Hindawi
Complexity
Volume 2022, Article ID 5090875, 17 pages
https://doi.org/10.1155/2022/5090875

Research Article

Research on the Edge Resource Allocation and Load Balancing Algorithm Based on Vehicle Trajectory

Shuxu Zhao, Xinyuan Chen, and Xiaolong Wang

Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

Correspondence should be addressed to Shuxu Zhao; zhaosx_2012@163.com

Received 12 February 2022; Revised 2 April 2022; Accepted 5 April 2022; Published 9 May 2022

Academic Editor: Daniele Salvati

Edge computing empowers the IoV to achieve performance requirements such as low latency and high computational load for in-vehicle services. However, the driving of vehicles is random and unevenly distributed, causing problems such as unbalanced load of edge servers and low edge resource utilization. Therefore, in this article, based on the vehicle trajectories, the edge resource allocation algorithm and load balancing algorithm are used to obtain the load prediction value of the edge server and then calculate the optimal edge resource quantity in order to reduce the resource idleness as much as possible. The experiments demonstrate that the application of the edge resource allocation algorithm and load balancing algorithm based on vehicle trajectory significantly reduces the blocking rate of edge resource requests by vehicles and improves the benefits of the overall IoV edge system.

1. Introduction

The Internet of vehicles (IoV) has emerged as a fundamental technique that substantially pushes forward the development of intelligent transportation [1]. The automotive applications that significantly benefit people's daily activities are increasingly abundant, such as accurate localization, green travel priority plan query, driving judgment, and entertainment information. This may lead to a considerable amount of data and inefficient service quality. With the increasing growth of built-in sensors and actuators, vehicles cannot fulfill the incremental information processing transmitted from the surrounding environment. In general, the IoV services are most delay-sensitive, while vehicles are not integrated with enough computing resources to meet the real-time requirements, leading to the high traffic demands from vehicles to the cloud center. However, the option of requesting to the distal cloud may contribute to the congested backhaul link and unstable connections [2]. As one of the key enabling technologies in the 5G era, the edge computing technology can reduce the cloud computing load and data processing delay by deploying sufficient computing resources as well as adequate storage capacities in the proximity of vehicles. With the configuration of EC, the IoV services can be offloaded to the ENs rather than the remote cloud center, and the transmission latency resulting from the cloud is reduced [3]. However, how much edge resources to deploy is a question, due to the high cost and limited amount of resources deployed at the edge. Therefore, designing an efficient edge resource allocation and selection algorithm will help improve the success rate of the vehicle's request for edge resources.

The authors in [4] propose a proactive resource allocation approach, in which all the cells that are one-hop away from the current cell will be provisioned with resources. The authors in [5] propose to allocate resources to a subset of neighboring cells, in which each cell is marked with different weight values that indicate the handover probability. In [6, 7], a load balancing algorithm between different cells is proposed. This algorithm uses the queueing theory and iterative methods to find the task redirection flow from an overloaded edge server to an unloaded edge server and alleviate the randomness of vehicle driving caused by the unbalanced loads among edge servers and the low resource utilization. In order to incorporate the spatial information of QoS into data prediction and recommendation, the study presented in [8] puts forward a location-aware nonnegative tensor factorization technique, in which a location stamp is attached to each piece of QoS data. The location-aware personalized CF method proposed in [9] leverages both locations of users and web services when selecting similar neighbors for the target user or service. The authors in [10] propose an optimization framework based on stochastic traffic analysis, to minimize the cost of resource provisioning under the premise of ensuring that the service blocking probability is less than a predefined threshold, in order to improve the utilization rate of edge resources. In [11], a citywide and real-time model is proposed for estimating the travel time of any path (represented as a sequence of connected road segments) in real time in a city, based on the GPS trajectories of vehicles received in current time slots and over a period of history, as well as map data sources. In [12, 13], the authors show that each edge server can retrieve data from other edge servers to serve users with a low latency guarantee when another edge server maintains relatively adequate resources in the same area. The authors in [14] propose a selection algorithm between vehicles and edge nodes based on mobile information, server available resources, and Quality of Service (QoS) conditions of service. The algorithm sorts these edge servers around the vehicle based on distance and service delay, in order to reduce the response delay at the edge system of the IoV. Referring to [15], a powerful way to reduce the completion time of a request in the mobile edge environment is to offload its tasks to nearby cloudlets, which consist of clusters of computers. The authors in [16] propose the edge resource allocation algorithm based on the improved vehicle trajectory prediction, the and traffic statistics method is used to obtain the load prediction value of the edge server; then, the optimal number of edge resources is calculated on the premise of reducing resource idleness as much as possible. The authors in [17] propose the task offloading problem from a matching perspective and aim to optimize the total network delay and a pricing-based one-to-one matching algorithm and pricing-based one-to-many matching algorithms for the task offloading. Three load sharing schemes, namely, no sharing, random sharing, and least loaded sharing, are mentioned in [18], which exploit the collaboration between clustered servers in different degrees, and the simulation experiment is conducted by queuing theory.

However, these methods do not integrate the calculation of minimum edge resources with the load balancing among edge nodes. In order to incorporate the keeping of QoS into the allocation and selection methods of IoV edge resources, it is challenging to achieve the trade-offs between reducing the expenditure of edge servers and maximizing the success rate of requesting to edge server by connected vehicle. This article is the first attempt to adjust the optimal edge resource in each edge node. The main contributions are summarized as follows:

(i) An optimization framework is developed in order to minimize the cost of configuring edge resources to compute the minimum amount of edge resources in each cell and in each period, while the service blocking probability is guaranteed to be smaller than a predefined threshold

(ii) A novel method is proposed for adjusting the load balancing of different edge servers and making a distribution of the minimum amount of resource among these edge servers, in order to improve the overall benefits of the edge system of the IoV

(iii) An algorithm is developed for connected vehicles to select edge nodes, and then, a set of experiments are performed to evaluate the validity and efficiency of the edge resource allocation and selection method based on vehicle trajectory prediction

2. Model of the Edge Computation System for IoVs

In order to show the operational process in IoVs, this article assumes the model of edge computation system for IoV as shown in Figure 1. The model is a three-layer architecture, generally including a user layer (data generation), edge layer (data filtering and decision-making), and cloud layer (data fusion, analysis, and macroadjustment).

In Figure 1, the vehicle moves among different adjacent cells. In each cell, there is an amount of edge devices configured servicing passing vehicles covered by the cell, such as roadside base stations (RSU, distributed on both sides of the road) and edge servers. The cloud computing center monitors and adjusts all the operational processes of edge services in all the cells. When a vehicle will soon leave cell A to cell B, the wireless connection between the vehicle and a road site unit (RSU) in cell A interrupts when the connection to B is built.

3. Trajectory Prediction Based on Historical Trajectory Data

Assume a working day is divided into a sequence of periods at equal intervals, where T is the total number of periods. In the interactive environment between the actual vehicle and edge system, the GPS trajectory data can be obtained by the real-time navigation and positioning system, and the future trajectory is predicted based on the fine-grained GPS trajectory data, in order to obtain the movement prediction of the vehicle between different communities. The geometric method is used to predict in advance the location of the vehicle at the beginning of the next period, based on the vehicle's location, speed, and direction:

$$S_n(t) = [S_1(t), S_2(t), \ldots, S_N(t)], \tag{1a}$$

$$\mathrm{SF}_n(t) = [\mathrm{SF}_1(t), \mathrm{SF}_2(t), \ldots, \mathrm{SF}_N(t)]. \tag{1b}$$

Since the number of vehicles covered by a cell in different periods is changing, in order to facilitate the comparison between the experimental prediction results and actual results, the fleet distribution vector is defined as $S_n(t)\,(0 < n \le N; 0 < t \le T)$ and the vehicle-mounted distribution vector is denoted by $SF_n(t)\,(0 < n \le N; 0 < t \le T)$. They are introduced in equations (1a) and (1b), which represents the actual number of vehicles and the total onboard capacity of vehicles at the beginning of period t

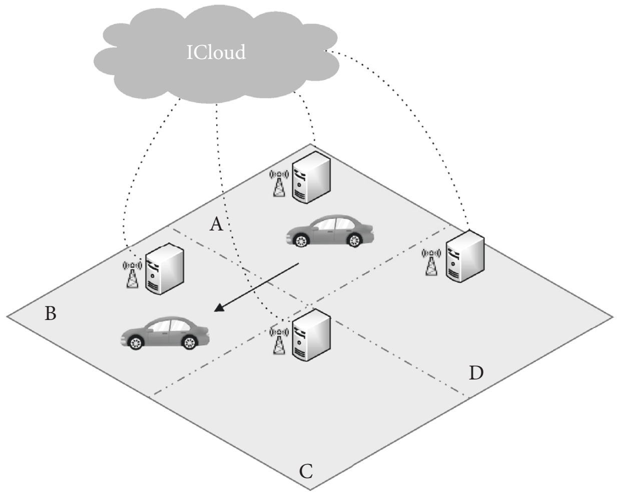

Figure 1: Model of the edge computation system in IoV.

covered by cell n, usually determined by the GPS positioning system. Note that N denotes the total number of cells divided in the area:

$$H(t) := \begin{bmatrix} H_{1,1}(t) & H_{1,2}(t) & \cdots & H_{1,N}(t) \\ H_{2,1}(t) & H_{2,2}(t) & \cdots & H_{2,N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ H_{N,1}(t) & H_{N,2}(t) & \cdots & H_{N,N}(t) \end{bmatrix}, \tag{2a}$$

$$\mathrm{HF} \vdots := \begin{bmatrix} \mathrm{HF}_{1,1}(t) & \mathrm{HF}_{1,2}(t) & \cdots & \mathrm{HF}_{1,N}(t) \\ \mathrm{HF}_{2,1}(t) & \mathrm{HF}_{2,2}(t) & \cdots & \mathrm{HF}_{2,N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{HF}_{N,1}(t) & \mathrm{HF}_{N,2}(t) & \cdots & \mathrm{HF}_{N,N}(t) \end{bmatrix}. \tag{2b}$$

In this article, based on the trajectory prediction of GPS trajectory data, the fleet mobility prediction matrix and vehicle-mounted mobility prediction matrix are obtained, as shown in equations (2a) and (2b), where $H_{i,j}(t)$ is the number of vehicles transferring from cell i to cell j in period t, and $\mathrm{HF}_{i,j}(t)$ represents the vehicle-mounted capacity of moving from cell i to cell j in period t, which is convenient to distribute requests among vehicle RSU and Cloudlet.

4. Analysis of Vehicle Arriving and Leaving

Based on the fleet mobility matrix obtained in Section 3, in order to improve the success rate of the vehicle requesting edge resources in the actual situation, it is necessary to estimate the time point when vehicle v removes from cell n to the neighboring cell n' (i.e., the time when the vehicle reaches the cell boundary). Since it is almost impossible to accurately estimate the vehicle's travel time, this article uses the probability distribution function of the arrival time, which represents the probability distribution of the arrival time of the vehicle from the perspective of probability [14]. Similarly, the probability distribution function of the departure time of the vehicle is used to represent the probability distribution of the departure time of the vehicle.

4.1. Analysis of Vehicle Arriving Flow and Leaving Flow. $P_{n,t}^{arr}$ and $P_{n,t}^{dep}$ represent the probability distribution of the arrival time entering cell n and the probability distribution of the departure time leaving cell n of a vehicle in period t, respectively. In this article, the normal distribution is used to represent the probability distribution of the arrival time and departure time, as shown in Figure 2. In fact, based on the fleet mobility matrix H (cf. Section 3), the number of vehicles accounts for the majority in period t when arriving at cell n, and a small number of vehicles do not arrive according to the results given by the vehicle trajectory prediction model.

Assuming that the arrival time of all the vehicles arriving in cell n in period t follows the same normal distribution, the expected number of vehicles arriving in cell n during period t is denoted by $\lambda_n(t)$. The computation is shown in the following equation:

$$\lambda_n(t) = \begin{cases} 0, & M = 0, \\ \sum_{k=1}^{M} \int_t^{t+1} P_{n,t}^{arr}(t)\mathrm{d}t, & M \neq 0, \end{cases} \tag{3}$$

where each integral represents the probability of each vehicle arriving at cell n in period t, and $M = \sum_{i=1,i\neq n}^{N} H_{i,n}(t)$.

Similarly, the expected number of vehicles leaving cell n during period t is denoted by $\mu_n(t)$:

$$\mu_n(t) = \begin{cases} 0, & M = 0, \\ \sum_{k=1}^{M} \int_t^{t+1} P_{n,t}^{\mathrm{dep}}(t)\mathrm{d}t, & M \neq 0, \end{cases} \tag{4}$$

where each integral represents the probability of each vehicle leaving cell n in period t, and $M = \sum_{j=1,j\neq n}^{N} H_{n,j}(t)$.

As the vehicles arrive and leave, the number of vehicles covered by a cell is changing. The expected arrival/departure number of vehicles cannot be directly used to calculate the optimal number of edge resources for each cell. Because the calculated expected average does not take into account all the possible values, if the actual number of vehicles arriving in cell n exceeds the expected average value, some vehicles may fail to request the edge resources of cell n.

4.2. Dynamic Analysis of Traffic Flow. Due to complex traffic conditions, different driving habits of different drivers and human subjective factors, it is inevitable that the vehicles will have high mobility problems. In fact, it is difficult to accurately calculate the number of resources that the edge devices need to provide. Even a big gap exists between the estimated number of vehicles covered by a cell and the actual number. In order to analyze the flow of vehicles arriving and leaving cell n at period t, the vehicle arriving and leaving flows are modeled as Poisson processes, in which the parameters of the two Poisson processes are $\lambda_n(t)$ and $\mu_n(t)$, respectively.

4.2.1. Vehicle Arriving Flow. A random variable $D_n(t)$ is defined in period t indicating the number of vehicles

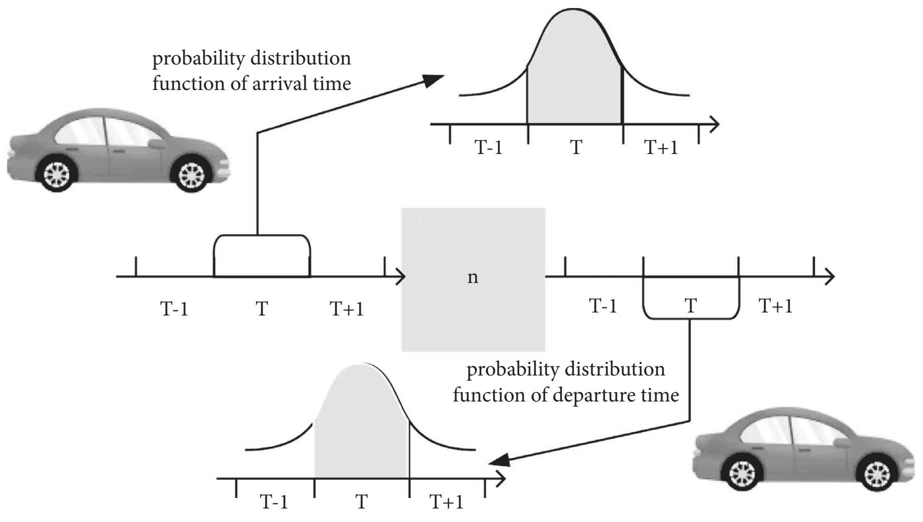

Figure 2: Probability distribution of the arrival/departure time of vehicles in cell n.

arriving at cell n, and aiming at considering the number of all the possible values, as shown in the following equation:

$$P_{D_n(t)}(x) = \frac{(\lambda_n(t))^x e^{-\lambda_n(t)}}{x!}, \tag{5}$$

such that $0 \le x \le \breve{B}_n(t)$, where $\breve{B}_n(t)$ represents the maximum possible number of vehicles during period t arriving at cell n, which is usually the upper bound of $D_n(t)$. The calculation formula is shown in the following equation:

$$\breve{B}_n(t) = \sum_{i=1, i \ne n}^{N} H_{i,n}(t). \tag{6}$$

4.2.2. Vehicle Leaving Flow. A random variable $U_n(t)$ is defined in period t indicating the number of vehicles leaving from cell n, and aiming at considering the number of all the possible values, as shown in the following equation:

$$P_{U_n(t)}(x) = \frac{(\mu_n(t))^x e^{-\mu_n(t)}}{x!}, \tag{7}$$

such that $0 \le x \le \widehat{B}_n(t)$, where $\widehat{B}_n(t)$ represents the maximum possible number of vehicles during period t leaving from cell n, which is usually the upper bound of $U_n(t)$. The calculation formula is shown in the following equation:

$$\widehat{B}_n(t) = \sum_{j=1, j \ne n}^{N} H_{n,j}(t). \tag{8}$$

4.2.3. Vehicle Composite Flow. A random variable $R_n(t)$ is defined to consider both the arriving flow and leaving flow while representing the difference between the number of arriving vehicles and the number of leaving vehicles at period t in cell n (cf. equation (9)). It indicates that the required number of edge resources is greater than the number of edge resources to be released.

$$R_n(t) = U_n(t) - D_n(t). \tag{9}$$

In order to consider all the possible values, a joint probability distribution function $R_n(t)$ is defined to represent the probability distribution of the difference between two independent streams. It is expressed in the following equation:

$$P_{R_n(t)}(x) = \sum_{k=0}^{\widehat{B}_n(t)} P_{U_n(t) D_n(t)}(k, k + x), \tag{10}$$

where $0 \le x \le \breve{B}_n(t)$.

$$P_{R_n(t)}(x) = \sum_{k=0}^{\widehat{B}_n(t)} P_{U_n(t)}(k) P_{D_n(t)}(k + x). \tag{11}$$

Within a short period of time in cell n, the arriving flow can be considered independent of the leaving flow. Therefore, equation (11) can be derived from equation (10).

5. Traffic Flow Prediction Model

Based on the prediction of the future trajectory of the vehicle, forecasting based on historical traffic flow can improve the success rate of the vehicle requesting edge services. In this article, according to the traffic statistics of the vehicle trajectory data, the historical traffic flow of the vehicles exhibits specific laws, such as the periodic similarity and correlation of the traffic flow. The experimental part of the traffic flow statistics is presented in Section 10.2. Although the short-term traffic flow has the characteristics of nonlinearity, uncertainty, and randomness, which brings certain difficulties to the prediction of the traffic flow, its characteristics such as periodic similarity, correlation, and self-organization are convenient for forecasting the traffic volume.

5.1. Characteristics of the Traffic Flow

5.1.1. Periodic Similarity of the Traffic Flow. Due to the living habits of travelers, the overall traffic flow distribution presents specific laws. In addition, the traffic flow of some road sections has a high degree of similarity in two different periods.

5.1.2. Correlation of the Traffic Flow. The correlation of the traffic flow in time is reflected by the fact that the traffic flow value of a community is affected not only by the previous period, but also by the traffic flow of the next period. The traffic flow value of a community in several consecutive periods shows a higher similarity.

5.1.3. Self-Organization of the Traffic Flow. All the vehicles follow the same principle in driving, and they all hope to smoothly and quickly reach the destination in order to maximize the group benefits. This phenomenon is referred to as self-organization.

5.2. Traffic Flow Statistics. According to the cells where a vehicle appears at the beginning of t_1, t_2, and t_3 in t, the traffic volume of each cell in t is counted. If a vehicle appears in the same cell at the beginning of the three small periods, only one vehicle is added to the cell. Otherwise, these cells where a vehicle occurs will be added. Using this method, the number of vehicles in each cell and in each period will be obtained.

5.3. Prediction of the Traffic Flow. Based on the characteristics of the traffic volume $R_n(t) < 0$, the traffic volume value in period t of cell n is related to the statistical average value of the traffic volume of the continuous-time periods of the last week in cell n, the statistical average value of the traffic volume of the continuous-time periods of the previous day in cell n, and the statistical average value of the traffic volume of the recent continuous periods in a certain extent. Since the linear model can be easily modeled and has a high speed and stable operation, the prediction of the traffic volume is solved by the linear model as

$$\begin{aligned} f_{t,n}(\mathbf{x}_{t,n}) &= w_{n1}x_{t,n,1} + w_{n2}x_{t,n,2} + w_{n3}x_{t,n,3} + b_n, \\ x_{t,n,1} &= \frac{1}{2k+1}\sum_{i=t-k}^{t+k} D_1(i,n), \\ x_{t,n,2} &= \frac{1}{2k+1}\sum_{i=t-k}^{t+k} D_2(i,n), \\ x_{t,n,3} &= \frac{1}{k}\sum_{i=t-k}^{t-1} D_3(i,n), \end{aligned} \tag{12}$$

where the input is a vector $\mathbf{x}_{t,n} = (x_{t,n,1}, x_{t,n,2}, x_{t,n,3})^T$, $\mathbf{w}_n = (w_{n1}, w_{n2}, w_{n3})^T$ represents three weight values, $D_1(i,n)$ denotes the traffic volume at period i in cell n of the last week, $D_2(i,n)$ is the traffic volume at period i in cell n on yesterday, $D_3(i,n)$ represents the traffic volume at period i in cell n on today, b_n is a constant, k represents the length of the selected periods, and the output $f_{t,n}(x)$ denotes the estimated traffic volume in cell n during period t.

In order to instantiate the linear model of traffic flow prediction, the dataset in the linear model is constructed using the traffic flow statistics of the vehicle GPS trajectory data used in this article. The optimal value of $\mathbf{w}_n$ and b_n is then obtained using the least squares method. Finally, the generated model is applied to forecast the traffic volume at the next period on cell n based on the existing traffic volume records. In order to simplify the calculation, the dataset is represented by a matrix of $\mathbf{X}_n$ with $U \times 4$ size and a vector of $\mathbf{y}_n$ with $U \times 1$ size, as shown in equations (13a) and (13b), with vector $\widehat{w}_n = (\mathbf{w}_n, b_n)$ absorbing $\mathbf{w}_n$ and b_n. The first three columns of each row in $\mathbf{X}_n$ are $x_{t,n,1}$, $x_{t,n,2}$, and $x_{t,n,3}$ in different periods in cell n, and $y_{i,n}$ in $\mathbf{y}_n$ corresponds to the actual traffic flow at period t in cell n:

$$\mathbf{X}_n = \begin{bmatrix} x_{1,n,1} & x_{1,n,2} & x_{1,n,3} & 1 \\ x_{2,n,1} & x_{2,n,2} & x_{2,n,3} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{T,n,1} & x_{T,n,2} & x_{T,n,3} & 1 \end{bmatrix}, \tag{13a}$$

$$\mathbf{y}_n = \left(y_{1,n}, y_{2,n}, \ldots, y_{T,n}\right), \tag{13b}$$

$$\widehat{w}_n^* = \left(\mathbf{X}_n^T\mathbf{X}_n\right)^{-1}\mathbf{X}_n^T y_n. \tag{14}$$

Different weight vectors $\mathbf{w}$ and the optimal value b for different cells are obtained using equation (14) and then substituted into equation (12) in order to predict the traffic flow of a cell in the next period, so as to improve the load prediction accuracy of the edge device.

6. Resource Allocation Model Based on QoS

Based on the traffic flow prediction model, the QoS model is proposed to achieve a certain degree of success rate of the requesting edge resources by the passing vehicles, according to actual application requirements. This section uses the fleet distribution vector $S(t)$, fleet mobility matrix $H(t)$, vehicle arriving $\lambda_n(t)$, and vehicle leaving $\mu_n(t)$ as input to cell n at period t, in order to predict the number of covered vehicles, and to divide the edge resource in each cell at the end of period t or the beginning of period $t+1$.

6.1. Prediction of the Vehicle Traffic Flow. Assuming that there are $C_n(t-1)$ available edge resources contained in cell n at the beginning of period t, the probability of a request being blocked when the vehicle enters cell n from other cells is expressed in the following equation:

$$P(\varnothing | C_n(t-1)) = \begin{cases} 0, & C_n(t-1) \geq \breve{B}_n(t), \\ \sum_{C_n(t-1)+1}^{\breve{B}_n(t)} P_{R_n(t)}(x), & \text{else.} \end{cases} \tag{15}$$

where ϕ represents a blocking event.

Some of the vehicles appear in cell n at the beginning of period t, and stay in cell n during the entire period t. They will not make changes to the edge resources of the cell, because their edge resource requests have already been provisioned at the beginning of period t. It is demonstrated that these requests have been supplied by cell n, and maintain the supplement of edge resources at the beginning of period t +1. Considering the worst case, if all the entering vehicles are earlier than all the vehicles leaving in cell n at period t, the blocking rate of the edge resource request can be easily increased by directly configuring according to the edge resource configuration result obtained by equation (9).

6.2. Algorithm Improvement Based on the QoS Model. The estimated value of $C_n(t-1)$ should be determined at the beginning of t that has passed. In addition, the accuracy rate of the vehicle transfer prediction for the next period $t+1$ based on the vehicle trajectory records in period t is low. Therefore, the minimum number of edge resources of cell n at the end of period t or the beginning of period $t+1$ is predicted based on the predicted transfer situation of vehicles in period t and the distribution of the fleet at the beginning of period t:

$$P(\varnothing | C_n(t-1)) \leq \epsilon. \tag{16}$$

Therefore, the model based on QoS aims at computing the minimum number of edge resources of cell n at the end of period t, so that equation (16) is developed, where ϵ is the user-defined service blocking rate, which represents the estimated blocking rate of the vehicle requesting edge resources.

6.3. Optimal Resource Allocation. Due to the fact that it is impossible to calculate the optimal value one by one in the case of large data volume, this article uses a grouping and binary search algorithm (two-stage algorithm) in order to calculate the minimum number of edge resources in period t and in cell n. The corresponding pseudocode is shown in Algorithm 1.

Using Algorithm 1, the optimal value of edge resources divided at the beginning of the next period of each cell is estimated. In addition, the optimal resources are allocated to each edge server according to the adjustment of load balance among different edge servers and the allocation of request redirection, in order to increase the success rate of the vehicles requesting edge resources.

6.4. Model Improvement Based on Traffic Flow Prediction. The result $f_{t,n}$ is first obtained from the traffic flow prediction model in cell n and in period t using equation (12):

$$C^{\text{opti}} = \begin{cases} C_{\min}, & f_{t,n} \leq C_{\min} + S_{n-1}(t), \\ f_{t,n} - S_{n-1}(t), & f_{t,n} > C_{\min} + S_{n-1}(t). \end{cases} \tag{17}$$

It is then compared with the optimal value of edge resource C^{opti} derived from Algorithm 1, in order to improve the prediction accuracy.

7. Request Allocation for the Edge System in IoV

Based on the nature of the dynamic connection between the vehicle and the edge node, this section proposes a request allocation model between the on-board terminals, edge servers and Cloudlet, so that the average processing time of each edge server in the cell is as equal as possible, in order to improve the performance of the edge system in IoV.

7.1. Model of Request Allocation for the Edge System in IoV. Assuming that the total amount of messages received by cell n in period t is $\lambda = \lambda^e + \lambda^v + \lambda^c$, it is distributed to edge servers, on-board units (OBUs), and Cloudlet. Three different levels of equipment then process the request sent by the vehicle, and send it back to the vehicle. The distribution table is presented in Table 1.

Due to the uncertainty of user demand and fluctuations in the number of requests received by each RSU in different periods, based on the study presented in [15, 16], the number of arriving vehicles follows the Poisson process. Therefore, this article assumes that the request flow arriving at each RSU from vehicles follows a Poisson process with an arrival rate of λ_i.

This article assumes that N, R, and M represent the number of Cloudlets in the targeted area, number of edge servers covered by each cell, and expected number of vehicles covered in each cell at the beginning of period t +1, respectively. The main terms of this section are presented in Table 2.

In addition, this article first analyzes the network performance in a cell as an example and then continuously adjusts the request allocation algorithm according to the experimental results. Afterwards, it is applied to all the other cells. The allocation graphic is presented in Figure 3, and the Cloudlet in cell n accepts the number of requests that fail in transmitting to edge server or on-board unit sent by vehicle, in order to relieve, which alleviates the emergency need for vehicular requests.

7.2. The Service Queue Model in IoV. In the edge system of IoV, the total response time can be obtained by:

$$P_0(\lambda, \mu, l) = \left[\sum_{k=0}^{l-1} \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k + \frac{1}{l!(1-\rho)}\left(\frac{\lambda}{\mu}\right)^l\right]^{-1}, \tag{18}$$

where d_{up} represents the uploading delay for a request from a vehicle to an RSU, d_{wait} denotes the queueing delay for a request at the edge server wired to the RSU, d_{pro} is the processing delay of the request at the edge server, and d_{down} represents the feedback delay of the calculation result back to the vehicle:

```
    Input: user-defined service blocking rate $\epsilon$, maximum capacity of edge device $C_{\max}$
    Output: Optimal value of edge resource $C_{\min}$ satisfied by (16)
(1)  $C_L, C_H \leftarrow 1$
(2)  while $P(\phi|C_H) > \epsilon$:
(3)      $C_L \leftarrow C_H$
(4)      $C_H \leftarrow 2 \times C_H$
(5)      if $C_H > C_{\max}$ then
(6)          return false
(7)  while True:
(8)      $C_{\min} \leftarrow (C_L + C_H)/2$
(9)      if $P(\phi|C_{\min}) \leq \epsilon$:
(10)         $C_H \leftarrow C_{\min}$
(11)     else:
(12)         $C_L \leftarrow C_{\min}$
(13)     if $C_H - C_L \leq 1$:
(14)         return $C_{\min}$
(15)     else:
(16)         return false
```

Algorithm 1: Two-stage algorithm.

Table 1: Distribution table of the vehicular requests.

λ_i	The amount of vehicular requests received from the coverage by RSUi
λ_i^e	The amount of vehicular requests allocated to ES connected to RSUi
λ_i^v	The amount of vehicular requests allocated to on-board units from the coverage of RSUi
λ_i^c	The amount of vehicular requests allocated to Cloudlet which RSUi belong to

Table 2: Distribution of vehicular requests.

Notation	Description
C	The set of clouds, $C = \{c_1, c_2, \dots, c_N\}$
E	The set of RSU, $E = \{RSU_1, \mathrm{RSU}_2, \dots, \mathrm{RSU}_R\}$
V	The set of vehicles, $V = \{v_1, v_2, \dots, v_M\}$
$f(i, j)$	Representing the number of requests redirected from $\mathrm{RSU}_i, i \in R$ to $\mathrm{RSU}_j, j \in R$

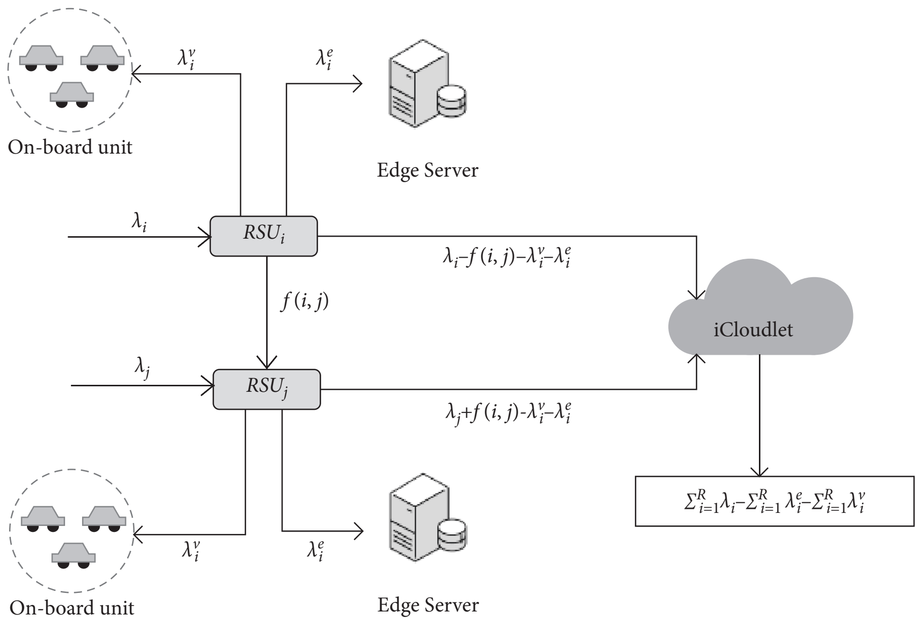
λ_i^v
λ_i^e
On-board unit
Edge Server
λ_i
RSU_i
$\lambda_i - f(i,j) - \lambda_i^v - \lambda_i^e$
$f(i,j)$
iCloudlet
λ_j
RSU_j
$\lambda_j + f(i,j) - \lambda_i^v - \lambda_i^e$
$\Sigma_{i=1}^R \lambda_i - \Sigma_{i=1}^R \lambda_i^e - \Sigma_{i=1}^R \lambda_i^v$
λ_i^v
λ_i^e
On-board unit
Edge Server

Figure 3: Distribution diagram of the vehicular requests.

$$E(d_{\text{wait}}) = f(\lambda, \mu, l) = \frac{(l\rho)^l \rho}{\lambda l!(1-\rho)^2} P_0. \tag{19}$$

The edge system in IoV can be modeled as a queueing network. The idle probability in the queueing system with l homogeneous servers is expressed in equation (18), where λ represents the request flow waiting for a server to process, and μ denotes the fixed service rate so that the service intensity meets $\rho = \lambda/l\mu < 1$ with an average queueing delay expressed in equation (19).

7.2.1. Queueing Model at Cloudlet. The Cloudlet in cell n can be modeled as a $M/M/\mathrm{b}$ queue, with b homogeneous servers and fixed service rate μ_c. Based on the previous assumption, the service intensity at Cloudlet in cell n is $\rho_c = \lambda_c/b\mu_c$, where λ_c represents the number of requests to Cloudlet in the unit interval, and the average calculation delay per request is $E(d_{\text{pro}}) = 1/\mu_c$. A network delay $d_{v\longrightarrow \text{Cloudlet}}$ is incurred by a unit request sent by a vehicle transfer from RSU to Cloudlet, with an ignored feedback delay. The average response delay d^c_{stol} of unit request at Cloudlet can then be computed as

$$E(d^c_{\text{stol}}) = f(\lambda^c, \mu_c, b) + \frac{1}{\mu_c} + d_{v\longrightarrow \text{Cloudlet}}. \tag{20}$$

7.2.2. Queueing Model at the Edge Server. Borrowing ideas from Reference [18], it is assumed that cell n contains a number of RSU clusters $\text{RS} = \{\text{RSU}s_1, \text{RSU}s_2, \ldots, \text{RSU}s_{\text{RS}}\}$, where RS represents the number of RSU clusters, and each cluster contains $b_i, i \in R$ edge servers. A cluster of RSU base stations with close distances is developed to queue up and process the requests sent by the vehicles. Similarly, the cluster can be modeled as a $M/M/b$ queue, and fixed service rate μ_e, with a service intensity $\rho_e = \lambda_e/b\mu_e$, where λ_e represents the number of requests to an edge server in a unit interval, and the average calculation delay per request is $E(\mathrm{d}_{\text{pro}}) = 1/\mu_e$. The delay $d_{v\longrightarrow \text{RSU}}$ is incurred by a unit request transfer from a vehicle to RSU, connected to a corresponding edge server. Afterwards, the average response delay d^e_{stol} of unit request on the edge server can be computed as

$$E(d^e_{\text{stol}}) = f(\lambda^e, \mu_e, b) + \frac{1}{\mu_e} + d_{v\longrightarrow \text{RSU}}. \tag{21}$$

7.2.3. Queueing Model of the Mobile Vehicle Network. This article assumes that each vehicle has an independent computational capacity to share processing requests. Given the number of requests arriving at RSU_i with arrival rate λ_i, the request flow in the waiting queue of moving on-board units covered by RSU_i can be considered a subprocess of the request flow arriving at RSU_i, with an arrival rate $\lambda^v_i \le \lambda^v$. When a vehicle comes into the wireless communication range of RSU_i, it picks up a request at the front of the waiting queue based on the on-board units and finishes processing before leaving the communication range of RSU_i. Based on [10], the request flow in the waiting queue of moving on-board units can be modeled as a Markov chain, and the model can be considered a $M/M/1$ queue. The service intensity on the queue is computed as $\rho^v_i = \lambda^v_i/\mu^v_i$, where the total vehicle capacity is $\mu^v_i = \sum_{j=1}^{a} \mu_j$ with a vehicles covered by RSU_i. The average response delay d^v_{stol} of unit request on the on-board unit is only related to λ^v_i and μ^v_i. Its value is calculated using the following equation:

$$E(d^v_{\text{stol}}) = \frac{1}{\mu^v_i - \lambda^v_i} + d_{v\longrightarrow v}. \tag{22}$$

Here, the delay $d_{v\longrightarrow v}$ is incurred by a unit request transfer from a vehicle to another.

7.3. Allocation of Requests Based on Average Delay Minimization. Due to the uncertainty of the trajectory prediction method based on the historical trajectories, the base stations that the vehicle may be connected to at the beginning of the next period form a cluster in order to alleviate the error caused by the vehicle entering different cells with different probabilities in the next period. Assuming that a vehicle submits requests $\lambda_i, i \in M$ in period t, the average response delay is then calculated using equation (23). This indicates that the request λ_i sent by the vehicle can be divided into three different requests $\lambda_i = \lambda^v_i + \lambda^e_i + \lambda^v_i$ that are calculated and integrated by the on-board unit, the edge server and the Cloudlet, respectively. The values of average response delay $E(d^{t,c}_{i,\text{stol}})$, $E(d^{t,e}_{i,\text{stol}})$, and $E(d^{t,v}_{i,stol})$ depend on the service intensity on different equipment, calculated using equations (20)–(22):

$$E\left(d^t_{i,\text{stol}}\right) = \max\left\{E\left(d^{t,c}_{i,\text{stol}}\right), E\left(d^{t,e}_{i,\text{stol}}\right), E\left(d^{t,v}_{i,\text{stol}}\right)\right\}. \tag{23}$$

Therefore, the optimal edge resource allocation problem in IoV is defined as follows. Under the premise that the minimal value of edge resources $C_{\min}$ in cell n and $\sum_{i=1}^{\text{RS}} b_i$ edge servers is deployed, allocating a certain volume of resources of each edge server so that the average response delay in cell n is minimized. The optimization formula is shown in equation (24), where $M = \sum_{i=1}^{N} H_{i,c}(t-1)$ represents the expected number of vehicles covered by cell n in period t, and $\text{MF} = \sum_{i=1}^{N} \text{HF}_{i,c}(t-1)$ denotes the expected total vehicular capacity covered by cell n in period t:

$$\begin{gathered} \underset{\lambda^e_i, \lambda^v_j}{\arg\min} \frac{1}{M} \sum_{i=0}^{M} E\left(d^t_{i,\text{stol}}\right), \\ \text{s.t.} \begin{cases} \lambda^c + \sum_{i=1}^{\text{RS}} \lambda^e_i + \sum_{i=1}^{M} \lambda^v_j \le \sum_{i=1}^{M} \lambda_i, \\ \lambda^e_i \le b_i \mu_e, \quad i \in \text{RS}, \\ \sum_{j=1}^{M} \lambda^v_j \le \text{MF}, \\ \mu_e = \dfrac{C_{\min} - \text{MF}}{\sum_{i=0}^{RS} b_i}. \end{cases} \end{gathered} \tag{24}$$

Since multiple variables in the optimal allocation problem are tightly coupled, it is complicated to solve the response time and request allocation and redirection problems. Therefore, in this article, two subprocesses of request allocation optimization and request redirection are used to approximate the optimal allocation problem.

8. Load Balancing of the Edge Computing System in IoV

It can be deduced that equation (24) is a nonlinear optimization problem, which is difficult to promptly solve in period t. Therefore, the general idea of the load balancing of the edge computing system in IoV is as follows. (1) According to the requests received in the next period $t+1$ and in cell n, the amount of requests allocated to on-board units and the amount of requests allocated to edge servers are calculated using the incremental cost method. (2) This article first computes the amount of redirected requests at each RSUs in the next period $t+1$ in cell t, making these average response delays among all the clusters in cell n similar to $d_{c,t}^{ave}$. (3) According to the actual trajectory of the vehicle, the vehicle decides to select and unload one of the optimal edge servers set, which is configured with the optimal number of edge resources, or to Cloudlet if the unloading fails. (4) The actual blocking rate of requests for edge resources by vehicle will be calculated, and the minimum number of servers required in Cloudlet is determined.

8.1. Minimization of the Average Response Delay.

8.1. Minimization of the Average Response Delay. Based on the previous assumption, cell n contains RS RSU clusters and the total capacity of M vehicles covered expectantly in period t. Therefore, the total requests add up to $\lambda = \sum_{i=1}^{M} \lambda_i$ generated at cell n in the beginning of period t. These overall requests are allocated through the incremental cost method with the following objective:

$$\min\left(E\left(d_{stol}^{t,e}\right) + E\left(d_{stol}^{t,v}\right)\right),$$
$$\text{s.t.}\begin{cases}\displaystyle\sum_{i=1}^{\mathrm{RS}} \lambda_i^e + \sum_{j=1}^{M} \lambda_j^v \le \lambda,\\ \displaystyle\sum_{j=1}^{M} \lambda_j^v \le \sum_{j=1}^{M} \mu_j^v.\end{cases} \tag{25}$$

According to equation (25), the condition that the number of requests allocated to the on-board units can be completed within a period is satisfied. In addition, the vehicle then unloads the rest of the requests to the Cloudlet, if the unloading at edge server is in failure, using the cost incremental method expressed in the following equation: $\lambda - w$

$$d_{c,t}^{ave} = \arg\min_{w}\left\{ f\left[(\lambda - w), \mu_e, \sum_{i=1}^{\mathrm{RS}} b_i\right] + \frac{1}{\mu^e} + \frac{1}{\sum_{j=1}^{M} \mu_j^v - w}\right\}. \tag{26}$$

Here, w is the number of requests unloaded to the onboard units, and the remaining $\lambda - w$ is the number of requests unloaded to edge servers.

According to the calculation result, the allocation plan is stored in each RSU in cell n, and the edge resource request sent by the vehicle is given as a reference according to the actual vehicle trajectory. The remaining requests $\lambda - w$ are allocated to RSU$s_i, i \in \mathrm{RS}$ according to the GPS historical trajectory. In other words, the vehicle will be the most likely to submit a resource request to the optimal RSU.

8.2. Load Balance among Edge Servers. The number of requests unloaded from vehicles in RSUs$_\mathrm{i}$, i ∈ RS in different periods is unstable, due to the driving randomness, and the states of idleness or busyness are inevitable. Therefore, it is assumed that the RSU*s* can access each other. Even RSU*s* redirect a part of requests contained to another *RSUs*. The average response delay for each RSU*s* in cell n is defined as $E(t_{\mathrm{stol}}(\lambda_i^e))$:

$$E\left(t_{\mathrm{stol}}\left(\lambda_i^e\right)\right) = f\left(\lambda_i^e, \mu_e, b_i\right) + \frac{1}{\mu_e} + d_{v\longrightarrow\mathrm{RSU}}. \tag{27}$$

In order to obtain the number of requests from the overloaded cluster to unloaded cluster between *RSUs*$_i, i \in \mathrm{RS}$, the transmission delay caused by request redirection is ignored. This article assumes that the expected minimum average response delay D in cell n is first calculated, and the number of redirected requests generated from an overloaded RSU*s* to an unloaded RSU*s* is then determined, so that the average response time in cell n is close to D. Finally, the value of D is adjusted according to the difference between the outgoing requests from the total overloaded clusters, and the incoming requests to the total unloaded clusters.

The initial value of D is estimated by $D = (T_{\max} + T_{\min})/2$, where $T_{\max} = \{t_{\mathrm{stol}}(\lambda_i^e)) | i \in \mathrm{RS}\}$ and $T_{\min} = \{t_{\mathrm{stol}}(\lambda_i^e)) | i \in \mathrm{RS}\}$. All the RSU*s* cluster are then partitioned in two disjoint sets: the set of overloaded clusters $Vo = \{i | t_{\mathrm{stol}}(\lambda_i^e)) > D\}$ and the set of unloaded clusters $Vu = \{i | t_{\mathrm{stol}}(\lambda_i^e)) < D\}$. Therefore, the definition of the load balancing problem in IoV is provided in the following equation:

$$\begin{aligned} &\left|E\left(t_{stol}\left(\lambda_i^e - \phi\right)_i\right) - D\right| \le \delta, \quad i \in Vo,\\ &\left|E\left(t_{stol}\left(\lambda_j^e + \phi_j\right)\right) - D\right| \le \delta, \quad j \in Vu,\\ &\text{s.t.} \sum_{i\in Vo} \phi_i = \sum_{j\in Vu} \phi_j. \end{aligned} \tag{28}$$

Here, ϕ_j is the number of the requests redirected from RSU$s_i, i \in V_o$, ϕ_j represents the number of requests redirected into RSUs_j, $j \in V_u$, and δ denotes a given threshold, which indicates the allowable difference between the average response delay after redirection and the value of D. The values of $\phi_i, i \in V_o$ and $\phi_j, j \in V_u$ are approximately calculated using the iterative algorithm.

8.3. The Request Redirection between Edge Servers. Based on the previous results of request redirection, the problem of the number of requests generated from RSU$s_i, i \in V_0$ to RSU$s_j, i \in V_u$ is transformed into a problem of finding the minimum cost and maximum flow in terms of network delay. Thus, the optimization objective is expressed as

$$\underset{r(i,j)}{\arg\min} \sum_{i=1}^{\mathrm{RS}} \sum_{j=1}^{\mathrm{RS}} \left| \max\{r(i,j), 0\} \cdot c_{ij} \right|$$
$$\text{s.t.} \begin{cases} r(i,j) = \begin{cases} -r(i,j), & i \neq j, \\ 0, & i = j, \end{cases} \\ \sum_{i \in Vo} r(i,j) = \phi_j, \quad j \in Vu, \\ \sum_{j \in Vu} r(i,j) = \phi_i, \quad i \in Vo, \end{cases} \tag{29}$$

where $r(i,j)$ represents the number of requests redirected from RSU$s_i, i \in$ RS to RSU$s_j, j \in$ RS, c_{ij} is the transmission delay among different clusters, $c_{ij} = \gamma d_{ij}$, γ is the proportional coefficient between transmission and distance, and d_{ij} is the Euclidean distance between RSUs_i and RSUs_j. This article uses the Vogel algorithm to calculate a suitable value of $r(i,j)$ in time, in a limited period t, based on equation (29), and adjust the optimal volume of edge resource at each RSU according to the optimal value of edge resources C^{opti} in cell n.

Under the precondition that the amount of deployed edge resources remains unchanged, the virtual amount of resources of each edge server can be adjusted according to the redirection result. Especially when the maximum resource amount of an edge server does not meet the requests issued by the surrounding vehicles, a part of the requests can be redirected to another edge server, so that passing vehicles can increase the success rate of unloading to the RSU, within the limited wireless communication range.

9. Dynamic Selection of Edge Nodes by Vehicles

For each edge server, the amount of available resource is in a dynamic change in different period t, with resources continuously occupied and released as the vehicles move. For each connected vehicle, the selection of edge server mainly considers the following factors: the amount of available resource of the edge server, representing the maximum amount of requests that can be offloaded to the edge server, the distance between the vehicle and the edge server, usually affecting the transmission delay of a request, and whether the response delay of the vehicular requests proceeded by the edge server and on-board unit of the vehicle itself can be met within a limited time. This is usually jointly determined by the amount of requests sent by the vehicle and the available resources of the edge server. Therefore it assumes that vehicle v (with vehicular capacity μ_v) decides to offload itself requests (the amount of the requests is λ_q) to the edge server e (the amount of available resources of edge server e is μ_e) at a time point, the delay return function satisfies:

$$\underset{x}{\arg\min}\, R(v,e) = \frac{x}{\mu_v},$$
$$E \begin{cases} 0 < \dfrac{\lambda_q - x}{\mu_e} + d_{v \longrightarrow \mathrm{RSU}} < \dfrac{x}{\mu_v}, \\ x \in \left(0, \lambda_q\right). \end{cases} \tag{30}$$

Here, $R(v,e)$ represents the delay result when vehicle v selects edge server e. The constraint condition indicates that the response delay at the on-board unit is slightly higher than the selected edge server in the actual offload process during driving.

The main thought of dynamic selection for a vehicle to RSU is to first determine the selectable RSU list within a radius of r_d, sorted by distance in ascending order, and then in each iteration selection to determine the delay result $R(v,e_i)$ such that $R(v,e_i) < T_{\mathrm{threshold}}$, where $T_{\mathrm{threshold}}$ usually represents the length of period t, and finally confirms to the offload to the selected RSU or to the Cloudlet. The pseudocode of algorithm 2 for dynamic selection for vehicle v in period t is also provided.

10. Experimental Evaluation

This article uses the vehicle historical trajectory data to conduct experiments and analysis on the proposed edge resource allocation and selection method, and verify its efficiency based on the proposed vehicle trajectory prediction.

10.1. Visualization of Vehicle Historical Trajectory Data. In this article, the trajectory data are extracted from the vehicle trajectory data in Chengdu in November 2016 applied by the Didi platform. The data format of each trajectory record information in the dataset is a four-tuple (id, Tstamp, longitude, and latitude) while recording a GPS vehicle trajectory positioning data every 3 s. Among them, id represents the order ID and Tstamp denotes the timestamp.

The records of order trajectories are first preprocessed, and the boundary is then determined. The final trajectory map is presented in Figure 4, which represents the trajectory of each vehicle from the beginning of the order to its end.

The line represents the trajectory of the vehicle, and the dot denotes the intervals of approximately 1 min. By observing a part of the records, it is deduced that the vehicle trajectory has some common characteristics. For instance, a slow vehicle is more likely to change direction and turn around, while a fast vehicle is more likely to maintain a relatively straight driving state. Based on these trajectory characteristics, geometric methods are used to predict the future trajectory with reference to the historical trajectory of the vehicle.

```
     Input: Distance threshold $r_d$
     Output: offloading decision $r_d$
(1)    Initialize $x_d = 0, i = 1$
(2)    Get connectable RSU set RSU$s = \{e_1, e_2, \ldots, e_L\}$ according to $r_d$, sort by distance in ascending order $S_R = \{e_1, e_2, \ldots, e_L\}$, where
     $L$ is the number of $S_R$
(3)    while $x_d = 0$ do:
(4)      select $e_i \in S_R$:
(5)      if
(6)        $x_d = i$, break
(7)      else $i++$
(8)      else
(9)    else while
(10)   if $x_d = 0$:
(11)     Offload to Cloud in cell $n$
(12)   end if
```

Algorithm 2: Dynamic selection for vehicles to edge servers.

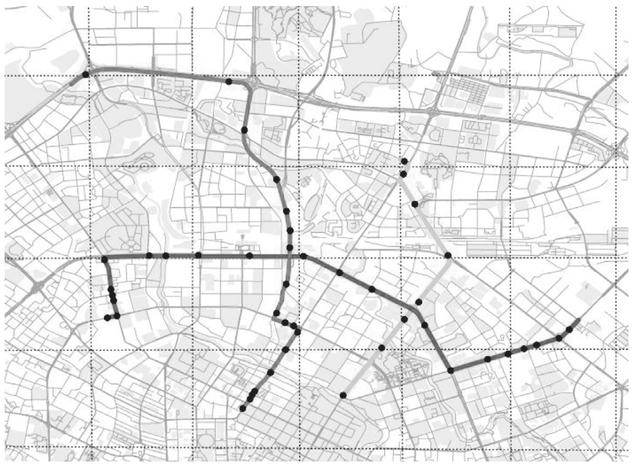
Figure 4: Map of the vehicle trajectory.

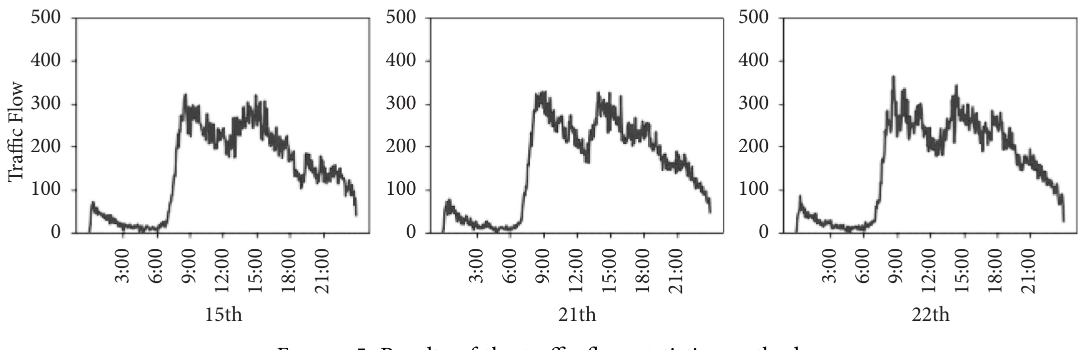

Figure 5: Results of the traffic flow statistics method.

10.2. Statistics of Historical Traffic Volume. By preprocessing the historical trajectory data of vehicles, the traffic flow is calculated in each cell and in each period. The results are shown in Figure 5 representing the traffic flow on the 15th, 22nd, and 23rd of November 2016.

It can be seen from Figure 5 that the traffic flow presents the characteristics of cycle similarity, correlation, and self-organization. It can be deduced that the traffic flow of one period of the day is related to the traffic flow of the same period of the previous week, which is also the same period of the previous day. It is convenient to predict the future traffic flow, based on the historical data of the traffic flow, to a certain extent.

10.3. Experimental Results Based on the Traffic Flow Prediction Model. The traffic flow prediction method proposed in 5.3 is used to predict the traffic flow in the last three weeks of November and compare it with the actual traffic flow statistics. In order to verify the accuracy of the developed traffic flow prediction model, the mean absolute percentage error (MAPE) (cf. equation (31)) and the root mean square error (RMSE) (cf. equation (32)) are used to calculate the actual error of the prediction model:

$$\mathrm{MAPE} = \sum_{i=1}^{N}\sum_{t=1}^{T}\left|\frac{y_{t,i} - f_{t,i}}{f_{t,i}}\right| \times \frac{100}{\mathrm{NT}}, \tag{31}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(f_{t,i} - y_{t,i}\right)^{2}}. \tag{32}$$

The obtained results are shown in Table 3.

10.4. Experimental Evaluation Method of Edge Server Load. In order to verify the efficiency of the proposed edge resource allocation method (Method-3) based on vehicle historical trajectory data, two benchmark methods are used for comparison:

(a) Complete configuration method (Method-1): according to the exact trajectory, the edge resources are deployed in all the cells along the trajectory

(b) Method based on motion estimation (Method-2): based on the vehicle's current motion state (speed, direction, and position), it predicts which cluster the vehicle is most likely to appear in the next period, and configure edge resources for the cluster in advance

10.5. Experimental Process. This experiment first preprocesses the historical trajectory data of vehicle orders from the Chengdu Didi platform in November 2016. This includes the normalizing data values that are conducive to simplifying the input, deleting the orders with too short time, and supplementing the missing values in the middle. The total number of periods in the time and the total number of cells in the space are then calculated. Afterwards, based on the vehicle density distribution obtained from the GPS trajectory data of the vehicle, RSU base stations are randomly set up at different locations in different cells. This article assumes that the network delay between the RSU and the vehicle is proportional to the physical distance, as in [7]. Due to the randomness of the distance between the RSU and the vehicle, a network delay is assigned according to the normal distribution: $0.05 \le N(0.05, 0.02) \le 0.25$. The two benchmark methods and the proposed method are used to allocate optimal edge resource on each edge server, and then, the three methods are tested using the dynamic selection algorithm. Similarly, the network delay is distributed among other nodes according to the normal distribution, except that the Euclidean distance between each pair of RSUs is used as the transmission delay reference standard in the solution of request redirection. The detailed information about the dataset and the parameter settings are provided in Table 4. The experimental environment is presented in Table 5.

Table 3: Error calculation results.

Error index	MAPE (%)	RMSE
Calculation result	23.68	11.21

For each RSU*s* cluster, the expected number of covered vehicles and the total expected vehicle capacity are first calculated in the next period t +1, based on the GPS historical trajectory data. The number of requests for each passing vehicle in the next period $t+1$ is then randomly generated according to normal distribution. Afterwards, the optimal value of edge resource in each RSU is predicted and adjusted using the request allocation algorithm and load balancing algorithm. Finally, the actual success rate of request offloading and blocking rate are calculated using the dynamic selection of surrounding RSU, based on the actual vehicular trajectory.

10.6. Experimental Results. The experimental comparison results between the proposed method and the benchmark methods are shown in Figure 6, with 1169 edge servers. The complete configuration method achieves a 100% success rate of vehicle requesting edge resources. The method based on motion estimation achieves 83.86%, and ϵ of 0.01, 0.05, and 0.1 achieves success rates of 90.97%, 88.85%, and 87.61%, respectively. It can be seen from Figure 6 that the complete configuration method can achieve a rate of 100%; however, the rate is achieved based on the cost of more edge resources.

In order to study the relationship between the actual success rate and ϵ, experiments on continuous values of ϵ for different numbers of edge servers are performed. The obtained results are shown in Figure 7. It can be seen that, as ϵ decreases, all the values slowly increase, and the actual success rate of edge resource maintains more than 90% when $\epsilon \le 0.02$. In addition, the increase in the number of edge servers alleviates the blocking rate of vehicle requests for edge resources to a certain extent, making the experimental results more averaged.

In order to further study the performance of the proposed edge resource allocation method, its stability is observed from the three perspectives of time, area, and number of vehicles.

In order to observe the experimental performance in different periods, vehicle trajectory data from the 17th (the number of edge servers is 1169) are considered, and the actual request success rate in different periods is observed. The experimental results are shown in Figure 8, where the horizontal line represents the average for the whole day, the value of which is 90.97%. It can be seen that the actual

Table 4: Settings of the dataset and parameters.

Parameters	Value
The length of a cell a	1.5 km
The length of T	3 min
The length of time window k	10
The amount of request per vehicle and per period λ_i	$U(1.6, 3.2)$
The number of RSU clusters in each cell b	9
The number of edge servers in each RSU cluster	[1 – 9]
The service rate an on-board unit μ_v	$U(0.5, 1.0)$
The service rate on Cloudlet μ_c	5
The transmission delay between vehicle and RSU $d_{v\longrightarrow \text{RSU}}$	$N(0.05, 0.02)$
The transmission delay between each RSU $d_{\text{RSU}\longrightarrow \text{RSU}}$	$N(0.15, 0.05)$
The transmission delay between each vehicle $d_{v\longrightarrow v}$	$N(0.1, 0.05)$
The transmission delay between vehicle and Cloudlet $d_{v\longrightarrow \text{Cloudlet}}$	$N(0.3, 0.05)$
δ	0.05
γ	0.3
Destination threshold r_d	0.5 km

Table 5: The experimental environment.

Setting	Description
CPU	Intel(R) Core(TM) i7-5500U CPU @2.40 GHz 2.40 GHz
RAM	16 GB
Operating system	Windows 10, 64-bit
Development	Python 3.9.7
Operating platform	PyCharm 2019.2.1

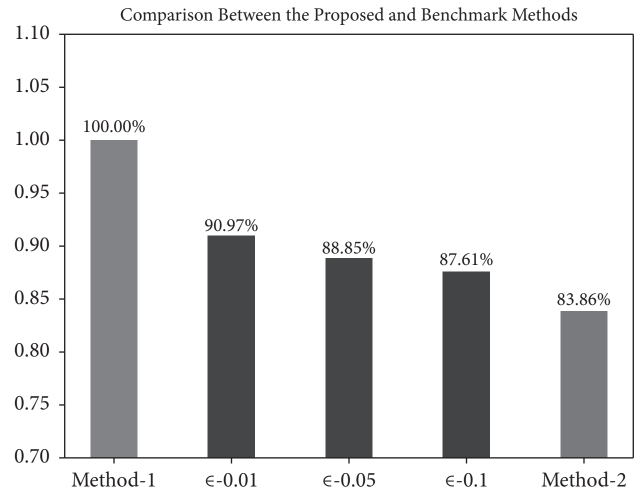

Figure 6: Comparison between the proposed and benchmark methods.

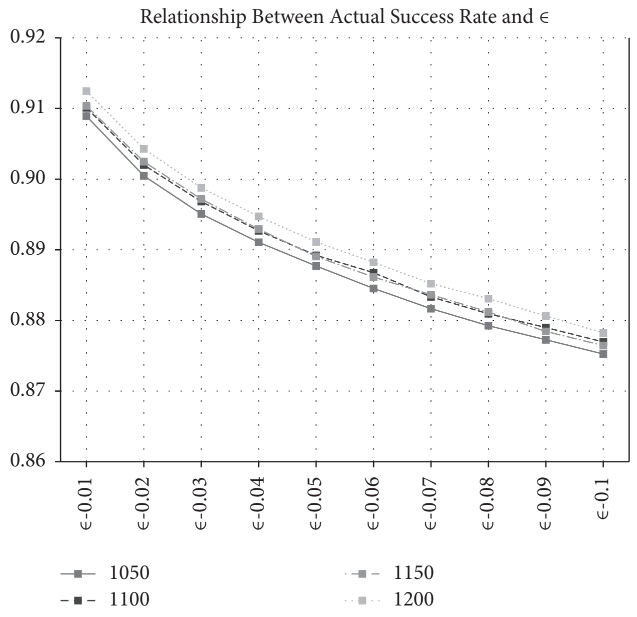

Figure 7: Relationship between the actual success rate and ϵ.

request success rate is low during the period of 17 : 00–19 : 00 and high during the period of 23 : 00–03 : 00 (+1). This is due to the fact that the increase in the number of vehicles and the complex traffic conditions during the peak traffic hours in the evening lead to a low efficiency of actual resource allocation. Moreover, the proposed method has a better performance than that of the reference method of motion estimation in terms of edge resource allocation. In addition, the actual edge resource request success rate gradually increases with the decrease in ϵ.

In order to observe the impact of different numbers of vehicles on the utilization performance of edge resources, a different number of vehicle subsets from the trajectory datasets on the 17th are randomly selected. The edge resources are allocated to different subsets, under the conditions of different values of ϵ with 1169 servers. The obtained results are shown in Figure 9.

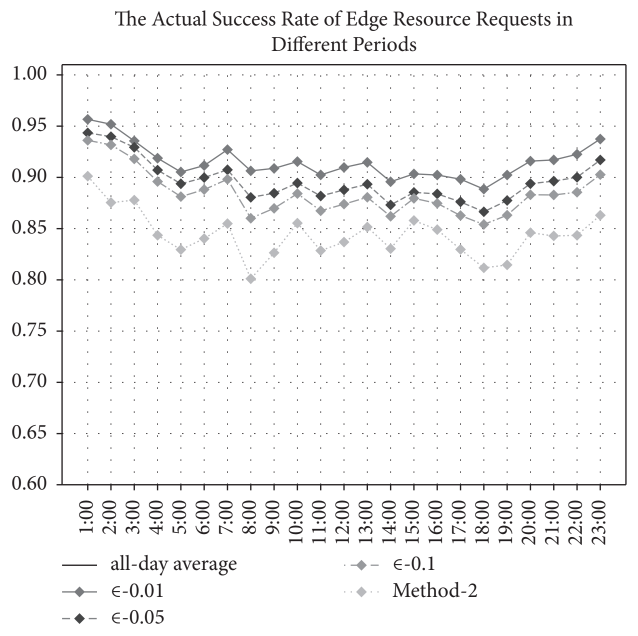

Figure 8: The actual success rate of edge resource requests in different periods.

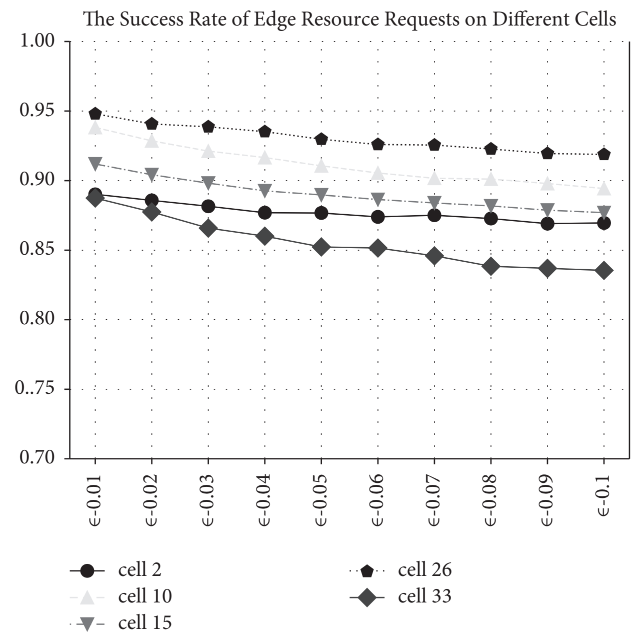

Figure 10: The success rate of edge resource requests on different cells.

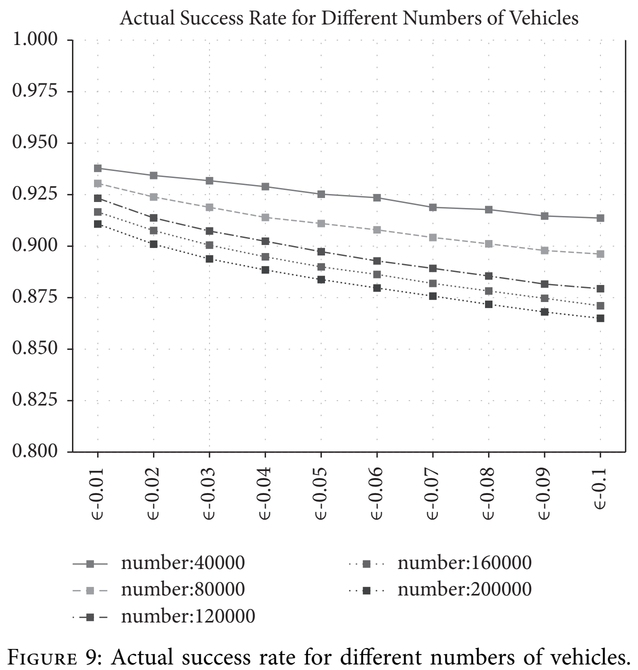

Figure 9: Actual success rate for different numbers of vehicles.

Figure 9 demonstrates that, with the increase in the user's predefined blocking rate, the actual success rate of different vehicle numbers gradually decreases. When $\epsilon = 0.01$, the experimental results of different numbers are the most similar. In addition, with the increase in ϵ, the difference between the actual edge resource request success rates of different numbers gradually increases, and the experimental results are more unstable. This is mainly due to the high mobility of driving behavior, which is prone to the idle state of edge resources and the blocking of actual resource requests. As the value of ϵ decreases, the deviation caused by high mobility can be better alleviated.

In order to evaluate the performance of the proposed edge configuration method in different cells, 5 cells are randomly selected on the 17th, and the relation between the actual request success rate and ϵ is assessed. The experimental results are shown in Figure 10.

It can be seen from Figure 10 that the results of the actual request success rate calculated from different cells are similar. In addition, they increase as the user predefined blocking rate decreases. This is mainly due to the difference in traffic flow in different cells. For cells with less traffic flow, such as the actual resource request success rate in cell 33, the success rate of the actual resource request greatly varies. Moreover, the traffic flow of cell 2 is relatively large, so that the calculation result is relatively stable and less affected by the change in the user's predefined blocking rate.

In order to study the influence of different numbers of edge servers on the utilization performance of edge resources, the proposed algorithm is implemented for different numbers of edge servers. The obtained results are shown in Figure 11.

Figure 11 shows that the results of the success rate of edge resource requests calculated by different numbers of edge servers are similar. In addition, they increase with the increase in the number of edge servers. The proposed method proposed is better than the mobile estimation, in terms of the actual success rate of edge resource requests. This is due to the fact that with the increase in the number of edge servers, the deployment is relatively balanced, which alleviates the prediction error caused by the high mobility of the driving, to a certain extent.

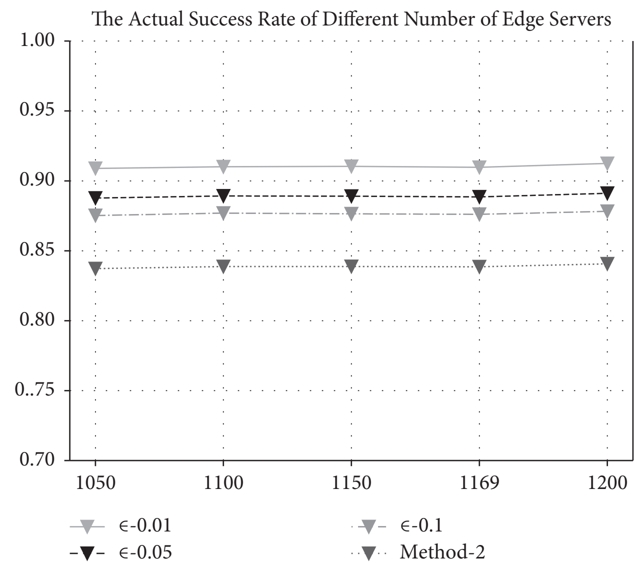

Figure 11: The actual success rate of different numbers of edge servers.

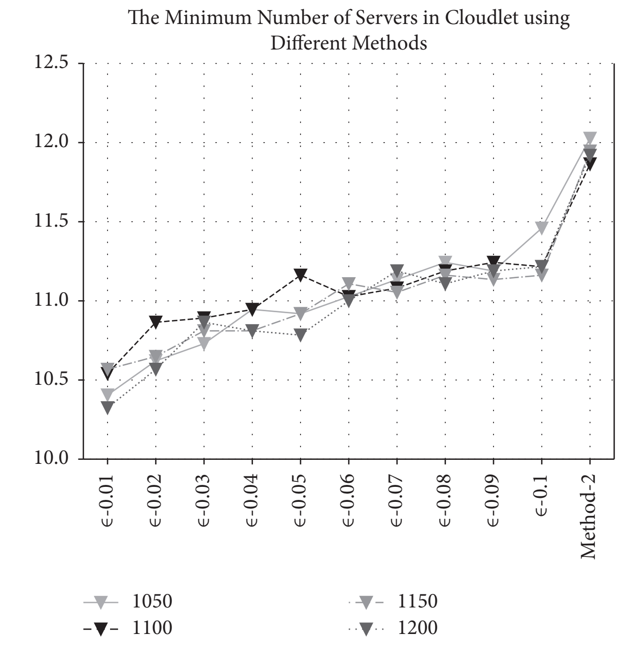

Figure 12: The minimum number of servers in Cloudlet using different methods.

Based on the experimental results obtained from the three perspectives of user's predefined blocking rate, number of vehicles, and region, the proposed method allows to obtain the minimum number of cloud servers in different cells and alleviate the emergency needs of the request after failing to offload to edge devices.

In order to study the relationship between the number of servers in Cloudlet and different edge resource allocation

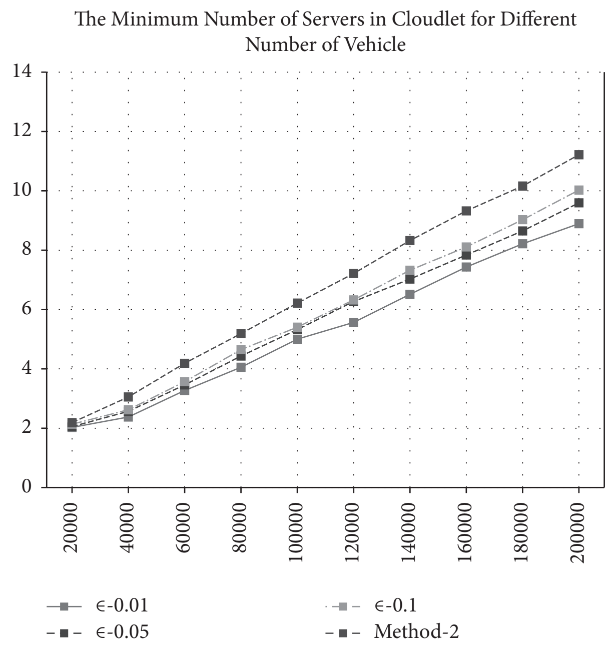

Figure 13: The minimum number of servers in Cloudlet for different numbers of vehicles.

methods, experiments on continuous values of ϵ under the condition of different numbers of edge servers are performed. The obtained results are shown in Figure 12. It can be seen that, as the value of ϵ decreases, the number of cloud servers slowly increases, and the average value becomes close to 11. The resultant value based on the motion estimation method is almost 12.

In order to evaluate the influence of different numbers of vehicles on the minimum number of servers in Cloudlet, subsets with different numbers of vehicles are randomly selected from the trajectory dataset on the 17th, and edge resource allocation is performed on different subsets under different values of ϵ and motion estimation methods, with 1169 servers. The obtained results are shown in Figure 13.

Figure 13 shows that, as the number of vehicles increases, the minimum number of servers in Cloudlet that need to be deployed gradually increases, and the gap between the different values of ϵ gradually increases. This is mainly due to the fact that, when the value of ϵ is determined, the more the number of vehicles is, the more requests blocked are sent by vehicles, and the more cloud servers are required.

The main conclusions are summarized based on the experimental results. (1) The motion estimation method seemingly fail to be applied in practice, and the method proposed in this article not only reduces the redundant cost of resource allocation but also approaches the result of the complete configuration method. (2) The method proposed in this article can effectively improve the actual use efficiency of edge servers, thereby reducing the deployment cost of servers in Cloudlet. (3) The method proposed in this article conducts the optimization in resource provisioning in each period while ensuring QoS service quality.

11. Conclusion

In this article, starting from the demand for edge resources of connected vehicles and the resource allocation benefits of the entire edge system, an edge resource allocation and selection method based on vehicle trajectory prediction is proposed. The proposed algorithm fully considers the high mobility of vehicles and the characteristics of historical traffic flow. It divides the minimum number of edge resources and deploys minimum servers in Cloudlet on the premise of not being higher than the user's predefined blocking rate. The experimental results demonstrate that the proposed algorithm has a high performance in edge resource utilization and a high prediction accuracy.

The future work should include the upgrading of mobility estimation strategy and requests allocation method between on-board-unit, edge server and cloud. The upgrading of mobility estimation strategy is replacing the cell in which a connected vehicle most likely appear with $k(k>2)$ cells and weight these different cells. Another possible future goal consists in improving the success rate of edge resource requests using the proposed method by dynamic allocation of edge server clusters, according to the driving conditions of connected vehicles and the load conditions of edge servers.

Data Availability

Gaia Data Open is only open to universities and scientific research institutions in China. The dataset is for scientific research use only and is strictly prohibited from being disseminated or used by others. Data have been desensitized. Please contact gaia@didiglobal.com using your school/research institution email for further.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Key R&D Program of Gansu Province (20YF8GA123).

References

[1] L. Zhou, W. Min, D. Lin, Q. Han, and R. Liu, "Detecting motion blurred vehicle logo in IoV using filter-DeblurGAN and VL-YOLO," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3604–3614, 2020.

[2] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5213–5222, 2021.

[3] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for Internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[4] A. Gaddah and T. Kunz, "Extending mobility to publish/subscribe systems using a pro-active caching approach," *Mobile Information Systems*, vol. 6, no. 4, pp. 293–324, 2010.

[5] S. Pack, H. Jung, T. Kwon, and Y. Choi, "SNC: a selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks," *ACM SIGMOBILE - Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 39–49, 2005.

[6] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2866–2880, 2016.

[7] X. Wang, Z. Ning, and L. Wang, "Offloading in Internet of vehicles: a fog-enabled real-time traffic management system," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4568–4578, 2018.

[8] H. E. Wei, L. Cui, G. Ren, Q. LI, and T. LI, "A New Paradigm for Personalized Mashup Recommendation Based on Dynamic Contexts in mobile Computing environments," *Scientia Sinica Informationis*, vol. 46, no. 6, pp. 677–697, 2016.

[9] J. Liu, M. Tang, Z. Zheng, X. Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for web service recommendation," *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 686–699, 2016.

[10] G. Tang, D. Guo, K. Wu, F. Liu, and Y. Qin, "QoS guaranteed edge cloud resource provisioning for vehicle fleets," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5889–5900, 2020.

[11] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 25–34, ACM, NY, USA, August 2014.

[12] Z. Xu, L. Zhou, S. C.-K. Chau, W. Liang, Q. Xia, and P. Zhou, "Collaborate or separate? distributed service caching in mobile edge clouds," in *Proceedings of the*, pp. 2066–2075, Toronto, ON, Canada, July 2020.

[13] X. Xia, F. Chen, Q. He, J. Grundy, M. Abdelrazek, and H. Jin, "Online collaborative data caching in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 281–294, 2020.

[14] L. Pacheco, H. Oliveira, D. Rosario, E. Cerqueira, L. Villas, and T. Braun, "Service migration for connected autonomous vehicles," in *Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, Rennes, France, July 2020.

[15] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *To Appear in IEEE Transactions on Cloud Computing*, vol. 5, pp. 725–737, 2015.

[16] S. Cao, Y. Zhang, P. Ding et al., "Research on Edge Resource Allocation Method Based on Vehicle Trajectories Prediction," in *Proceedings of the 2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pp. 200–206, Shenyang, China, November 2021.

[17] P. Liu, J. Li, and Z. Sun, "Matching-based task offloading for vehicular edge computing," *IEEE Access*, vol. 7, Article ID 27640, 2019.

[18] L. Liu, S. Chan, G. Han, M. Guizani, and M. Bandai, "Performance modeling of representative load sharing schemes for

clustered servers in multiaccess edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4880–4888, 2019.

[19] J. He, L. Cai, P. Cheng, and J. Pan, "Delay minimization for data dissemination in large-scale VANETs with buses and taxis," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1939–1950, 2016.

[20] H. Zhu, M. Li, L. Fu, G. Xue, Y. Zhu, and L. M. Ni, "Impact of traffic influxes: revealing exponential intercontact time in urban VANETs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1258–1266, 2011.