodes will not only greatly

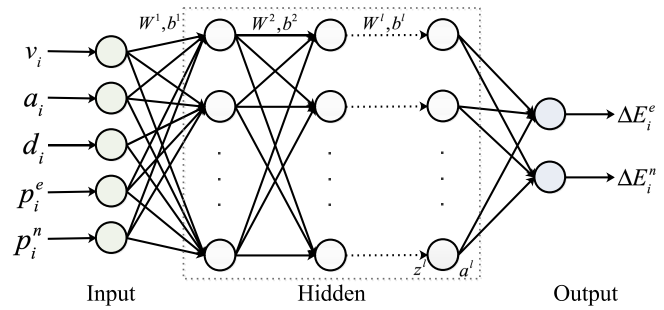


Fig. 2. The deep neural network for predicting error evolution.

increase the complexity of the network structure, but also reduce the speed of network learning, and the network is more likely to fall into local optimum in the learning process. In general, the number of hidden layer nodes are related to the number of training set samples, the number of input nodes, and the number of output nodes. The upper bound of hidden layers nodes that do not cause overfitting is [46]:

$$N_h = \frac{S}{(\lambda * (I + O))}, \tag{7}$$

where I and $O$are the number of input and output nodes, respectively, S means number of samples in the training data set, and λ is a scale factor, typically a constant within [2, 10].

According to Andrew's course [47], when the number of hidden layers is greater than 1, the number of nodes of each hidden layer should be the same. A typical diagram of DNN used in this work is shown in Fig. 2.

Let z^l, a^l and b^l represent the vector of all inputs, the vector of all outputs, and the vector of all offsets of the l'th layer, respectively. Thus, the relationship between z^l and a^{l-1} can be expressed as:

$$z^l = W^l * a^{l-1} + b^l, \tag{8}$$

where W^l represents the weight from a^{l-1} connected to z^l. We can also describe their relationship as a vector form:

$$\begin{aligned} a^l &= \sigma(z^l) \\ &= \sigma(W^l * a^{l-1} + b^l), \end{aligned} \tag{9}$$

where $\sigma(\cdot)$ is activation function.

B. DNN Training and Parameter Learning

The activation functions used by the hidden layers and output layer of the network are the ReLU function $\sigma(z) = max(z, 0)$ and Sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$, respectively. The loss function used in this article is mean square error (MSE), which is suitable for numerical prediction problems and can be calculated as follows:

$$f(\theta, x, y) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y} - y_i)^2, \tag{10}$$

and

$$\theta = \{W, b\}, \tag{11}$$

where x and y are the input and output vectors, respectively. θ is the parameter vector of the loss function. W and b are the linear relationship coefficient matrix and bias vector of each hidden layer and output layer. n is the number of samples, $\hat{y}$ and y_i represent the actual output and desired output of i'th sample, respectively.

Therefore, for each sample, $f = \frac{1}{2}\|\hat{y} - y\|_2^2$. Suppose the network has L layers, we can get $\hat{y} = a^L$. Combine (9) to solve the gradient of the output layer (*i.e.*, L'th layer) based on the loss function:

$$\frac{\partial f(\theta, x, y)}{\partial W^L} = \frac{\partial f}{\partial z^L}\frac{\partial z^L}{\partial W^L} = (a^L - y)(a^{L-1})^T \odot \sigma'(z^L), \tag{12}$$

and

$$\frac{\partial f(\theta, x, y)}{\partial b^L} = \frac{\partial f}{\partial z^L}\frac{\partial z^L}{\partial b^L} = (a^L - y) \odot \sigma'(z^L), \tag{13}$$

where $\odot$ means Hadamard product.

For hidden layers, we can obtain

$$\frac{\partial f(\theta, x, y)}{\partial z^l} = \frac{\partial f(\theta, x, y)}{\partial z^L}\frac{\partial z^L}{\partial z^{L-1}} \cdots \frac{\partial z^{l+1}}{\partial z^l}. \tag{14}$$

Combining (8) and (14), the W^l and b^l of the l'th layer are easy to derive:

$$\frac{\partial f(\theta, x, y)}{\partial W^l} = \frac{\partial f}{\partial z^l}\frac{\partial z^l}{\partial W^l} = \frac{\partial f}{\partial b^l}(a^{l-1})^T, \tag{15}$$

and

$$\frac{\partial f(\theta, x, y)}{\partial b^l} = \tfrac{\partial f}{\partial z^l}\tfrac{\partial z^l}{\partial b^l}. \tag{16}$$

According to (8) and (9), the relationship between z^l and z^{l+1} can be represented as

$$\begin{aligned} z^{l+1} &= W^{l+1} * a^l + b^{l+1} \\ &= W^{l+1} * \sigma(z^l) + b^{l+1}. \end{aligned} \tag{17}$$

Therefore, we can obtain

$$\frac{\partial f(\theta, x, y)}{\partial b^l} = (W^{l+1})^T \frac{\partial f}{\partial b^{l+1}} \odot \sigma'(z^l), \tag{18}$$

and

$$\frac{\partial f(\theta, x, y)}{\partial W^l} = (W^{l+1})^T \frac{\partial f}{\partial b^{l+1}} \odot \sigma'(z^l)(a^{l-1})^T. \tag{19}$$

Parameter adjustment according to the following equations during the learning process:

$$W^l = W^l - \eta * \frac{\partial f}{\partial b^l}(a^{l-1})^T \tag{20}$$

and

$$b^l = b^l - \eta * \frac{\partial f}{\partial b^l}, \tag{21}$$

where η is learning rate. The detailed prediction process of DNN is shown in Algorithm 1.

Algorithm 1 Algorithm of Training DNN

```
Input: The training samples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, number of layers $L$, maximum number of iterations $I_{max}$, iteration step size $s$, and stop iteration threshold $\varepsilon$;
Output: The linear relationship coefficient matrix $W$ and bias vector $b$ of each hidden layer and output layer;
Initialize the values of $W$ and $b$ as random values;
for $iter = 1 : I_{max}$ do
    for $i = 1 : n$ do
        Calculate the forward process of each data and retain the value of the activation function for each layer $a_i^l$;
        Calculate the loss value by loss function $f(\theta, x, y)$;
        Calculate the gradient of the output layer $\frac{\partial f}{\partial z_i^L} = \frac{\partial f}{\partial a_i^L} \odot \sigma'(z_i^L)$;
        for $l = 2 : L$ do
            Calculate $\frac{\partial f}{\partial z^l} = (W_i^{l+1})^T \frac{\partial f}{\partial z_i^{l+1}} \odot \sigma'(z_i^l)$;
        end for
        for $l = 2 : L$ do
            Update $W^l$ and $b^l$ by
            $W^l = W^l - \eta \sum_{i=1}^n \frac{\partial f}{\partial z_i^l} (a_i^{l-1})^T$;
            $b^l = b^l - \eta \sum_{i=1}^n \frac{\partial f}{\partial z_i^l}$;
        end for
        if Changes in $W$ and $b$ less than $\varepsilon$ then
            Stop training.
        end if
    end for
end for
```

C. Positioning Error Correction

In this paper, other vehicles, especially common vehicles, can use the error evolution of blockchain in sensor-rich vehicles in the near future to correct their positioning errors. This subsection proposes a strategy based on transfer learning to perform positioning error correction.

For sensor-rich vehicles, the DNN-based prediction model and the resulting error evolution results are stored on the blockchain. The specific data storage and sharing process based on vehicular blockchain will be introduced in Section V. For the common vehicle i, after obtaining the model parameters and positioning errors of other J vehicles, it first judges whether or not to use the information. By comparing the positioning error of vehicle j, $j \in J$ (*i.e.*, $\Delta E_j = (\Delta E_j^e, \Delta E_j^n)$) and average error other J vehicles $\Delta E_{ave} = (\frac{1}{J} \sum_{j=1}^J \Delta E_j^e, \frac{1}{J} \sum_{j=1}^J \Delta E_j^n)$. For each vehicle j, if $||\Delta E_j, \Delta E_{ave}|| \geq \delta$, where δ is a certain threshold, it is considered to provide erroneous data and the data is not used. After culling the erroneous data, the model parameters and positioning errors of other vehicles are averaged separately. Then, vehicle i adds a hidden layer to the neural network and continues training, thereby obtaining error evolution, which can be used to correct its positioning.

When vehicle i uses data stored in the blockchain by other vehicles, it also needs to pay the corresponding fee. Vehicles that provide multiple error information will be financially penalized or removed from the vehicular blockchain. This will incent more vehicles to provide information and ensure the accuracy of the data.

V. Blockchain Construction

In this section, we introduce the blockchain-based positioning error evolution model sharing method, the system architecture as well as the running smart contracts.

A. Vehicular Blockchain

The sensor-rich vehicle can get its own positioning error evolution about the current road segment. Through the previous analysis, this information can be shared with other vehicles in similar locations in the near future. In this paper, error correction can be done in two ways: One is to obtain the positioning error evolution of sensor-rich vehicles and directly combine the GPS positioning information of the vehicle to obtain the corrected position. The other is to predict the positioning error of a common vehicle through the error prediction model obtained by sensor-rich vehicles, thereby performing error correction. Considering the timeliness of information and the validity of location, MECNs can serve as a storage and sharing center for information. However, vehicles may provide incorrect data for various reasons (even selfishness), and MECNs may be attacked rather than fully reliable. Therefore, blockchain technology can be used to achieve secure and reliable storage and sharing of information.

Since GPS information and positioning error information can only be applied to vehicles within the area, we adopt a consortium blockchain based on vDLT blockchain platform, which is faster than the traditional public blockchain and the cost is greatly reduced. The vehicular blockchain system architecture is shown in Fig. 1, which is consisted of vehicle nodes, MECNs. The consensus process is controlled by pre-selected nodes (MECNs). In this scenario, sensor-rich vehicles are mainly data providers, and common vehicles are mainly data requesters. The data provider only encrypts and transmits the data and transmits the data as a blockchain transaction to the nearby MECN for a corresponding reward. In particular, MECNs are not only used for data storage but also perform the smart contract as controllers during data sharing.

B. Smart Contracts for Information Sharing

Smart contracts based on blockchain technology can not only take advantage of the cost-efficiency of smart contracts but also avoid the interference of malicious behaviors on the normal execution of contracts, so that information storage and sharing based on smart contracts can run efficiently and safely. In addition, since the block generation period (0.5s) is inconsistent with the time scale of the vehicle information broadcast cycle (200ms), smart contracts can also be used to solve this problem.

The newly concluded contract collection will spread to the entire network in the form of blocks. As shown in Fig. 3, each block contains the following information: hash of the current block, hash of the previous block, the timestamp when the

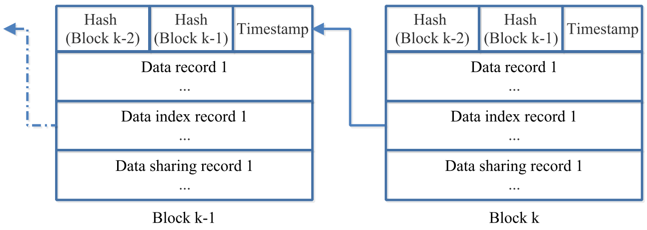


Fig. 3. Smart contract recorded and data recorded in blocks of a blockchain.

consensus was reached, and transactions records set. A block mainly contains three types of records, that is, data storage records, data index records, and data sharing records.

(1) *Smart Contract for Recording (SCR)*: Before performing a smart contract for data sharing, the data provider needs to store the encrypted data to the system. We design a storage solution based on the InterPlanetary File System (IPFS) [48]. IPFS is an emerging standard for storing content addressable files. Content-addressed storage is a mechanism for storing information that can be retrieved based on its content rather than its location. The SCR is deployed on the blockchain, in which all files are linked to IPFS files. The file link content mainly includes the local home directory path, the file name (i.e. hash of the data), and the file size:

$$File\ Link = (/home/||file\ name||file\ size)$$

The request message for recording data from vehicle v_i to a nearby MECN $MECN_m$, which contains the identifier K_i^{pu}, the hash of the latest ined block $h(Block_{t-1})$, the file link $File\ Link_i$ and timestamp, that is:

$$D\_Req_{v_i \to MECN_m} : \left\{ \begin{array}{c} K_i^{pu} \\ h(Block_{t-1}) \\ File\ Link_i \\ timestamp \end{array} \right\}_{K_i^v} .$$

Particularly, the signature of requester's private key K_i^v is used to guarantee the truthfulness of the requested information. Then v_i will send the encrypted shared data to $MECN_m$. After the $MECN_m$ receives the request message and data, it will verify that the hash of file content and file size are correct. Once the verification is successful, $MECN_m$ will wrap the received request into the new record message and add other information such as identifier K_j^{pu}, the file link $File\ Link_i$ and timestamp. Similarly, the signature K_j^{MECN} are meant to ensure truthfulness and allow anyone to verify that the information has been tampered with. The record message can be expressed as:

$$D\_Rec_{v_i \to MECN_m} : \left\{ \begin{array}{c} D\_Req_{v_i \to MECN_m} \\ K_m^{pu} \\ File\ Link_i \\ timestamp \end{array} \right\}_{K_m^{MECN}} .$$

In case the file is correctly verified, $MECN_m$ broadcasts the data record to other MECNs acting as verification nodes to verify the signature to ensure its validity. After the consensus on the event is achieved, the smart contract will be successfully executed and synchronize to the blockchain, indicating that the file has been saved successfully, as shown in Fig. 4.

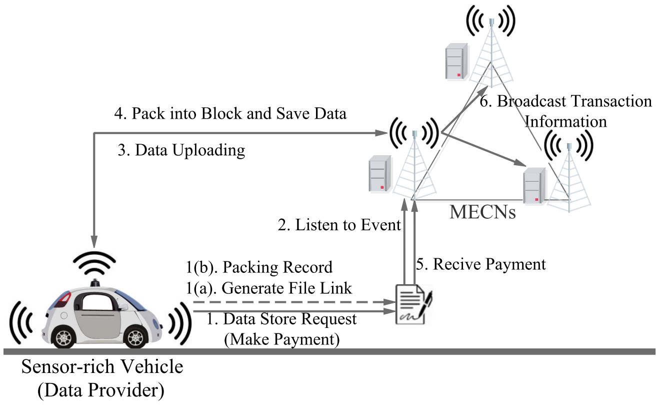


Fig. 4. Smart contract process for data storage.

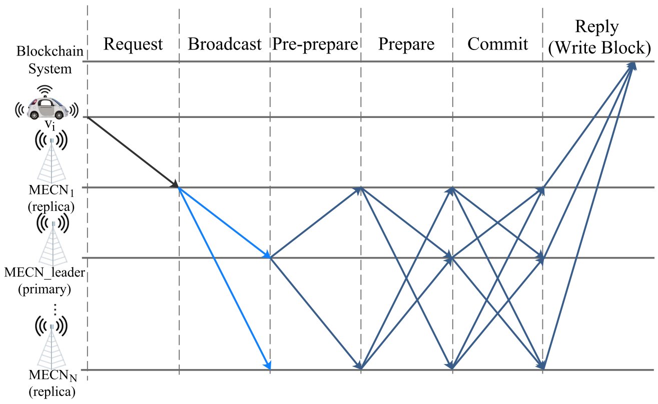


Fig. 5. The consensus process for SCR.

The consensus process for SCR is based on PBFT [34], which has been widely used in many practical scenarios. In addition, the consensus nodes are obtained by random calculation, and at least $3f+1$ MECNs are randomly selected, where $f \geq 1$. The MECN calculates whether it is selected according to its private key, and broadcasts the result to other MECNs. This method ensures that the nodes selected in each round are random and different, which increases security. The consensus process mainly has the following steps as depicted in Fig. 5 and Fig. 6:

- *Broadcast*: v_i sends request to $MECN_m$ (consensus nodes or accounting nodes). For accounting nodes, it needs to forward the transaction to the consensus nodes, then the consensus nodes broadcast the transaction to other consensus nodes. $MECN_m$ sends broadcast the event in Fig. 5 and Fig. 6.
- *Pre-Prepare*: The primary (MECN-leader) broadcasts the Pre-Prepare message. The MECN_leader (primary node) will validate the transaction immediately after packing the transactions, and then include the validation result into the Pre-Prepare message for the whole network broadcast. Pre-Prepare message contains both the ordered Transaction information and block validation result.

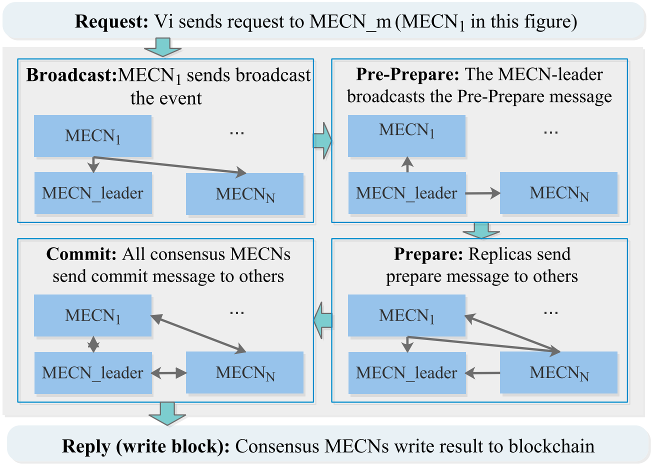

Request: Vi sends request to MECN_m (MECN$_1$ in this figure)
Broadcast:MECN$_1$ sends broadcast the event
MECN$_1$
...
MECN_leader
MECN$_N$
Pre-Prepare: The MECN-leader broadcasts the Pre-Prepare message
MECN$_1$
...
MECN_leader
MECN$_N$
Commit: All consensus MECNs send commit message to others
MECN$_1$
...
MECN_leader
MECN$_N$
Prepare: Replicas send prepare message to others
MECN$_1$
...
MECN_leader
MECN$_N$
Reply (write block): Consensus MECNs write result to blockchain

Fig. 6. The consensus mechanisms for SCR.

- *Prepare*: Replicas send prepare message to others. After receiving a Pre-Prepare message from MECN_leader, a replica checks the validity of the message. When the replica discovers that the message is from the MECN_leader and receives it for the first time, the replica will broadcast a Prepare message indicating consistency with the result of the MECN_leader.
- *Commit*: All consensus MECNs send commit message to others. After receiving the Prepare message, a replica verifies and compares the verification result with the Pre-Prepare message from the MECN_leader. When a Prepare message is found to pass the feedback agreed by $2f$ replicas, it indicates that it enters the commit state. The replicas broadcast a Commit message to other replicas.
- *Reply (write block)*: Consensus MECNs write the result to blockchain. A replica will receive Commit information from other nodes. When it finds that the information has been approved by $2f + 1$ (including itself), it considers the transaction to reach a consensus and attempts to send the execution result to the block producer.

During a period of time, each MECN receives a data storage request from different vehicles within communication range and generates a corresponding data record. These records will be gathered by the leader. And after the transaction is verified, it is packaged into blocks and broadcasted to other pre-selected MECNs on the blockchain for verification, which decide whether to add the block to the end of the blockchain or not. After the consensus is achieved, the data records contained in the block will be persistent. More details about the consensus process for blockchain are introduced in Section V. C. Other MECNs will save data synchronously through IPFS network.

(2) *Data index recording*: When υ_i uploads its data, it also generates a corresponding data index for other vehicles to quickly filter data, as shown below:

$$Data\ index_i = (File\ Name||Description|| \\ Data\ Provider||Timestamp).$$

Similar to data recording, υ_i sends the request for recording the data index to the nearby MECN $MECN_m$, that is

$$DI\_Req_{\upsilon_i \to MECN_m} : \left\{\begin{array}{c} K_i^{pu} \\ h(Block_{t-1}) \\ Data\ index_i \\ timestamp \end{array}\right\}_{K_i^{\upsilon}}.$$

After verification, $MECN_m$ will wrap the received request into a new data index record message

$$DI\_Rec_{\upsilon_i \to MECN_m} : \left\{\begin{array}{c} DI\_Req_{\upsilon_i \to MECN_m} \\ K_m^{pu} \\ Data\ index_i \\ timestamp \end{array}\right\}_{K_m^{MECN}}.$$

Similarly, during the period in which the next block is generated, each MECN obtains a data index request from the sensor-rich vehicles within the communication coverage. The process of adding data index records to the blockchain is similar to data recording.

(3) *Smart Contract for Sharing (SCS)*: The common vehicle (data requester) υ_j gets the latest block from the blockchain and selects the data that needs to be requested to be shared according to its own situation and data description. When υ_j issues a request for location information sharing, the SCS is triggered. Assuming υ_j selects the data provided by υ_i, a request message to MECN $MECN_m$ will be sent and make a payment.

$$DS\_Req_{\upsilon_j \to MECN_m} : \left\{\begin{array}{c} K_j^{pu} \\ h(Block_{t-1}) \\ File\ Name_i \\ timestamp \end{array}\right\}_{K_j^{\upsilon}}.$$

After the request has listened, the $MECN_m$ first verifies the certificate of υ_j and the existence of requested data through IPFS. The $MECN_m$ then encrypts the data using the requester's public key and stores it as a single-access link as an access authorization to the requester and secure protection of the data. When the data requester uses the link to access the data to start the download, the link will be invalid. Then, a smart contract sends payment to the data provider υ_i immediately, as shown in Fig. 7. The consensus process for SCS is also adopted to PBFT, similar to SCR. Then the transaction record of the data sharing event will be recorded on the blockchain, which is similar to data recording and data index recording.

C. Consensus Process for Blockchain

During a certain period, the leader will aggregate data records, data index records, and data sharing records as well as package them into blocks. This block will be added to the end of the blockchain after the consensus is achieved. The consensus process is adopted to BFT-DPoS consensus algorithm, which can reduce the block generation interval to 0.5s and make the block confirmation speed increase significantly [35]. The DPoS is a consensus mechanism for authorizing equity. Compared to the PoW and PoS mechanisms of the traditional

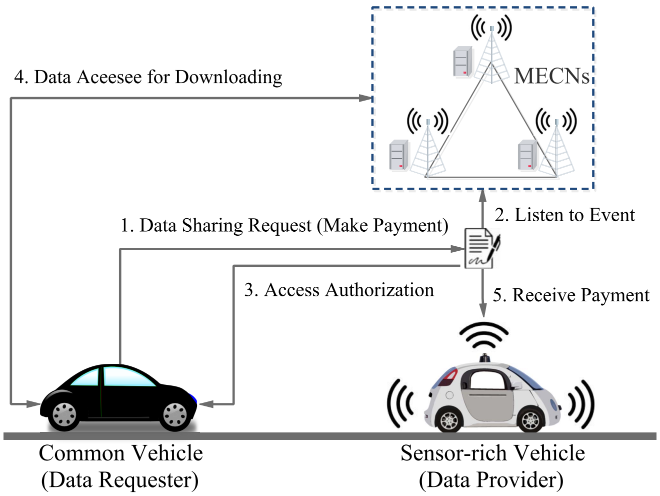


Fig. 7. Smart contract process for data sharing.

public blockchain, DPoS does not waste power resources to compete for billing rights.

It votes to select a number of block producers, ensuring the normal operation of the entire network. The MECNs on the blockchain compete for block producers based on the number of vehicle coins they hold, trust values, and computing power. During the voting process, every MECN on the block uses the rights in their hands to support trusted candidates and then determines a certain number of producers based on the overall voting situation, recorded as $MECN_P(n)$, $n = 1, 2, \dots, N$. Determine the production sequence for the $MECN_P$ and take turns to produce new blocks. Only one of the $MECN_P$ is allocated at a time for block production. Each $MECN_P(n)$ generates one block every 0.5 seconds, and a total of 6 blocks are generated. If a producer misses producing block, it will be skipped and the producer will be deleted. Each time a production cycle is completed, the producers will be re-selected. Let the block producer become the primary node. After the block is generated, the primary node broadcasts the block to the remaining $N-1$ nodes at the same time and obtains the verification feedback from each node. If more than $2/3$ of the nodes agree with the verification, the block is persistent and irreversible.

VI. Simulation Results and Discussions

In this section, the performance of our proposed algorithm is evaluated by a large number of simulations, which are divided into three aspects. First, we evaluated the accuracy and timeliness of the positioning error prediction algorithm. Next, the error correction accuracy of the proposed method is evaluated by comparison with other positioning methods. Finally, by evaluating the robustness and security of the algorithm, the impact of false data on the performance of CP methods is reflected. In the simulation, it is assumed that all vehicles are equipped with the same GPS positioning performance.

A. Positioning Error Prediction

First, we verify the performance of the positioning error prediction algorithm in terms of accuracy and timeliness. We use the autonomous vehicle based on the comma.ai platform and an On-Board Unit (OBU) to obtain vehicle location, speed, braking status and other parameters [49]. The speed of vehicle is 30 to $50km/h$. Vehicle acceleration is -4 to $4m/s^2$. The travel direction of the vehicle is east or north. Each road section is approximately $1km$. In particular, we analyze the location data to obtain the current GPS systematic error as well as random error range and use it as simulation data. According to the result of the error analysis, the systematic error E_s is about $20m$ and the random error obeys Gaussian distribution $E_r \sim N(0, 4.63)$. Detailed parameter settings for DNN are listed in Table III.

TABLE III
DNN Settings

Input nodes	$[v_i\ a_i\ d_i\ p_i^e\ p_i^n]$
Output nodes	$[\Delta E_i^e\ \Delta E_i^n]$
Hidden layer	4, 5
Number of nodes for each hidden layer	20
Loss function	MSE
Learning rate	0.005
Optimization Algorithm	Stochastic gradient descent
Early stopping rule	Loss in generality

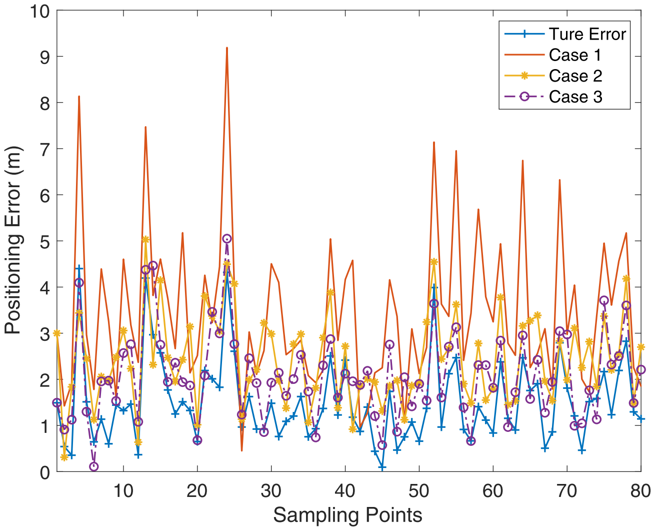


Fig. 8. Comparison of positioning errors between different positioning methods.

We divide the error evolution model sharing into the following three cases: 1) common vehicle i directly uses the positioning error shared by sensor-rich vehicles to correct its position (case 1); 2) common vehicle i retrains DNN using data shared by sensor-rich vehicles (case 2); 3) common vehicle i directly use sensor-rich vehicles to share the prediction model to add a hidden layer and continue training (case 3). In addition, the sensor-rich vehicles that perform the positioning error information sharing are set to 3. The threshold δ introduced in Section IV.C. is set to $3m$.

For the sake of display, we extracted the first 80 samples. It can be seen from Fig. 8, among these three cases,

TABLE IV
ERROR CORRECTION PERFORMANCE EVALUATION OF DIFFERENT CASES

	Case 1	Case 2	Case 3
Maximum (m)	4.87	2.33	1.71
Minimum (m)	0.83	0.12	0.005
Mean (m)	1.88	0.92	0.53
MSE (m^2)	5.04	1.24	0.59

the positioning error of case 1 is larger, while cases 2 and 3 have similar positioning performance. Although the feasibility of the vehicle to improve the positioning accuracy through positioning error sharing is obtained through analysis in Section III, there are still differences between vehicles. Therefore, when these error information are directly used to improve the positioning accuracy of other vehicles, performance is affected. Cases 2 and 3 both re-estimate the error evolution based on their own data and information shared by sensor-rich vehicles, which is more suitable for their own situation. In addition, we have listed some special results in Table. IV, which represent the maximum, minimum, average and mean square error (MSE) between the positioning error and the true positioning error obtained in the three cases. Similar to the results in Fig. 8, the difference between the positioning error of case 1 and the true value is the largest, with a maximum difference of 4.87 m. And the error between case 2 and case 3 and the true value are within 2.5 m. Similarly, it can be seen from the table that the MSE in each case is 5.04, 1.24 and 0.53, respectively, indicating that the prediction model in case 3 describes the experimental data with better accuracy. However, the minimum column data indicates that the prediction results in all three cases are very similar to the real values, indicating that it is feasible to improve the vehicle positioning accuracy by means of error sharing.

We further evaluate the timeliness of case 2 and case 3 under the same conditions. Learning loss of case2 and case3 are depicted in Fig. 9. It can be seen that learning loss of case 3 drops faster than case 2, indicating that case 3 is more efficient in learning. The reason is that for case 2, common vehicle i needs to retrain DNN based on data shared by itself and sensor-rich vehicles. For case 3, the common vehicle i adds a layer of continuous training on top of the other vehicle-trained network architecture. Since the previous network model has been trained and the network parameters have been determined, the learning loss has to drop much faster, which is more time-efficient.

B. Positioning Error Correction

Subsequently, we study the positioning performances of different positioning methods, such as GPS, vehicular cooperative positioning (VCP), RSU-based ranging positioning (RRP) and the proposed error model sharing (EMS) method. The results are shown in Fig. 10 and Table V. In this case, the GPS maximum positioning error is close to 27m. It can be seen that EMS, VCP, and BRP all improve the accuracy of positioning error to some extent. Among these comparison algorithms, the worst performance is VCP, which has a maximum positioning error of 9.56m. The underlying reason is that the positioning of the vehicle itself is not accurate, so there is an error in the position information provided during cooperative positioning. In addition, when information is exchanged, V2V communication has delays, packet loss, etc., resulting in a decrease in positioning performance. By contrast, the positioning error of EMS and RRP is the smallest, because RSU and landmark have accurate position information. The maximum positioning error of EMS and RRP are 4.15m and 6.57m. Although in more than half of the cases, the positioning error of EMS, VCP, and RRP is within 3m. However, VCP and RRP have a larger error range than EMS. In addition, the RRP method requires at least two RSUs for positioning accuracy to be guaranteed.

In order to further analyze the different positioning methods to improve the GPS positioning error, we decompose the positioning error into two directions: the east and the north.

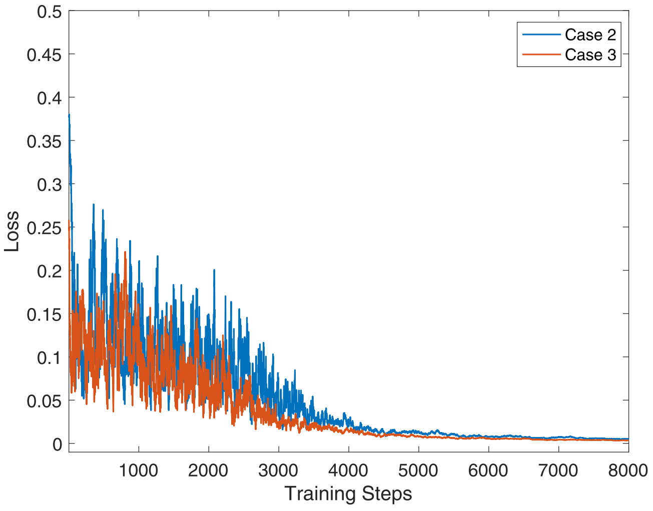


Fig. 9. Training curves tracking the learning loss of case 2 and case 3.

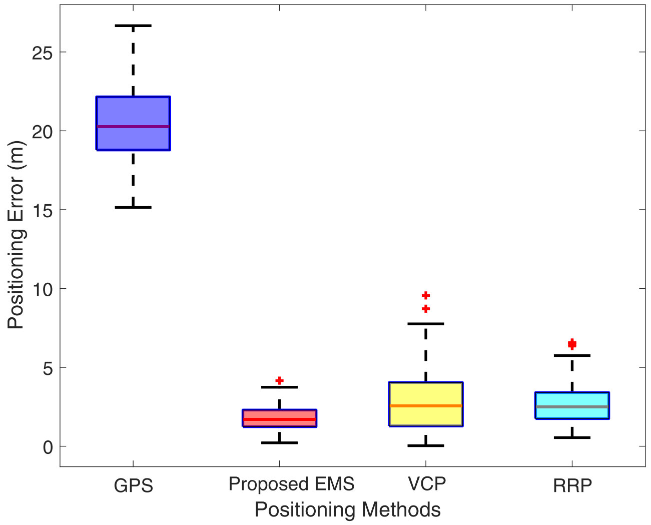
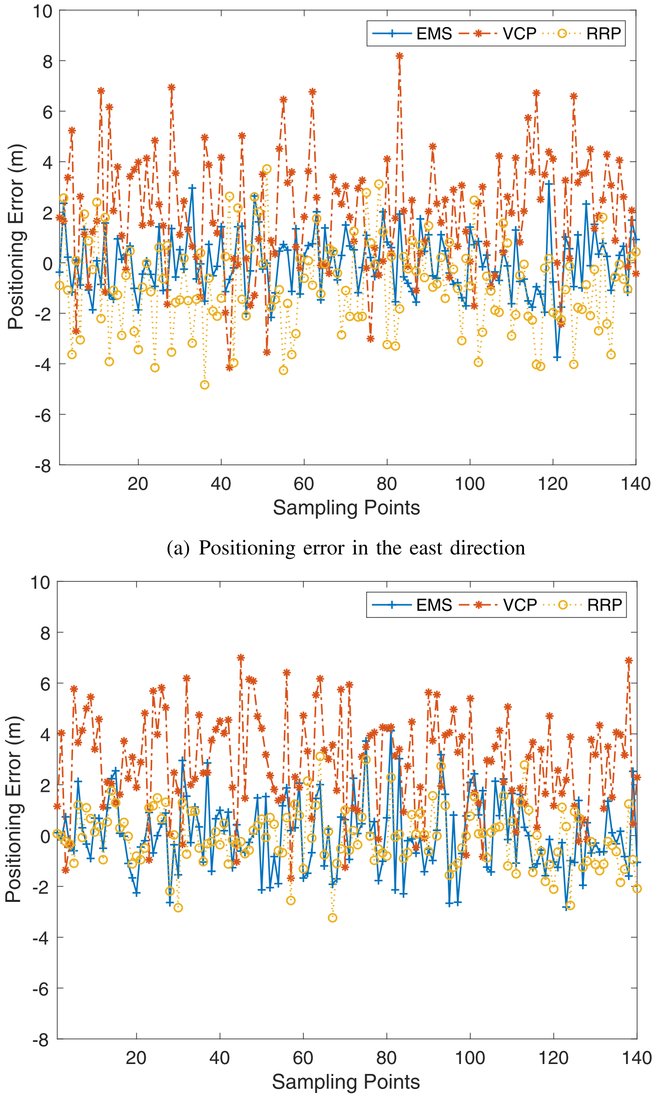


Fig. 10. Comparison of positioning errors between different positioning methods.



(a) Positioning error in the east direction

(b) Positioning error in the north direction

Fig. 11. Positioning Error of different methods.

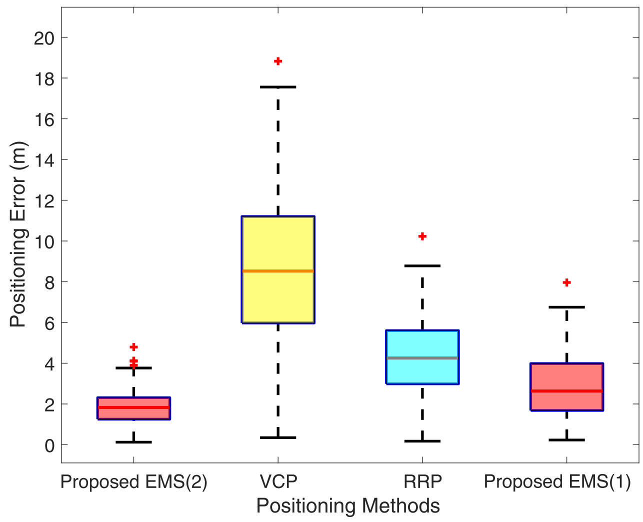


Fig. 12. Comparison of positioning errors between different positioning methods with fewer cooperators.

TABLE V
ACCURACY EVALUATION OF DIFFERENT METHODS

	GPS	Proposed EMS	VCP	RRP
Maximum Error (m)	27	4.15	9.56	6.57
Median Error (m)	20.22	2.49	2.89	2.65
Maximum Error in Single Direction (m)	-	3.58 (north)	8.07 (east)	6.00 (east)

TABLE VI
ROBUSTNESS AND SECURITY EVALUATION OF DIFFERENT METHODS

		Proposed EMS	VCP	RRP
Robustness	Maximum Error (m)	4.15 (EMS 2) 6.79 (EMS 1)	18.96	10.23
	Median Error (m)	1.82 (EMS 2) 2.79 (EMS 1)	8.2	4.15
Security	Maximum Error (m)	7.06 (case a) 4.08 (case b)	10.15 (a) 14.93 (b)	7.79 (a) 10.54 (b)
	Median Error (m)	3.76 (case a) 1.75 (case b)	6.75 (a) 10.69 (b)	3.98 (a) 7.39 (b)

As can be seen from the Fig. 11 and Table V, the positioning errors of the EMS and RRP methods in both directions are shifted in the positive and negative directions with $y = 0$ as the center, while the error of the VCP method is not centered on $y = 0$. In addition, the maximum positioning errors of EMS, VCP, and RRP in a single direction are $3.58m$ (north), $8.07m$ (east) and $6.00m$ (east), respectively. The result verifies that the vehicle GPS positioning error can be decomposed into systematic error and random error, wherein the systematic error provides the same error magnitude and direction for each GPS receiver within a certain range, as we analyzed in Section III. The VCP method does not eliminate the systematic error. There is still a large positioning error. Both EMS and RRP methods can better eliminate GPS systematic errors, resulting in higher positioning accuracy.

C. Robustness and Security Analysis

In order to evaluate the robustness of the proposed method, we set the vehicles for cooperative positioning in the VCP and RSUs in RRP to 2. For the EMS method, set the vehicles for data sharing to 2 and 1, respectively. The VCP method and RRP method cannot perform positioning error correction if only one vehicle or RSU available. The results are depicted in Fig. 12 and Table VI. EMS (2) indicates the use of information shared by two sensor-rich vehicles, while EMS (1) indicates the use of one. Combined with Fig. 10, the performance of the error correction for each method may decrease as the number of vehicles or RSUs used for cooperation is reduced. For VCP, when the cooperative positioning vehicle changes from 3 to 2, the maximum positioning error increases by $10.89m$. Similarly, the maximum positioning error of RRP is increased by $3.66m$. In contrast,

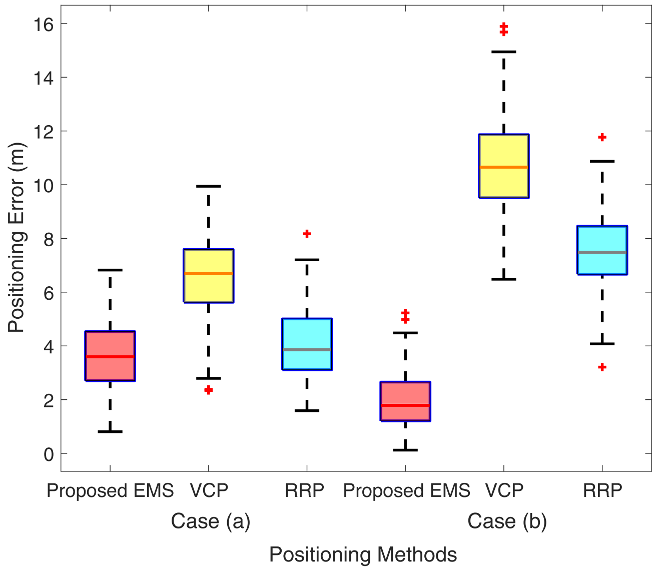


Fig. 13. Comparison of positioning errors between different positioning methods with malicious node.

EMS still maintains good performance. When only two sensor-rich vehicle data are available, the maximum positioning error increases from $4.15m$ to $6.79m$. In addition, even if only one sensor-rich vehicle data is available, the positioning error correction performance is still higher than VCP (three cooperative vehicles) and RRP (two RSUs). It shows that the EMS method is more adaptable to the environment, and can still achieve acceptable error correction results without sufficient sensor-rich vehicles.

Next, for a situation where there are three vehicles or three RSUs co-located, we randomly set one of them as a malicious node. Malicious nodes provide false location information, which affects the correction of positioning errors. We evaluate the performance of the three positioning methods in this case, and the results are shown in Fig. 13 and Table VI. Case (a) represents that the false data provided by the malicious node is slightly different from the real data (*i.e.*, positioning error $< \delta$), and case (b) represents that the data error is large. First, false data has a large impact on the performance of VCP and RRP, and the positioning error increases with the increase of false data error. For the EMS algorithm, when the false data difference is small, the common vehicle will still accept the positioning error correction, thus increasing the positioning error. When the false data difference is large, the data will be eliminated without affecting the final positioning accuracy. In addition, due to the nature of the blockchain, the data written to the blockchain cannot be tampered with. Therefore, for the false data provider, it can be directly removed from the blockchain system, thus ensuring the accuracy and security of the data. Therefore, the positioning error correction performance is similar to that of EMS (2) in Fig. 12.

VII. Conclusion

In this paper, a vehicular blockchain-based positioning error correction framework has been proposed for securely and efficiently improving the GPS positioning accuracy of vehicles. In particular, by analyzing the GPS positioning error characteristics, the feasibility of using sensor-rich vehicles to compensate for the positioning accuracy of common vehicles is obtained. Next, for the landmark discontinuous scenario, the DNN-based error correction algorithm has been proposed to obtain the positioning error evolution. Then, we have designed a vehicular blockchain based on vDLT platform for secure and efficient information storage and sharing. In addition, by fully exploiting the characteristics of blockchain and edge computing, smart contracts for data recording and sharing have been developed, which are running on the vehicular blockchain. In the proposed framework, the MECNs not only act as storage units, but also perform smart contracts and perform DNN operations. Extensive simulations have exhibited the effectiveness and accuracy of our proposed method compared with the traditional method. For many error prediction models stored on blockchain, how to choose the best provider to achieve higher positioning accuracy is one of the research priorities of future work. In addition, considering the impact of multipath in urban environments, how to eliminate random errors is another problem that needs to be studied.

References

[1] S. Demetriou, P. Jain, and K. Kimx, "CoDrive: Improving automobile positioning via collaborative driving," in *Proc. Int. Conf. IEEE Comput. Commun. (IEEE INFOCOM)*, Apr. 2018, pp. 1–9.

[2] L. Wei, C. Cappelle, and Y. R. Wei, "Camera/laser/GPS fusion method for vehicle positioning under extended NIS-based sensor validation," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 11, pp. 3110–3122, Nov. 2013.

[3] M. Sefati, M. Daum, B. Sondermann, K. D. Kreiskother, and A. Kampker, "Improving vehicle localization using semantic and pole-like landmarks," in *Proc. Int. Conf. Intell. Veh. Symp. (IV)*, 2017, doi: 10.1109/IVS.2017.7995692.

[4] G. De Angelis, G. Baruffa, and S. Cacopardi, "GNSS/cellular hybrid positioning system for mobile users in urban scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 313–321, Mar. 2013.

[5] N. Alam and A. G. Dempster, "Cooperative positioning for vehicular networks: Facts and future," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1708–1717, Dec. 2013.

[6] M. Yassin and E. Rachid, "A survey of positioning techniques and location based services in wireless networks," in *Proc. IEEE Int. Conf. Signal Process. Informat. Commun. Energy Syst.*, Feb. 2015, pp. 1–5.

[7] M. Ke, Y. Xu, A. Anpalagan, D. Liu, and Y. Zhang, "Distributed TOA-based positioning in wireless sensor networks: A potential game approach," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 316–319, Feb. 2018.

[8] V. Pasku, A. De Angelis, M. Dionigi, A. Moschitta, G. De Angelis, and P. Carbone, "Analysis of nonideal effects and performance in magnetic positioning systems," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 12, pp. 2816–2827, Dec. 2016.

[9] Y. Wang, X. Duan, D. Tian, M. Chen, and X. Zhang, "A DSRC-based vehicular positioning enhancement using a distributed multiple-model Kalman filter," *IEEE Access*, vol. 4, pp. 8338–8350, 2016.

[10] J. Li, J. Gao, H. Zhang, and T. Z. Qiu, "RSE-assisted lane-level positioning method for a connected vehicle environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2644–2656, Jul. 2019, doi: 10.1109/TITS.2018.2870713.

[11] O. Walter, J. Schmalenstroeer, A. Engler, and R. Haeb-Umbach, "Smartphone-based sensor fusion for improved vehicular navigation," in *Proc. 10th IEEE Workshop Positioning Navigat. Commun.*, Mar. 2013, pp. 1–6.

[12] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019, doi: 10.1109/JIOT.2018.2875542.

[13] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial Internet of Things: A dueling deep Q-learning approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4627–4639, Jun. 2019, doi: 10.1109/JIOT.2018.2871394.

[14] J. Xie *et al.*, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019, doi: 10.1109/COMST.2019.2899617.

[15] F. P. Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," *Sensors*, vol. 17, no. 271, pp. 1–27, Feb. 2017, doi: 10.3390/s17020271.

[16] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, Apr. 2018, doi: 10.1109/JIOT.2018.2812300.

[17] K.-W. Chen *et al.*, "Vision-based positioning for Internet-of-vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 364–376, Feb. 2017.

[18] E. Javanmardi, Y. Gu, M. Javanmardi, and S. Kamijo, "Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area," *IATSS Res.*, vol. 43, no. 1, pp. 1–13, Apr. 2019.

[19] A. Y. Hata and D. F. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer LIDAR," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 420–429, Feb. 2016.

[20] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor fusion-based low-cost vehicle localization system for complex urban environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1078–1086, May 2017.

[21] H. Kloeden, D. Schwarz, E. M. Biebl, and R. H. Rasshofer, "Vehicle localization using cooperative RF-based landmarks," in *Proc. Int. Conf. Intell. Veh. Symp. (IV)*, Jul. 2011, pp. 387–392.

[22] S. Venkatraman, J. Caffery, and H. R. You, "A novel ToA location algorithm using LoS range estimation for NLoS environments," *IEEE Trans. Veh. Technol.*, vol. 53, no. 5, pp. 1515–1524, Sep. 2004.

[23] N. Drawil and O. Basir, "Intervehicle-communication-assisted localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 678–691, Sep. 2010.

[24] G. M. Hoang, B. Denis, J. Härri, and D. T. M. Slock, "Distributed link selection and data fusion for cooperative positioning in GPS-aided IEEE 802.11p VANETs," in *Proc. IEEE Workshop Positioning Navigat. Commun.*, Mar. 2015, pp. 1–6.

[25] A. Kakkavas, M. H. C. Garcia, R. A. Stirling-Gallacher, and J. A. Nossek, "Multi-Array 5G V2V relative positioning: Performance bounds," in *Proc. Int. Conf. IEEE Global Commun. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 206–212.

[26] H. Aly, A. Basalamah, and M. Youssef, "Robust and ubiquitous smartphone-based lane detection," *Pervasive Mobile Comput.*, vol. 26, pp. 35–56, Feb. 2016.

[27] J. Yin, Q. Wan, S. Yang, and K. C. Ho, "A simple and accurate TDOA-AOA localization method using two stations," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 144–148, Dec. 2016.

[28] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan, "A comparative study on machine learning algorithms for indoor positioning," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, Madrid, Spain, 2015, pp. 1–8.

[29] M. Becker and B. Ahuja, "Implementing real-life indoor positioning systems using machine learning approaches," in *Proc. 8th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Larnaca, Cyprus, 2017, pp. 1–6.

[30] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE BigData Congr.*, Honolulu, HI, USA, Jun. 2017, pp. 557–564.

[31] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: http://www.ildgroup.si/uploads/product/20/bitcoin.pdf

[32] (Dec. 2017). *Ethereum*. [Online]. Available: https://www.ethereum.org/

[33] D. Larimer. (Nov. 2013). *Transactions as Proof-of-Stake*. [Online]. Available: https://bravenewcoin.com/assets/Uploads/TransactionsAsProofOfStake10.pdf

[34] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, Nov. 2002.

[35] (Mar. 16, 2018). *EOS.IO Technical White Paper v2*. [Online]. Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md#consensus-algorithm-bft-dpos

[36] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE SP*, San Jose, CA, USA, May 2016, pp. 839–858.

[37] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular ad-hoc networks," in *Proc. Int. Conf. ACM Pervasive Ubiquitous Comput., Adjunct*, Sep. 2016, pp. 137–140.

[38] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 119–125, Dec. 2017.

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

[40] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019, doi: 10.1109/JIOT.2018.2836144.

[41] Y. Fu, C. Li, H. Luan, Y. Zhang, and G. Mao, "Infrastructure-cooperative algorithm for effective intersection collision avoidance," *Transport. Res. C, Emer.*, vol. 2018, no. 89, pp. 188–204, 2018.

[42] S. Miura, L. T. Hsu, F. Chen, and S. Kamijo, "GPS error correction with pseudorange evaluation using three-dimensional maps," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3104–3115, Dec. 2015.

[43] I. Skog and P. Hndel, "In-car positioning and navigation technologies–A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 4–21, Mar. 2009.

[44] J. Du and M. J. Barth, "Next-generation automated vehicle location systems: Positioning at the lane level," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 48–57, Mar. 2008.

[45] A. Courville, I. Goodfellow, Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[46] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesús, *Neural Network Design*. Cambridge, MA, USA: Martin Hagan, 2014.

[47] A. Ng, "Machine learning," Stanford Univ., Stanford, CA, USA, Jul. 2008. [Online]. Available: https://www.youtube.com/view_play_list?p=A89DCFA6ADACE599

[48] (Feb. 9, 2018). *EOS Storage Whitepaper*. [Online]. Available: https://medium.com/eosio/start-an-eos-meetup-in-your-community-761be355fce

[49] E. Santana and G. Hotz, "Learning a driving simulator," Aug. 2016, *arXiv:1608.01230*. [Online]. Available: https://arxiv.org/abs/1608.01230

Changle Li (M'09–SM'16) received the Ph.D. degree in communication and information system from Xidian University, China, in 2005. He conducted his Post-Doctoral Research in Canada and the National Institute of information and Communications Technology, Japan, respectively. He had been a Visiting Scholar with the University of Technology Sydney. He is currently a Professor with the State Key Laboratory of Integrated Services Networks, Xidian University. His research interests include intelligent transportation systems, vehicular networks, mobile ad hoc networks, and wireless sensor networks.

Yuchuan Fu received the B.Eng. degree in telecommunication engineering from the Xi'an University of Posts and Telecommunications, China, in 2014. She is currently pursuing the Ph.D. degree in telecommunication engineering with Xidian University, Xi'an, China. Her current research interests include communication protocol and algorithm design in vehicular networks and wireless sensor networks.

Fei Richard Yu (S'00–M'04–SM'08–F'18) received the Ph.D. degree in electrical engineering from The University of British Columbia in 2003.

From 2002 to 2006, he was with Ericsson, Lund, Sweden, and a start-up in California, USA. He joined Carleton University in 2007, where he is currently a Professor. His research interests include wireless cyber-physical systems, connected/autonomous vehicles, security, distributed ledger technology, and deep learning. He is a registered Professional Engineer in ON, Canada. He is a fellow of the Institution of Engineering and Technology. He is a Distinguished Lecturer, the Vice President (Membership), and an Elected Member of the Board of Governors of the IEEE Vehicular Technology Society. He received the IEEE Outstanding Service Award in 2016, the IEEE Outstanding Leadership Award in 2013, the Carleton Research Achievement Award in 2012, the Ontario Early Researcher Award (formerly Premiers Research Excellence Award) in 2011, the Excellent Contribution Award from IEEE/IFIP TrustCom 2010, the Leadership Opportunity Fund Award from the Canada Foundation of Innovation in 2009, and the Best Paper Awards from IEEE VTC 2017 Spring, ICC 2014, Globecom 2012, IEEE/IFIP TrustCom 2009, and International Conference on Networking 2005. He has served as the technical program committee co-chair for numerous conferences. He serves on the Editorial Boards for several journals, including the Co-Editor-in-Chief for *Ad Hoc and Sensor Wireless Networks*, a Lead Series Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.

Tom H. Luan (M'14–SM'18) received the B.Eng. degree from Xi'an Jiaotong University, China, in 2004, the M.Phil. degree from The Hong Kong University of Science and Technology in 2007, and the Ph.D. degree from the University of Waterloo, Ontario, Canada, in 2012. He is currently a Professor with the School of Cyber Engineering, Xidian University, Xi'an, China. He has authored/coauthored more than 40 journal articles and 30 technical articles in conference proceedings. He awarded one U.S. patent. His research mainly focuses on content distribution and media streaming in vehicular ad hoc networks and peer-to-peer networking and the protocol design and performance evaluation of wireless cloud computing and edge computing. He served as a TPC Member for IEEE Globecom, ICC, and PIMRC; and the Technical Reviewer for multiple IEEE Transactions, including TMC, TPDS, TVT, TWC, and ITS.

Yao Zhang received the B.Eng. degree in telecommunication engineering from the Xi'an University of Science and Technology, China, in 2015. He is currently pursuing the Ph.D. degree in telecommunication engineering with Xidian University, Xi'an, China. His current research interests include communication protocol and performance evaluation of vehicular networks, edge caching, and wireless sensor networks.