Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.035526

ARTICLE

A Trusted Edge Resource Allocation Framework for Internet of Vehicles

Yuxuan Zhong[1], Siya Xu[1], Boxian Liao[1], Jizhao Lu[2], Huiping Meng[2], Zhili Wang[1], Xingyu Chen[1,*] and Qinghan Li[3]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[2]State Grid Henan Information & Telecommunication Company, Zhengzhou, 450018, China

[3]School of Information Studies, Syracuse University, Syracuse, 13244, USA

*Corresponding Author: Xingyu Chen. Email: chenxy@bupt.edu.cn

Received: 24 August 2022 Accepted: 08 December 2022 Published: 29 November 2023

ABSTRACT

With the continuous progress of information technique, assisted driving technology has become an effective technique to avoid traffic accidents. Due to the complex road conditions and the threat of vehicle information being attacked and tampered with, it is difficult to ensure information security. This paper uses blockchain to ensure the safety of driving information and introduces mobile edge computing technology to monitor vehicle information and road condition information in real time, calculate the appropriate speed, and plan a reasonable driving route for the driver. To solve these problems, this paper proposes a trusted edge resource allocation framework for assisted driving service, which includes two stages: the blockchain generation stage (the first stage) and assisted driving service stage (the second stage). Furthermore, in the first stage, a delay-and-throughput-oriented block generation model for the mobile terminal is designed. In the second stage, a balanced offloading algorithm for assisted driving service based on edge collaboration is proposed to solve the problems of unbalanced load of cluster mobile edge computing (MEC) servers and low resource utilization of the system. And this paper optimizes the throughput of blockchain and delay of the transportation network through deep reinforcement learning (DRL) algorithm. Finally, compared with joint computation and communication resources' allocation (JCCR) and resource allocation method based on binary offloading (RAB), our proposed scheme can optimize the delay by 7.4% and 26.7%, and support various application services of the vehicular networks more effectively.

KEYWORDS

Blockchain; load balancing; vehicular networks; resource allocation

1 Introduction

The Internet of Vehicles (IoV) is a new mobile ad hoc network recent years, which ensures road safety and improves people's driving experience. With the increasing traffic flow of urban road networks, drivers inevitably encounter congested areas in the process of driving, which greatly increases travel time, fuel consumption, and additional gas emissions. To solve this problem, assisted driving technology is introduced to reduce the probability of accidents, and correct the vehicle driving routes in time after detecting accidents to improve transportation efficiency. We can obtain the current vehicle position, direction, speed, and other driving data through the onboard unit (OBU), and infer the driver's dangerous driving intensity, to verify the driver's behavior and curb the accident from the source [1]. We comprehensively consider the current vehicle information (such as vehicle identification (ID), location, acceleration, speed, and destination) and road condition information, calculate the reasonable speed, and plan a recommended driving route to prompt the driver.

In the traditional assisted driving scheme, information processing is concentrated on one side, which is vulnerable to attack. And each server needs to maintain a large amount of account information, resulting in a waste of resources for many servers, which may also lead to server performance bottlenecks [2]. Blockchain technology can establish trust relationships for nodes in vehicular networks, effectively solve the security problems such as decentralized management and privacy protection of the IoV, and reduce the risk of information tampering and data center attacks [2]. However, many existing public blockchains cannot meet the requirements of high throughput and low delay of assisted driving tasks.

In the application of smart transportation, the frequent driving operation of vehicles makes the nearby data processing and strict delay requirements become extremely important. MEC has the ability of high bandwidth and low latency. It can provide computing services for all kinds of mobile network edges, which can reduce the latency caused by long-distance communication transmission, and make higher performance. Therefore, we use MEC to realize the real-time processing of computing-intensive tasks generated by driving assistance and reduce the total delay of service [3].

The demand for a large number of vehicles for assisted driving makes vehicle computing services face great challenges. Many schemes to solve the problem of assisted driving have been proposed. A driving assistance scheme based on the driver's behavior was proposed in [1,4], which obtains driving information through sensors and cameras, and provides reasonable suggestions to drivers after processing. But it cannot adjust the driving route dynamically. References [2,5] proposed anonymous identity authentication schemes, which applied blockchain in the IoV service to protect the security of vehicle information, but they did not consider improving the throughput of blockchain. References [3,6] designed a scheme to deploy the edge computing function to roadside units (RSUs). However, they did not consider the task allocation of collaborative offloading and load balancing of MEC servers. To sum up, the existing researches still have many problems [1–6].

To solve these problems, we first uses a digital map to configure road condition information such as ramp geometry and traffic information on the road-side unit (RSU) and then generates a block on the vehicle to store some parameters of the vehicle obtained through OBU, such as vehicle ID, location, acceleration, speed, destination. In this process, the block size, block interval, and offloading decision of tasks are jointly optimized to reduce delay and maximize transaction throughput. Then the parameters of the vehicle will be uploaded to the RSU, and the destination, speed, and other attributes will be comprehensively analyzed to plan the appropriate speed and route. At the same time, the block will be generated on the RSU and the processing results will be stored in it. Aiming at the computing-intensive tasks generated in this stage, we propose a balanced offloading algorithm for assisted driving service based on edge cooperation and jointly optimize the offloading mode and ratio to further optimize the total delay of the system. The main contributions of this paper are as follows:

(1) A trusted processing framework for assisted driving service is established, the whole framework includes two stages: the first stage designs a delay-and-throughput oriented block generation model for mobile terminals, and the second stage proposes a balanced offloading algorithm for assisted driving service based on edge collaboration. The two-stage optimization problem is modeled as the Markov decision process (MDP), and the paper introduces the deep reinforcement learning (DRL) algorithm to solve it.

(2) A load-balancing model for clustered MEC servers is proposed by comprehensively considering the usage of storage and computing resources of MEC servers. The paper designs a load factor based on the model, and models the MEC server as the M/M/1 queue, which affects the task offloading ratio in the collaborative computing mode and further optimized the total delay of the system.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 designs the trusted processing framework for assisted driving services. Section 4 proposes the two-stage joint optimization problems, and uses the DRL algorithm to solve these problems. Section 5 analyses the simulation results, and Section 6 draws a conclusion.

2 Related Work

This section introduces the existing assisted driving technology, the application of blockchain in vehicular networks, and the computing offloading methods based on MEC.

2.1 Assisted Driving Technology

To improve the efficiency of traffic operations and ensure the safety of drivers, many assisted driving schemes have been proposed. Many schemes [1,4,7,8] deal with the driving information but do not consider the efficiency of the processing. All the calculation tasks are completed on the vehicle. Zhang et al. [9] proposed an abnormal behavior detection scheme for autonomous vehicles. The scheme offloads the computing tasks to the MEC server with strong computing power and large storage space, which greatly improves the efficiency of service processing. However, the scheme ignores the security of data. Therefore, to solve computing-intensive tasks generated in assisted driving process and security problems, we introduce MEC technology and blockchain to the paper.

2.2 Application of Blockchain in the Vehicular Networks

To guarantee the safety of data, several works [10,11] built blockchain systems for IoV. Bai et al. [12] proposed a driving behavior warning scheme based on blockchain. The road information and driving data are regarded as transactions to ensure that the correct road condition is broadcast to other vehicle locations. Yao et al. [2] proposed an anonymous authentication mechanism, which can effectively resist the attack of data tampering by combining encryption with blockchain. Yang et al. [13] proposed a consensus mechanism based on proof-of-work (PoW) and proof-of-stake (PoS) for blockchain in the IoV. To prevent the dissemination of forged information from interior terminals, Zheng et al. [14] designed an anonymous authentication scheme based on blockchain, which ensures the safety of communication between vehicles.

The use of blockchain technology can effectively guarantee the security requirements of the IoV, but the above schemes all ignore the transaction throughput of blockchain. In order to improve transaction throughput, this paper optimizes the size and interval of generated blocks.

2.3 Computing Offloading Methods Based on Mobile Edge Computing

MEC servers can provide flexible computing and storage resources for various terminals at the edge of the network. Many pieces of literature [15–17] have proposed various solutions to solve the problem of computing offload in mobile edge computing. Feng et al. [3] proposed an edge computing framework for autonomous vehicles. It arranges the vehicles on the road for scheduling according to the information collected from the adjacent vehicles, to ensure that the vehicle computing resources are fully utilized. Zhao et al. [18] designed a collaborative computing offloading scheme, which made full use of computing resources and reduced the completion time of tasks. However, the above two schemes do not consider the impact of vehicle movement on task offloading. Boukerche et al. [19] proposed a data retrieval scheme for computing offloading in vehicle edge computing (VEC), which modeled the computing offloading task in VEC, estimated the user's position, and then selected the appropriate access point. Yang et al. [20] proposed a mobile-aware task offloading scheme, which optimized the task offloading decision, delay, and computing resource allocation.

Many schemes adopt a binary offloading decision, which does not consider the problem of computing task allocation in the process of multi-device collaborative offloading. We propose a load-balance-based offloading ratio decision model, which takes the resources of the MEC server into account. And we model the resource allocation process in the system as an MDP and introduce A3C to solve the problem.

3 System Model

3.1 Application Model

We propose a trusted edge resource allocation framework for assisted driving service, which consists of a blockchain system, multiple MEC servers, and multiple vehicles, as shown in Fig. 1. In the vehicular network, $\mathbf{M} = \{1, 2, ..., m\}$ represents the set of vehicles. The vehicles collect road conditions and driving data. The OBU and RSU are transmitted by the orthogonal spectrum. Vehicles can generate blocks in the terminal layer. At this time, the blockchain system needs mining in the consensus process, which will generate a computing-intensive task T_m. To ensure the credibility of driving data, we store the data (vehicle ID, location, speed, acceleration, destination, etc.) in the block. The MEC servers are deployed in RSUs and $\mathbf{N} = \{1, 2, ..., n\}$ represents the set of RSUs. After the vehicle is connected to the RSU, the vehicle transmits the parameters existing in the block to the RSU. Then, the RSU processes the data collected by vehicles, generates the assisted driving task T_k, generates the block on the RSU, and stores the processed results (appropriate speed and route) into the blockchain system.

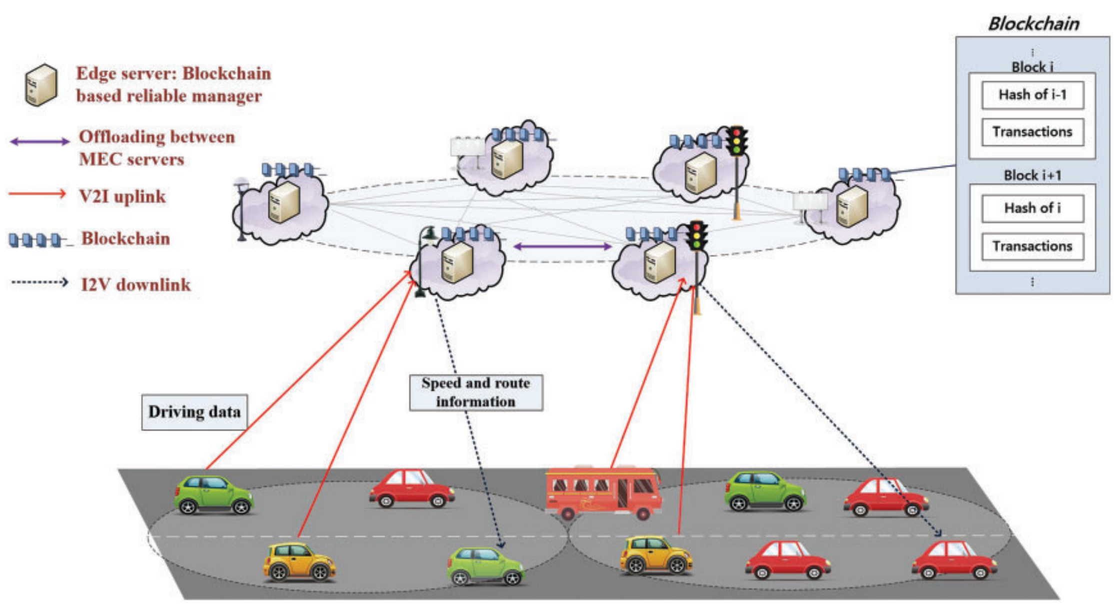


Figure 1: The blockchain-based processing framework for assisted driving service

3.2 Blockchain System Model

We deploy the blockchain system on the mobile terminal and RSU to store the vehicular parameter information (storage transaction) and result information (policy transaction) into the block generated by the terminal and RSU, respectively. A blockchain is a digital ledger in which all executed transactions are stored. It uses a distributed point-to-point network to generate a continuous, ordered list of records. Each block contains a set of signed transactions, which are verified by a consensus mechanism. Using blockchain technology, we can maintain a reliable and consistent database [13], and ensure the safety and credibility of the vehicular network. We introduce the delegated Byzantine fault tolerance (dBFT) consensus mechanism and participate in consensus through proxy voting.

To improve efficiency, the consensus nodes are MEC servers that have a high voting rate. In addition, the voting rate depends on the staking and available resources [21]. We use $\Omega_s(t) = \{\Omega_1(t), \Omega_2(t), \ldots, \Omega_n(t)\}$ to express the staking set, and use $\mathbf{C}_s(t) = \{C_1(t), C_2(t), \ldots, C_n(t)\}$ to express the available computing resource set of the RSU. The RSU has a first-out data buffer, and the processing queue can be expressed as:

$$F_n(t+1) = \max\{F_n(t) - f_R + \rho_n R_l(t), 0\} \tag{1}$$

where $R_l(t)$ is the vehicle calculation rate, f_R is the central processing unit (CPU) cycle frequency of the MEC server and ρ_n is the processing density (in CPU cycles/bit). The computing resources that the RSU can use for the blockchain system in slot t is:

$$C_n(t) = \max\{F - F_n(t), C_{\min}\} \tag{2}$$

Assuming that ϑ is the size of transactions, $S(t)$ means the block size (in MB) and $I(t)$ represents the block interval (in seconds). We express transaction throughput as:

$$\Phi(t) = \lfloor S(t)/\vartheta \rfloor / I(t) \tag{3}$$

3.3 Mobile Communication Model

Assuming that the coverage area of RSU is d_g. Vehicles can transmit data to RSU through vehicle-to-infrastructure (V2I) communication. According to the Shannon theorem, the transmission rate between the moving vehicle and the RSU can be expressed as:

$$R_m(t) = B_n \log_2\left\{1 + \frac{P_n(t)\, g_n(t)\, (d_n(t))^{-\theta}}{\sigma_n^2(t)}\right\} \tag{4}$$

where B_n is the transmission channel bandwidth, $P_n(t)$ is the transmission power, $g_n(t)$ is the channel gain, $\sigma_n(t)$ is the background noise power, θ is the path loss index, d_n is the distance between the vehicle and the RSU [20]. $d_n(t)$ can be denoted as

$$d_n(t) = \sqrt{{h_n}^2 + \left(l_n - d_g/2 + v_n t\right)^2} \tag{5}$$

where l_n is the position of the vehicle n and v_n is the speed of vehicle n.

3.4 Computational Model

There are two stages in the proposed framework. In the first stage of mobile terminal blockchain generation, a mining task T_m will be generated. In the second stage of assisted driving service, the RSU process and analyze the data collected by vehicles, and then generate the calculation task T_k. Both T_m and T_k are computing-intensive tasks, so we propose two different computing modes to perform T_m and T_k: local computing mode and collaborative computing mode [22].

We use $a_n(t) \in \{1, 2, 3\}$ to represent the computing offloading decision, $a_n(t) = 1$ represents the local computing mode (mode 1), $a_n(t) = 2$ represents the collaborative computing mode for adjacent MEC servers (mode 2), and $a_n(t) = 3$ represents the collaborative computing mode for mobile terminals and MEC servers (mode 3).

3.4.1 Local Computing Mode (Mode 1)

(1) Local mobile terminal independent computing mode (mode 1-V)

For the local mobile terminal, we use a parameter tuple $< D_m, X_m, \varphi_m >$ to express a task, where D_m is the data size (in bits), X_m is calculation intensity (in CPU cycles/bit), and φ_m is completion time (in seconds) of the blockchain mining task T_m. f_v is the CPU cycle frequency of vehicles, and the local vehicle processing time T_m can be denoted as:

$$T_m(t) = \frac{D_m X_m}{f_v} \tag{6}$$

(2) Local MEC server independent computing mode (mode1-R)

The task T_k offloaded to RSU is expressed by a tuple $< D_t, X_t, \lambda_t, \varphi_t >$, where includes the task data size D_t, the calculation intensity X_t, the proportion of result data to task data λ_t, and maximum processing time φ_t. Suppose that the task generated by the vehicle conforms to the Poisson distribution, so the task will be sent to RSU with an arrival rate λ_i. We further model the MEC server as the M/M/1 queue. Therefore, average delay $\Delta T_l(t)$ of the local RSU includes queuing delay and execute delay, and it can be denoted as:

$$\Delta T_l(t) = \frac{1}{1/T_{exe,l}(t) - \lambda_i} \tag{7}$$

where $T_{exe,l}(t)$ is the processing delay of the local RSU to execute the local task T_k, and it can be calculated by:

$$T_{exe,l}(t) = \frac{D_t X_t}{f_R} \tag{8}$$

3.4.2 Collaborative Computing Mode for Adjacent Mobile Edge Computing Servers (Mode 2)

Similar to [23], we analyze the performance of mode 1. The calculation task T_k can be divided into two parts: one part is αD_t, which is calculated on the surrounding RSU n_j, where α is the offloading ratio, and the other part is $(1 - \alpha) D_t$, which is calculated on the local RSU n_i, and the final calculation results will be sent back and summarized on the local RSU n_i. As the task generated by the vehicle follows the Poisson distribution and be sent to local RSU n_i at a stable arriving rate, the total processing delay of tasks on n_i is affected by load factor δ_i, arrival rate $(1 - \delta_i)\lambda_i$ and the local processing delay $T_{exe,i}(t)$. Then, the average delay $\Delta T_i(t)$ of the RSU n_i can be denoted as:

$$\Delta T_i(t) = \frac{1}{1/T_{exe,i}(t) - (1 - \delta_i)\lambda_i} \tag{9}$$

$$T_{exe,i}(t) = \frac{(1 - \alpha) D_t X_t}{f_R} \tag{10}$$

Moreover, the task offloaded to the surrounding RSU n_j follows the Poisson process. Similarly, the average delay $\Delta T_j(t)$ of RSU n_j can be denoted as:

$$\Delta T_j(t) = \frac{1}{1/T_{exe,j}(t) - \delta_i \lambda_i} \tag{11}$$

where $\delta_i \lambda_i$ is the arrival rate, and $T_{exe,j}(t)$ can be denoted as:

$$T_{exe,j}(t) = \frac{\alpha D_t X_t}{f_R} \tag{12}$$

The transmission rate $R_{i,j}(t)$ and communication delay $T_{i,j}(t)$ of the offloading part of the calculation task in mode 1between RSU n_i to n_j can be denoted as:

$$R_{i,j}(t) = B_m \cdot \log_2\left(1 + \frac{P_{i,j}(t)\, g_{i,j}(t)}{\sigma_{i,j}^2(t)}\right) \tag{13}$$

$$T_{i,j}(t) = \frac{\alpha D_t}{R_{i,j}(t)} \tag{14}$$

where $P_{i,j}(t)$ is the transmission power, $g_{i,j}(t)$ is the channel gain, and $\sigma_{i,j}(t)$ is the background noise power from n_i to n_j in time slot t [21].

Therefore, the total delay of the system in the assisted driving service stage can be denoted as:

$$T_{tot,s2}(\mathrm{t}) = (2 - a_n(\mathrm{t}))\, \Delta T_l(t) + (a_n(\mathrm{t}) - 1)\, T_{tot,ij}(t) \tag{15}$$

3.4.3 Collaborative Computing Mode for Mobile Terminals and Mobile Edge Computing Servers (Mode 3)

Similar to mode 2, we analyze the performance of mode 3. The mining task T_m can be divided into two parts: $(1 - \beta) D_m$ is calculated on the local mobile terminal, βD_m is calculated on the MEC server n_k, where β is the offloading ratio, and the processed result is returned to the mobile terminal. The time taken by the mobile terminal to execute part of the mining task can be expressed as:

$$T_{exe,m}(t) = \frac{(1 - \beta) D_m X_m}{f_v} \tag{16}$$

The time taken to offload the remaining tasks from the vehicle to RSU n_k can be expressed as:

$$T_{tra}(t) = \frac{\beta D_m}{R_m(t)} \tag{17}$$

where $R_m(t)$ is the transmission rate.

Therefore, the average time delay $\Delta T_k(t)$ of the RSU n_k can be denoted as:

$$\Delta T_k(t) = \frac{1}{1/T_{exe,k}(t) - \lambda_i} \tag{18}$$

The total delay of the system in blockchain generation stage is:

$$T_{tot,s1}(\mathrm{t}) = \frac{1}{2}(3 - a_n(\mathrm{t}))\, T_m(t) + \frac{1}{2}(a_n(\mathrm{t}) - 1)\, T_k(t) \tag{19}$$

3.5 Load Balancing Model

We use $W_{cap,i}(t) = \left\{E_{cap,i}(t),\ C_{cap,i}(t)\right\}$ to represent the performance of RSU n_i, where $E_{cap,i}(t)$ means storage resources and $C_{cap,i}(t)$ represents computing resources of the RSU n_i. The storage resource utilization $E_i(t)$ and the computing resource utilization $C_i(t)$ of the MEC server n_i can be denoted as:

$$E_i(t) = \frac{E_{cur,i}(t)}{E_{cap,i}(t)} \tag{20}$$

$$C_i(t) = \frac{C_{cur,i}(t)}{C_{cap,i}(t)} \tag{21}$$

where $E_{cur,i}(t)$ is the storage resource usage and $C_{cur,i}(t)$ represents the computing resource usage of RSU n_i.

We define the load rate of RSU n_i as:

$$F_{load,i}(t) = \gamma E_i(t) + \eta C_i(t) \tag{22}$$

where $\gamma + \eta = 1$, γ and η represent the weight.

The average load rate $F_{\text{ave}}(t)$ can be calculated by:

$$F_{\text{ave}}(t) = \frac{1}{n}\sum_{i=1}^{n} F_{load,i}(t) \tag{23}$$

The more balanced the load of the cluster, the more stable it will be. We use the standard deviation σ_F to measure the load balance of the cluster, and calculate it by:

$$\sigma_F = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(F_{load,i}(t) - F_{\text{ave}}(t)\right)^2} \tag{24}$$

Finally, we define the load factor δ_i as the load of the cluster:

$$\delta_i = \frac{F_{load,i}(t) - F_{\text{ave}}(t) + \sigma_F}{2\sigma_F} \tag{25}$$

4 Problem Formulation

Since the wireless channel has Markovian property, we modeled the blockchain-based processing framework for assisted driving service as MDP. Generally speaking, reinforcement learning (RL) can solve MDP problems greatly. The RL adjusts the strategy adaptively according to the reward function through learning, and then obtains the optimal strategy. However, due to traditional RL has low learning efficiency and quality in complex environment, we use DRL to solve the problem of resource allocation [24].

4.1 Blockchain Generation Stage (The First Stage)

4.1.1 Optimization Objective

In the process of driving, the vehicle acquires the driver and vehicle data and generates a block on the vehicle. At this time, the blockchain mining process on the mobile terminal will produce computing-intensive tasks, which seriously affects the system delay. To solve this problem, we consider an optimization problem with a delay-and-throughput focus that jointly optimize the offloading location, block size, and block interval, which is defined in formula (26):

$$
\begin{aligned}
P1:\ \max\ & \sum_{t=0}^{T-1}\left[\omega_1\Phi(t)-(1-\omega_1)\,\omega_2\sum_{n=1}^{N}T_n(t)\right] \\
\text{s.t.}\quad & a_n(\mathrm{t})\in\{1,3\} && \mathrm{C1} \\
& \beta(\mathrm{t})\in[0,1] && \mathrm{C2} \\
& 0\le f_v, f_R\le f_{\max} && \mathrm{C3} \\
& T_n(t)\le\varphi_m && \mathrm{C4} \\
& 0\le P_{tot,n}(t)\le P_m && \mathrm{C5} \\
& D_m\le Z_m && \mathrm{C6}
\end{aligned}
\tag{26}
$$

where ω_1 $(0<\omega_1<1)$ and ω_2 are the weight factor and the mapping factor of the optimization function, respectively. Constraints C1 and C2 represent the effectiveness of the offloading decision. Constraint C3 indicates that the CPU frequency of the vehicle and RSU cannot exceed the maximum value. Constraint C4 represents the limit of the completion delay. Constraint C5 indicates that the total transmission power should be less than the available power P_m of the system. Constraint C6 represents that the data size of the offloading task cannot exceed the link capacity Z_m.

4.1.2 Problem Transformation

The state space of the current **timeslot** can be expressed as the combination of the number of stacks $\Omega_s(t)=\{\Omega_1(t),\Omega_2(t),\ldots,\Omega_n(t)\}$, the available storage resource $\mathbf{E_s}(t)=\{E_1(t),E_2(t),\ldots,E_n(t)\}$ of the MEC server and channel condition $\mathbf{G_1}(t)=\{\mathrm{g}_n(t)\}$:

$$
\mathbf{S}_{s1}(t)\triangleq\{\Omega_s(t),\mathbf{E_s}(t),\mathbf{G}_1(t)\}
\tag{27}
$$

We denote the state transition probability density function by f. The probability from one system state $s_{s1}(t)$ to the next system state $s_{s1}(t+1)$ is:

$$
Pr\left(s_{s1}(t+1)\,|s_{s1}(t),a_{s1}(t)\right)=\int_{s_{s1}{}^{t}}^{s_{s1}{}^{t+1}} f\left(s_{s1}(t),a_{s1}(t),s_{s1}\right)ds_{s1}
\tag{28}
$$

Offloading mode $\mathbf{a}(\mathrm{t})$, offloading rate from vehicles to MEC servers $\beta(\mathrm{t})$, block size $S(\mathrm{t})$ and block interval $I(\mathrm{t})$ constitute the action space. $\mathbf{A}_{s1}(t)$ mean the action set in the formula (29):

$$
\begin{aligned}
&\mathbf{A}_{s1}(t)\triangleq\{\mathbf{a}(t),\beta(\mathrm{t}),S(\mathrm{t}),I(\mathrm{t})\},\beta(\mathrm{t})\in[0,1]\\
&\mathbf{a}(\mathrm{t})\triangleq\{a_1(\mathrm{t}),a_2(\mathrm{t}),\cdots,a_N(\mathrm{t})\}
\end{aligned}
\tag{29}
$$

To improve the efficiency and security of consensus process, we select the MEC server which has a high voting rate as the consensus node according to the stake and available resources held by the node, and use the dBFT algorithm as the consensus mechanism. Using the limit fraction method [21], as the equations below, the size of a block and the interval between blocks are determined:

$$
S(\mathrm{t})\in[0.2,S_{\max}]
\tag{30}
$$

$$
I(\mathrm{t})\in[0.1,I_{\max}]
\tag{31}
$$

where $S_{\max}$ is the maximum value of block size and $I_{\max}$ is the upper limit of block interval.

Finally, the reward function can be expressed as:

$$
r_{s1}=\begin{cases}W_{s1}(\mathrm{t}), & \text{if } C1-C6 \text{ are satisfied}\\ 0, & \text{otherwise}\end{cases}
\tag{32}
$$

where $W_{s1}(\mathrm{t})=\omega_1\Phi(\mathrm{t})-(1-\omega_1)\,\omega_2\sum_{n=1}^{N}T_n(t)$.

4.2 Assisted Driving Service Stage (The Second Stage)

4.2.1 Optimization Objective

To optimize the resource allocation and realize the balanced offloading of the assisted driving service, we put forward an optimization problem to further reduce total latency and jointly optimize the offloading mode and offloading rate:

$$\begin{aligned} P2: \quad & \min \sum_{t=0}^{T-1} \sum_{n=1}^{N} T_{tot,n}(\mathrm{t}) \\ \text{s.t.} \quad & a_n(\mathrm{t}) \in \{1, 2\} && \mathrm{C1} \\ & \alpha(\mathrm{t}) \in [0, 1] && \mathrm{C2} \\ & 0 \le f_R \le f_{\max} && \mathrm{C3} \\ & T_{tot}(t) \le \varphi_t && \mathrm{C4} \\ & c_j \le C_{cap,j}(t), \quad j \in \mathbf{N} && \mathrm{C5} \\ & e_j \le E_{cap,j}(t), \quad j \in \mathbf{N} && \mathrm{C6} \end{aligned} \tag{33}$$

Among them, constraints C1 and constraints C2 guarantee the decision on the offloading mode and offloading ratio of the task is effective. Constraint C3 represents that the RSU's CPU frequency does not exceed its maximum value $f_{\max}$. The total delay to execute a computing task is limited to constraint C4. Constraints C5 and C6 indicate that the MEC server needs to meet the computing resource requirement c_j and storage resource requirement e_j in task assignment.

4.2.2 Problem Transformation

Our state space includes available computing resources $\mathbf{C_s}(t) = \{C_1(t), C_2(t), \ldots, C_n(t)\}$, available storage resources $\mathbf{E_s}(t) = \{E_1(t), E_2(t), \ldots, E_n(t)\}$, and channel condition $\mathbf{G_2}(t) = \left\{\mathrm{g}_{i,j}(t)\right\}$ of RSUs:

$$\mathbf{S}_{s2}(t) \triangleq \{\mathbf{C_s}(t), \mathbf{E_s}(t), \mathbf{G_2}(t)\} \tag{34}$$

The probability of leaving the current state $s_{s2}(t)$ to the next state $s_{s2}(t+1)$ can be defined as:

$$Pr(s_{s2}(t+1) | s_{s2}(t), a_{s2}(t)) = \int_{s_{s2}{}^{t}}^{s_{s2}{}^{t+1}} f(s_{s2}(t), a_{s2}(t), s_{s2})\, ds_{s2} \tag{35}$$

The action space consists of offloading mode $\mathbf{a}(\mathrm{t})$ and offloading ratio decision $\alpha(\mathrm{t})$. $\mathbf{A}_{s2}(t) \triangleq \{\mathbf{a}(t), \alpha(\mathrm{t})\}$ mean the action set. Where, $\alpha(\mathrm{t}) \in [0, 1]$, and $\mathbf{a}(\mathrm{t})$ is:

$$\mathbf{a}(\mathrm{t}) \triangleq \{a_1(\mathrm{t}), a_2(\mathrm{t}), \cdots, a_N(\mathrm{t})\} \tag{36}$$

We express the reward function as:

$$r_{s2} = \begin{cases} W_{s2}(\mathrm{t}), & \text{if } C1 - C6 \text{ are satisfied} \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

4.3 Algorithm Design

Since optimization algorithms have the problem of falling into the local optimal solution or using too long a solution time, the computation speed or quality of them is low. By comparison, the DRL algorithm can effectively solve these problems. Value-based approach and policy-based approach are two main types of RL. The value-based approach solve problems by predicting the total value of a given state from now to the future. The policy-based approach adjusts the probability distribution of each action in a given state. A3C is an asynchronous RL algorithm [25]. A3C can work in discrete and continuous action spaces, which combines the advantages of two RL methods. Through the A3C algorithm, computing resources are fully utilized and data correlations are effectively broken. We apply A3C algorithm for our problem.

As in Fig. 2, the blockchain-based processing framework for assisted driving service includes two stages: the blockchain generation stage and assisted driving service stage. First of all, we use the sensors to get the dynamic and driving data of the vehicle and store some parameters of the vehicle, such as vehicle ID, location, acceleration, speed, destination, etc., into the block generated on the terminal. Then, the collected vehicle information and road condition information are comprehensively calculated and analyzed. This stage will produce computing-intensive tasks. To optimize the total latency and realize the load balancing among cluster MEC servers, we propose a balanced offloading algorithm for assisted driving service based on edge collaboration, and jointly optimize the offloading mode and offloading ratio. And the processing results of assisted driving service will be stored in the block generated on the RSU.

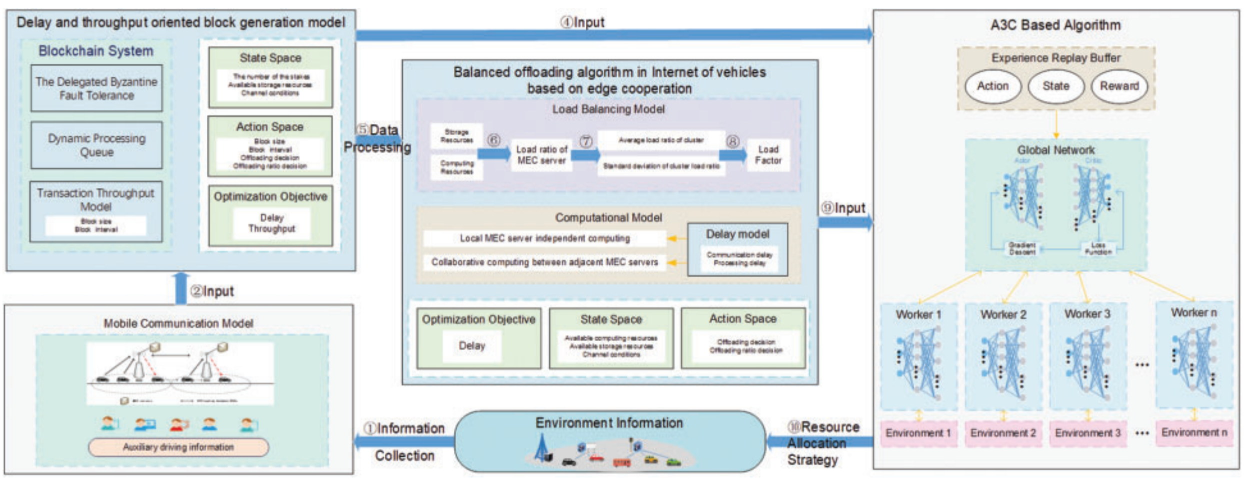


Figure 2: The flow chart of the algorithm

5 Simulation Results and Analysis

5.1 Parameter Settings and Comparison Algorithms

We use TensorFlow2.0 for simulation on a Python 3.7-based simulator. Table 1 lists relevant parameters and super parameters in the simulation. To validate the efficiency of the first stage algorithm, we use several algorithms for comparison: (1) FAJORA [26]: The size of block is fixed; (2) FBT [21]: The block interval is fixed.

Table 1: Experimental parameters and hyperparameter setting

Symbol	Definition	Value
f_R	CPU cycle frequency of the MEC server	2.4 GHz [27]
f_v	CPU cycle frequency of the mobile terminal	1.0 GHz
B_n	Bandwidth between the vehicle and RSU	180 KHz [26]
B_m	Bandwidth between RSUs	10 MHz [28]

(Continued)

Table 1 (continued)

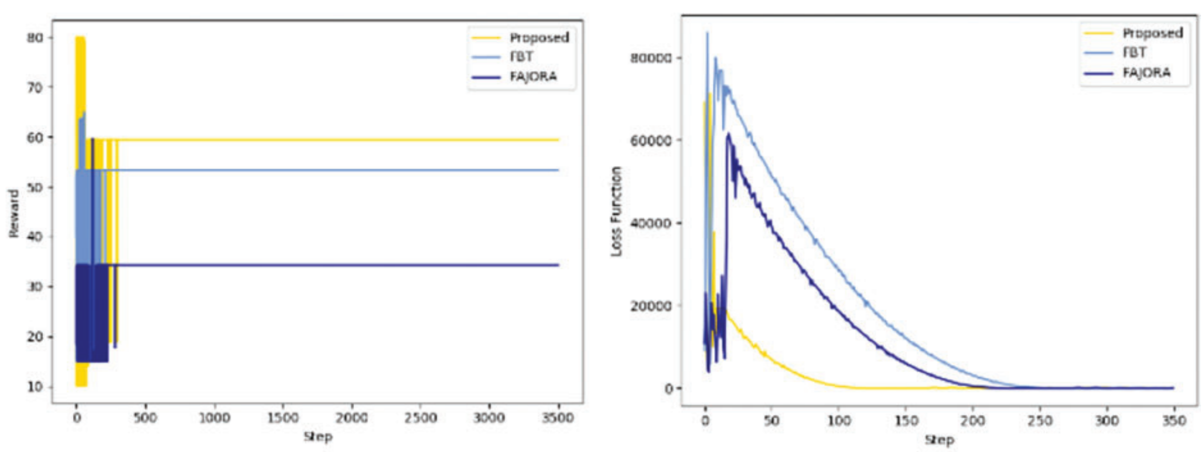
Symbol	Definition	Value
d_g	The coverage area of RSU	600 m
h_n	Height of the MEC server	20 m
θ	Path loss index	2.5 [29]
N_0	Noise power density	−174 dBm/Hz
λ_t	The data size ratio of the output and input	0.01 [22]
ω_2, ω_1	Weighted values	0.2, 0.5 [23]

At the same time, to validate the performance of the second stage algorithm, the following comparison algorithms are selected: (1) JCCR [23]: A joint allocation scheme of computing and communication resources is proposed by optimizing the offloading ratio in the process of computing offload, but the load difference of cluster MEC servers is not considered; (2) RAB: A resource allocation scheme for collaborative offloading, but with binary offloading; (3) Simulated annealing algorithm (SA): Obtain a near-optimal solution through reducing the temperature parameter from its initial parameter in polynomial time.

5.2 Simulation Results

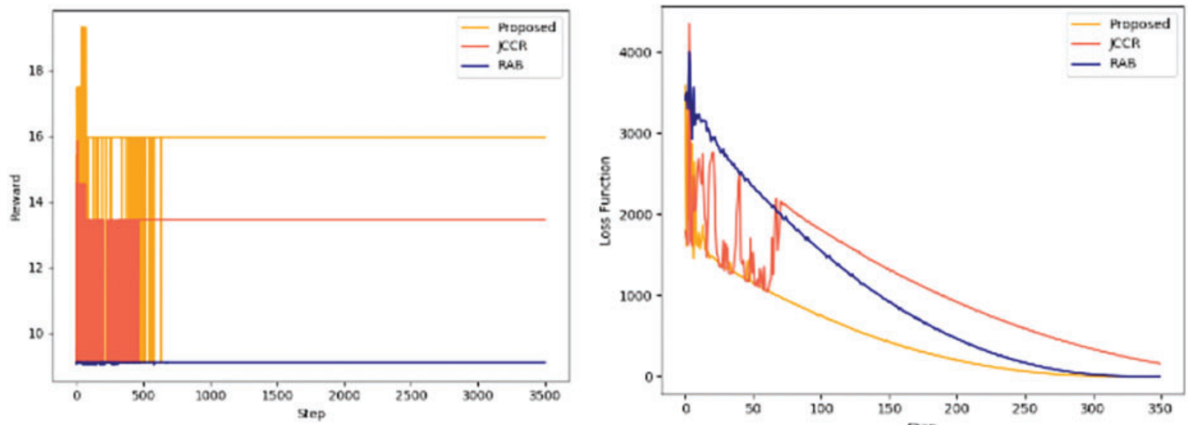
This part compares the performance of the above algorithms and analyze the influence of different parameter settings.

Fig. 3 shows the learning curve based on A3C, in which we compare the changes in the reward and loss function of the proposed algorithm in different stages with other comparison algorithms. From Figs. 3a and 3c, it can be found that the proposed algorithm has the highest reward, and its overall performance is better than the other two algorithms. Figs. 3b and 3d show that the loss function of the proposed algorithm converges faster and has better robustness.



(a) Reward in the first stage (b) Loss function in the first stage

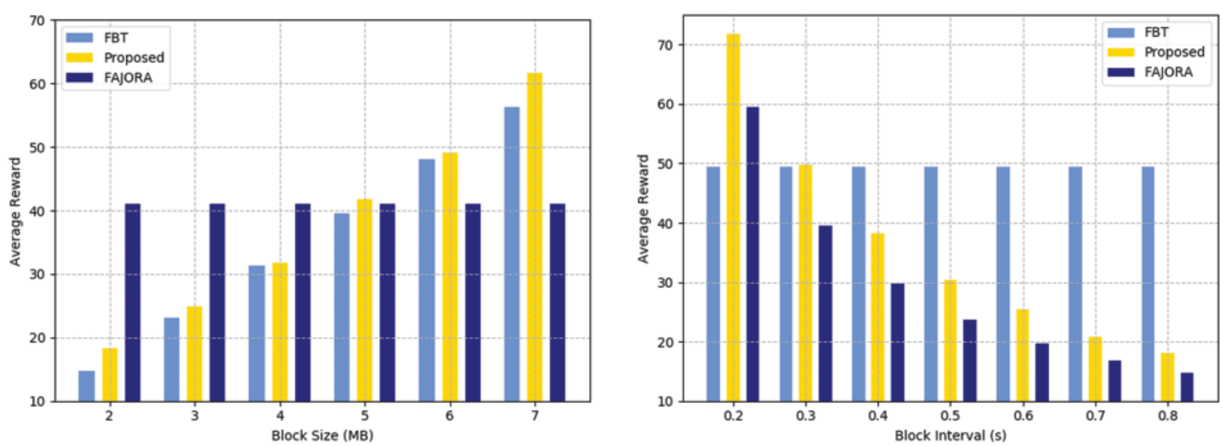
Figure 3: (Continued)



(c) Reward in the second stage (d) Loss function in the second stage

Figure 3: The learning curve of the DRL-based methods

Fig. 4 compares the effects of block size and block interval on the reward under FAJORA, FBT, and the proposed algorithm. As we can see from Fig. 4a, except for the FAJORA, when block size increases, the reward increases slowly accordingly. Meanwhile, Fig. 4b illustrate that, except for FBT, as block interval increases, the reward decreases accordingly. This is because the impact of block interval on transaction throughput is inverse, while the impact of block size is positive, which has an impact on the reward. And the proposed algorithm has better performance, it is because the size of blocks algorithm is the same in the FAJORA, and the block interval is the same in the FBT, which affects the transaction throughput of the blockchain.



(a) Average reward at different block sizes (b) Average reward at different block intervals

Figure 4: Influence of block size and block interval on reward

We can see the variation of the system's total delay under different CPU cycle frequency f_R from Fig. 5. With the increase in f_R, the total delay of all schemes decreases correspondingly. This is because the increase in CPU frequency f_R will improve the calculation speed of executing tasks, shorten the execution time of tasks, and reduce the total delay. The SA algorithm of comparison is a heuristic algorithm, but some random factors are introduced into the search process. As shown in Fig. 5, although the heuristic algorithm converges quickly, it is easy to fall into the local optimal solution prematurely. As a DRL algorithm, A3C can train a high-quality model by continuously learning from the different environment state, and make action decisions according to it. Therefore, the solution quality of the SA algorithm is much worse than the quality of the A3C algorithm.

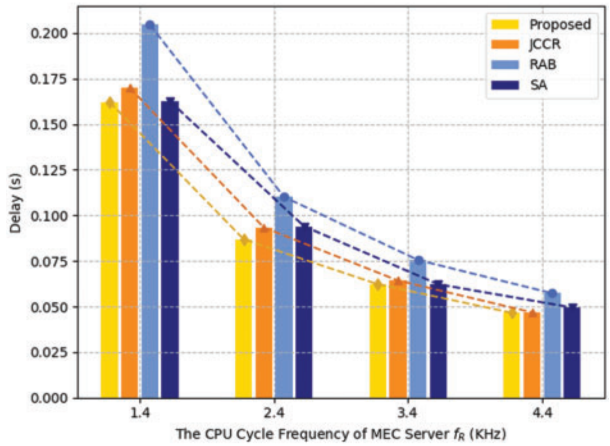


Figure 5: The effect of CPU cycle frequency f_R on delay

From Fig. 6, we can see the impact of different data sizes of tasks on the average reward and the offloading ratio in the proposed algorithm. Fig. 6 shows that, as the D_t increases, the average reward decreases, and the offloading ratio gradually increases. This is because when the D_t increases, more computing tasks will be offloaded to the devices with a lower load state for collaborative computing, which will affect the offloading ratio. At the same time, we can also observe that among all the comparison algorithms, the reward of the proposed algorithm is always highest, followed by JCCR, RAB. The reason is that under the same D_t, the proposed algorithm considers the resource utilization of cluster MEC servers, and offloads more computing tasks to MEC servers with low load state, so that MEC servers are in a load balancing state, which can well improve the load balancing degree. However, the JCCR does not consider load balancing, and the RAB adopts the binary offload method during the offloading process, which leads to uneven resource allocation of the cluster.

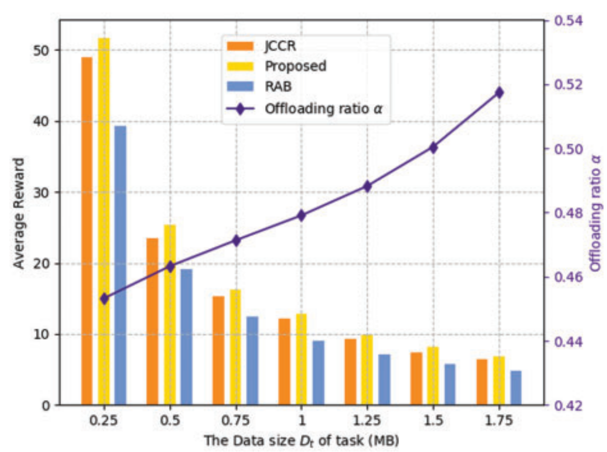


Figure 6: The change of average reward and offloading ratio under different data sizes D_t

6 Conclusion

To solve the security problems in vehicular networks, we propose a trusted edge resource allocation framework for assisted driving service, which includes two stages: the blockchain generation stage and assisted driving service stage. The framework improves the transaction throughput of the blockchain and reduces total latency of the transportation system by establishing the delay-and-throughput-oriented block generation model for mobile terminals. At the same time, we design a balanced offloading algorithm for assisted driving service based on edge collaboration to realize the balanced offloading of the cluster MEC servers. Simulation results show that our algorithm performs better than compared algorithms in average delay. In future work, we will study differentiated service processing to be suitable for more application scenarios, such as IoT, and industrialization scenarios.

Acknowledgement: Thanks for the technical support provided by the State Key Laboratory of Networking and Switching Technology.

Funding Statement: This work has been supported by State Grid Corporation of China Science and Technology Project "Key Technology and Application of New Multi-Mode Intelligent Network for State Grid" (5700-202024176A-0-0-00).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yuxuan Zhong, Siya Xu, Boxian Liao; data collection: Jizhao Lu, Huiping Meng; analysis and interpretation of results: Zhili Wang, Xingyu Chen; draft manuscript preparation: Boxian Liao, Qinghan Li. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: If readers are interested in the experimental data, you can contact with the email: xxuan@bupt.edu.cn.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Yin, B. Chen, K. R. Lai and Y. Li, "Automatic dangerous driving intensity analysis for advanced driver assistance systems from multimodal driving signals," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 4785–4794, 2018.

[2] Y. Yao, X. Chang, J. Mišić, V. B. Mišić and L. Li, "BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3775–3784, 2019.

[3] J. Feng, Z. Liu, C. Wu and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10660–10675, 2017.

[4] Y. Xing, C. Lv, Z. Zhang, H. Wang, X. Na *et al.,* "Identification and analysis of driver postures for in-vehicle driving activities and secondary tasks recognition," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 1, pp. 95–108, 2018.

[5] Z. Lu, W. Liu, Q. Wang, G. Qu and Z. Liu, "A privacy-preserving trust model based on blockchain for VANETs," *IEEE Access*, vol. 6, pp. 45655–45664, 2018.

[6] I. Sorkhoh, D. Ebrahimi, R. Atallah and C. Assi, "Workload scheduling in vehicular networks with edge cloud capabilities," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8472–8486, 2019.

[7] F. Wael, "A lightweight vehicle detection and tracking technique for advanced driving assistance systems," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 3, pp. 2693–2710, 2020.

[8] M. J. Fazel Ashrafi, S. Yousefi, H. Karimi, S. M. Hosseini, H. Rostami *et al.,* "Highway chain collision avoidance using inter-vehicular communications," in *ICCKE 2013*, Mashhad, Iran, pp. 135–140, 2013.

[9] R. Zhang, K. Cai and Y. Huang, "Scheme of autonomous vehicle abnormal behavior detection technology based on edge computing," in *Proc. of HPCCT & BDAI '20*, Qingdao, China, pp. 131–136, 2020.

[10] X. Wang, X. She, L. Bai, Y. Qing and F. Jiang, "A novel anonymous authentication scheme based on edge computing in Internet of Vehicles," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3349–3361, 2021.

[11] P. Wang and W. Susilo, "Data security storage model of the Internet of Things based on blockchain," *Computer Systems Science and Engineering*, vol. 36, no. 1, pp. 213–224, 2021.

[12] H. Bai, C. Wu, Y. Yang, G. Xia and Y. Jiang, "A blockchain-based traffic conditions and driving behaviors warning scheme in the Internet of Vehicles," in *2019 IEEE 19th Int. Conf. on Communication Technology (ICCT)*, Xi'an, China, pp. 1160–1164, 2019.

[13] Z. Yang, K. Yang, L. Lei, K. Zheng and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2019.

[14] D. Zheng, C. Jing, R. Guo, S. Gao and L. Wang, "A traceable blockchain-based access authentication system with privacy preservation in VANETs," *IEEE Access*, vol. 7, pp. 117716–117726, 2019.

[15] Y. Xu, Z. Jin, X. Zhang and L. Zhang, "An optimization scheme for task offloading and resource allocation in vehicle edge networks," *Journal of Internet of Things*, vol. 2, no. 4, pp. 163–173, 2020.

[16] J. Almutairi and M. Aldossary, "Resource management and task offloading issues in the edge-cloud environment," *Intelligent Automation & Soft Computing*, vol. 30, no. 1, pp. 129–145, 2021.

[17] A. Xiong, M. Chen, S. Guo, Y. Li, Y. Zhao *et al.,* "An energy aware algorithm for edge task offloading," *Intelligent Automation & Soft Computing*, vol. 31, no. 3, pp. 1641–1654, 2022.

[18] J. Zhao, Q. Li, Y. Gong and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.

[19] A. Boukerche and V. Soto, "An efficient mobility-oriented retrieval protocol for computation offloading in vehicular edge multi-access network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2675–2688, 2020.

[20] C. Yang, Y. Liu, X. Chen, W. Zhong and S. Xie, "Efficient mobility-aware task offloading for vehicular edge computing networks," *IEEE Access*, vol. 7, pp. 26652–26664, 2019.

[21] J. Feng, F. Richard Yu, Q. Pei, X. Chu, J. Du *et al.,* "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6214–6228, 2020.

[22] Q. Wang, S. Shao, S. Guo, X. Qiu and Z. Wang, "Task allocation mechanism of power Internet of Things based on cooperative edge computing," *IEEE Access*, vol. 8, pp. 158488–158501, 2020.

[23] J. Wang, D. Feng, S. Zhang, J. Tang and T. Q. S. Quek, "Computation offloading for mobile edge computing enabled vehicular networks," *IEEE Access*, vol. 7, pp. 62624–62632, 2019.

[24] Z. Tong, X. Deng, F. Ye, S. Basodi, X. Xiao *et al.,* "Adaptive computation offloading and resource allocation strategy in a mobile edge computing environment," *Information Sciences*, vol. 537, no. 2, pp. 116–131, 2020.

[25] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap *et al.,* "Asynchronous methods for deep reinforcement learning," in *Int. Conf. on Machine Learning*, New York, USA, pp. 1928–1937, 2016.

[26] J. Du, L. Zhao, X. Chu, F. R. Yu, J. Feng *et al.,* "Enabling low-latency applications in LTE-A based mixed fog/cloud computing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1757–1771, 2019.

[27] Y. Kim, H. Lee and S. Chong, "Mobile computation offloading for application throughput fairness and energy efficiency," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 3–19, 2019.

[28] Y. Dai, D. Xu, S. Maharjan and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377–4387, 2019.

[29] B. Zhang, X. Jia, K. Yang and R. Xie, "Design of analytical model and algorithm for optimal roadside AP placement in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7708–7718, 2016.