WILEY | Hindawi

Research Article

Deep Learning-Based Service Scheduling Mechanism for GreenRSUs in the IoVs

Jitong Li,[1,2] Chao Wang[iD],[1] Daehee Seo,[3] Xiaoman Cheng,[1] Yunhua He,[1] Limin Sun,[4] and Ke Xiao[1]

[1]*School of Information Science and Technology, North China University of Technology, 100144, China*
[2]*School of Computing, Beijing University of Posts and Telecommunications, 100876, China*
[3]*College of Convergence Engineering, Sangmyung University, 03016, Republic of Korea*
[4]*The Institute of Information Engineering, Chinese Academy of Science, 100093, China*

Correspondence should be addressed to Chao Wang; wangchao.andy@gmail.com

Green roadside units (RSUs), also called renewable energy-powered RSUs, are utilized recently rather than the traditional electric-powered RSUs with high power consumption and the large infrastructure deployment cost in the Internet of vehicles (IoVs). However, the power of the green RSUs is limited and unstable, which is affected by the battery size and charging environment. Therefore, a big challenge to deploy green RSUs in the IoVs is to schedule their service process properly, in order to extend the service efficiency of RSUs. In this paper, a deep learning-based communication scheduling mechanism is proposed regarding the service scheduling problem. In particular, a three-part scheduling algorithm consisting of RSU clustering, deep learning-based traffic prediction, and a vehicle access scheduling algorithm is presented to maximize the service number of vehicles and minimize the energy cost. An extensive simulation is done, and the simulation results indicate that our algorithm can serve more vehicles with less energy consumption compared with other scheduling mechanisms under different scenarios.

1. Introduction

The Internet of vehicles (IoVs) is composed of lots of vehicles and RSUs. These vehicles are equipped with on-board units (OBUs) including a global positioning system (GPS) and the RSUs are connected to the centralized network or service providers [1]. In the IoVs, each vehicle can communicate with each other at the ad hoc style with the DSRC technique or Pc5 interface of LTE-V. This kind of communication is referred as vehicle-to-vehicle (V2V) communication. Besides, these vehicles can also communicate with RSUs with the interface of LTE-V [2] and this kind of communication is commonly referred to vehicle-to-infrastructure (V2I) communication. Based on the two styles of communication, most of services in the IoVs can be carried out, such as location services [3], blockchain services [4], social services [5], and crowd sensing services [6].

In the IoVs, the RSUs play important roles. They not only extend the service range of the IoVs but also provide high-speed message forwarding services for vehicles with high bandwidth links to the service providers [7]. However, the forwarding services require high power consumption [8]. In addition, the RSUs are powered by physical infrastructures linked to the electrical grid, which costs too much. Especially in some rural areas, the cost of deploying electrical facilities for RSUs may be much higher. At last, traditional electric-powered RSUs cause carbon dioxide emissions indirectly. Therefore, the requirement of renewable power, such as being polar and windy, to be integrated into RSUs instead of the traditional powered functions is proposed by several papers [9].

However, there are some critical problems to be solved while deploying the renewable energy-powered RSUs. First, the power on the RSU is limited due to the battery size

and charging environment. In addition, the power consumption of communication is influenced by the communication time, communication frequency, communication distance, and channel quality. Without a proper communication scheduling strategy, the power may be exhausted quickly, which will have a serious impact on the normal development of services in the IoVs. Thus, this paper focuses on how to maximize the performance of renewable energy-powered RSUs in order to serve as many vehicles as possible with less energy consumption.

Motivated by this, a deep learning-based communication scheduling mechanism is proposed in this paper, composed of three parts. First, a clustering method named t-SNE is utilized to classify all the RSUs to several clusters, each of which has a special evolving rule on traffic flows. Second, we use a deep learning-based algorithm to predict the future traffic flow for each RSU cluster, with the history knowledge of the traffic status under each RSU. At last, once the prediction results are obtained, a service ability scheduling algorithm and a priority-based RSU access algorithm are proposed to schedule the vehicle access order. The contributions of this paper are outlined as follows:

(1) A clustering method t-SNE is utilized, based on which the future traffic flow for each RSU can be a prediction with a deep learning algorithm

(2) Based on the future traffic prediction, we propose a service ability scheduling algorithm and a priority-based RSU access algorithm to maximize the RSU service efficiency

(3) An extensive simulation is deployed to demonstrate that the proposed algorithm can achieve higher efficiency with less energy cost compared with other scheduling mechanisms under different scenarios

The rest of the paper is organized as follows. Section Related Work summarizes the related work. Section System Model presents the system model and problem formulation. Section Deep Learning-Based Communication Scheduling Algorithm details the deep learning-based communication scheduling mechanism. And section Simulations presents the simulation results. Lastly, section Discussion and Conclusion draws the conclusion.

2. Related Work

It is necessary to reduce energy consumption as much as possible for RSUs in the IoVs, and several papers have done some work referring to the grid-powered RSUs. In [10], the authors focus on the hardware design of RSUs and propose a centralized control module to schedule the energy cost on RSUs. Their idea is based on the artificial neural network and is composed of three algorithms to achieve the purpose of the green scheduler. In [11], the authors incorporate a sleep mechanism on grid-powered roadside units to further reduce power consumption. However, it is not fit for the renewable energy-powered RSUs. The off-grid wind-powered RSUs are considered in [12]. Varied from conven-

tional reliability analysis, the reliability in this paper is redefined in a highway environment. Then, the energy models are developed and the minimum battery size can be determined when a certain reliability standard and quality of service are achieved.

Due to the high cost of grid-powered RSUs, the renewable energy-powered RSUs are proposed. As the energy on this kind of RSUs is limited, how to minimize the energy consumption is also a wide concern. An energy-efficient scheduling framework is proposed in [13]. The task scheduling and energy consumption are considered jointly in the proposed heuristic algorithm. To minimize energy consumption and satisfy task latency constrains, an imitation learning-enabled online task scheduling algorithm is proposed in [14]. In [15], it is assumed that any vehicle's position in the network is strongly deterministic. Then, three energy-efficient online traffic scheduling algorithms are introduced to minimize the long-term power consumption subject to the communication requests associated with the passing vehicles.

The scheduling problem in renewable energy-powered RSUs is also a hot research topic. A joint scheduling and power control scheme is proposed in [16]; it is formulated as a mixed-integer nonlinear programming (MINLP) problem. Paper [17] investigates the problem of scheduling the downlink communication for renewable energy-powered RSUs toward vehicles, with the objective of maximizing the number of served vehicles. In [18], the authors propose low-complexity algorithms for downlink traffic scheduling in green vehicular roadside infrastructure. However, the algorithms require some priori information, which may not be always available in the IoVs. In [19], a reinforcement learning technique for optimizing downlink scheduling is proposed in an energy-limited vehicular network. Its objective is to equip RSUs with the required artificial intelligence to realize an optimal scheduling policy that will guarantee the operation of the vehicular network during the discharge cycle while fulfilling the largest number of service requests. However, works [17, 19] just simulate the scenario in the simulator without testing the algorithms with vehicle trajectories of reality.

Moreover, some works have been done from the perspective of services to save energy of green RSUs. In a green communication scenario, the safety and QoS are both a concern in [20]. A deep reinforcement learning model named deep Q-network is proposed, which learns an energy-efficient scheduling policy. With the policy, the battery of an RSU is extended and the safety of environment is promoted while the quality of service (QoS) levels is met. Aiming at the problem of power deficiency in solar-powered roadside units (SRSUs), the challenge of QoS loss is addressed in [21], in which a two-phase approach is proposed. With the purpose of energy consumption and time delay guarantee, a distributed packet scheduling optimization strategy is proposed in [22] for the renewable energy-powered RSUs. Based on the proposed strategy, the system energy consumption can be minimized and the delay of the system can also be reduced, in which an optimization model is established based on the Lyapunov theory. To solve
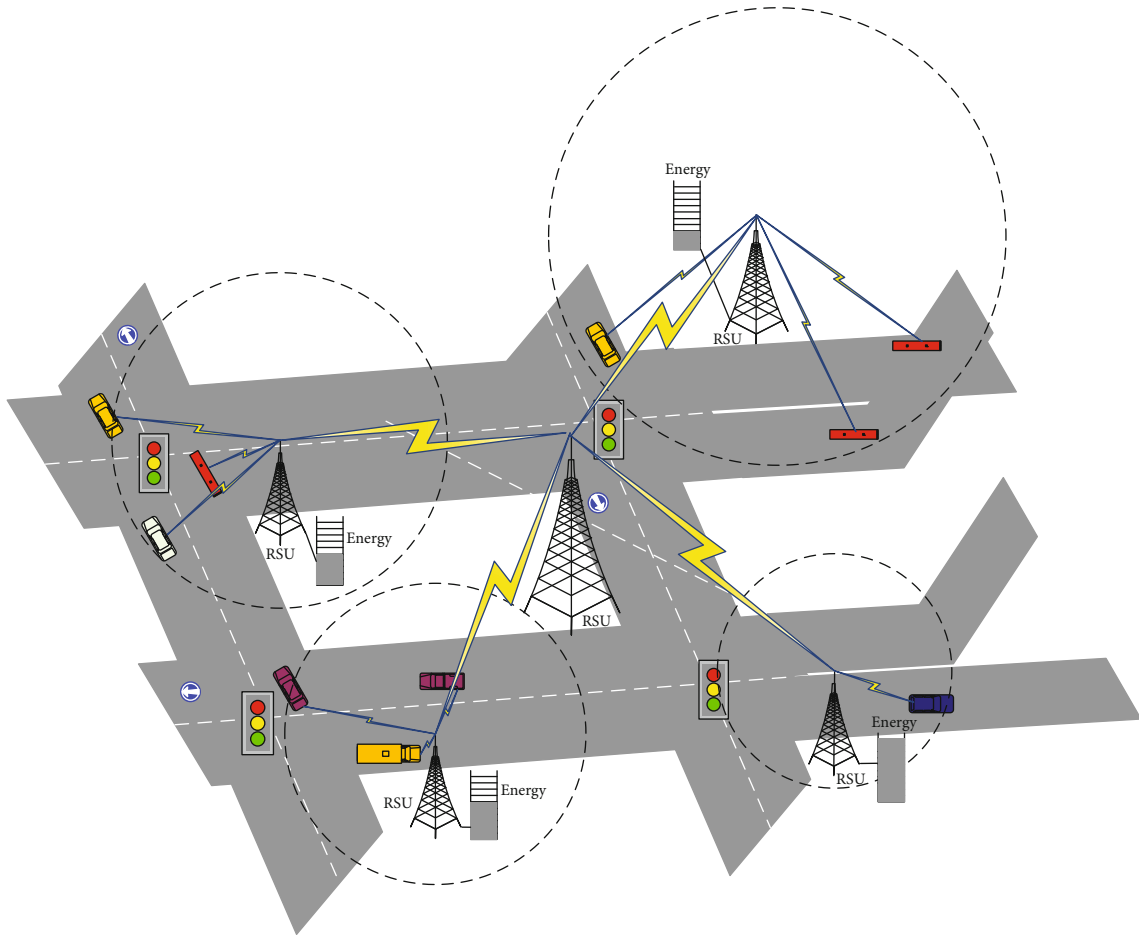
FIGURE 1: Solar-powered RSUs in the scenario of IoVs.

the problems that existed in battery-enabled RSUs and electric vehicles (EVs), an intelligent energy-harvesting framework is constructed in [23], where RSUs and EVs are integrated. Based on a three-stage Stackelberg game developed by the authors, the utilities of RSUs and EVs are maximized.

3. System Model

3.1. Scenario Description. We consider an Internet of vehicles scenario, shown as Figure 1. In the scenario, all the vehicles are equipped with a DSRC or LTE-V communication module, with which each vehicle can interact with the ITS service center. Besides, any vehicle accessing to the centralized network must connect to an RSU with its communication module. Due to consideration of energy saving, more and more RSUs are powered by renewable energy such as solar energy. Thus, the energy on RSUs is limited and each RSU can only serve finite number of vehicles. Thus, in this paper, we focus on the communication scheduling mechanism of RSUs to maximize the number of service vehicles.

3.2. Problem Formulation. Based on the above scenario, we construct the system model as an undirected graph $G(R, E, V)$, shown in Figure 2. Figure 2 is a formalized definition
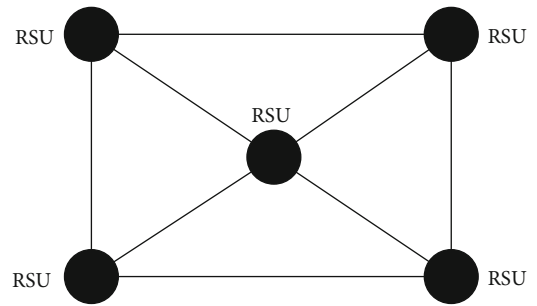

FIGURE 2: System model.

of the described scenario. RSUs in Figure 1 are mapped to the vertexes in Figure 2, and the communication links among RSUs are described as the edges between vertexes. Graph G denotes the Internet of vehicles scenario and the details of the network are described as follows:

(1) R is the set of RSUs, denoted as $R = \{r_1, r_2, \cdots, r_n, \cdots, r_M\}$. In the network, there are M RSUs and everything can be accessed by vehicles passing by. With the assumption that each RSU is powered by renewable energy, the communication bandwidth of each
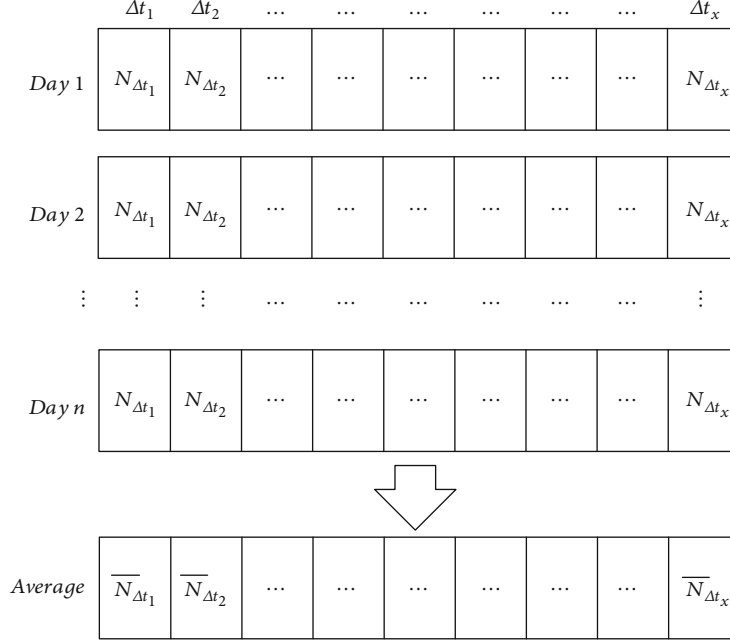
FIGURE 3: Data preprocessing.

RSU can be tuned with time. In this paper, S_{r_n} is the service ability of RSU $r_n(r_n \in R)$ and its value is determined by the energy charged, with $\max_{r_n}$ being the maximum value of S_{r_n} determined by the battery size. Here, we assume that S_{r_n} is an integer and it denotes the number of vehicles that r_n can server. For any RSU r_n, each communication with a vehicle will cost some service ability of r_n, which is assumed to be 1 in this paper. If the time period is considered, then S_{r_n} can be rewritten as $S_{r_n}^{\Delta t}$, which means the service ability of r_n at the time period Δt. Besides, in order to maintain the service ability at a time period, some energy still will be costly. In other words, if there is no vehicle accessing, the RSU still spends service ability on maintaining the communication service. Here, another assumption is made that if no vehicle uses the allocated service ability in the time period, the service ability will be wasted to maintain the service, and the amount of service ability is the same as the communication cost to vehicles

(2) An edge is denoted as $E_{ij}(E_{ij} \in E)$ that connects two RSUs r_i and r_j. It means that r_i and r_j are adjacent RSUs and a vehicle can move from the coverage range of r_i to that of r_j

(3) V is the set of vehicles, denoted as $V = \{v_1, v_2, \cdots, v_i, \cdots, v_N\}$. For any vehicle $v_i \in V$, it moves from the coverage of one RSU to another RSU, and at any time, it can only communicate with no more than one RSU. The trajectory of vehicle v_i is L_{v_i}, which is a location sequence, such as $(r_1^{\Delta t_1}(v_i), r_2^{\Delta t_2}(v_i), r_3^{\Delta t_3}(v_i), \cdots, r_n^{\Delta t_x}(v_i))$. Here, $r_n^{\Delta t_x}(v_i)$ means that vehicle v_i

at time period Δt_x is covered by RSU r_n. Besides, we define another term $S_{r_n}^{v_i}(\Delta t_x)$. If $S_{r_n}^{v_i}(\Delta t_x)$ is 1, RSU r_n serves vehicle v_i at time period Δt_x successfully; otherwise, $S_{r_n}^{v_i}(\Delta t_x) = 0$

In this scenario, we focus on the communication scheduling problem on each RSU to optimize the network. The objective is to find a proper combination $S_{r_n}^{v_i}(\Delta t_x)$ for each RSU and vehicle for each time period that maximizes the service time. The problem is formulated as follows.

$$\text{Maximize throughput} = \frac{\sum_{v_i \in V} \sum_{\Delta t_x \in T} \sum_{r_n \in R} S_{r_n}^{v_i}(\Delta t_x)}{\text{total communication}}, \quad (1)$$

$$\text{Maximize service rate} = \frac{|\text{serviced vehicles}|}{|V|}, \quad (2)$$

subject to

$$\text{Service vehicles} = \left\{ v \mid v \in V, S_{r_n}^{v}(\Delta t_x) = 1, \exists \Delta t_x \in T, r_n \in R \right\}, \quad (3)$$

$$\sum_{\Delta t_x \in T} \sum_{v_i \in V} S_{r_n}^{v_i}(\Delta t_x) \le \min\left(B_{r_n}^{\Delta t_x}, \max_{r_i}\right), \quad \text{for } \forall r_n \in R, \quad (4)$$

$$r_n^{\Delta t_x}(v_i) \in L_i, \quad \forall S_{r_n}^{v_i}(\Delta t_x) = 1. \quad (5)$$

Formula (1) means that given a proper service scheduling set for each RSU and vehicle, with the limited energy in RSUs, maximize the successful communication rate between vehicles and RSUs. Formula (2) is used to maximize the ratio between the number of vehicles that can communicate with RSUs and the total number of vehicles. Constraint
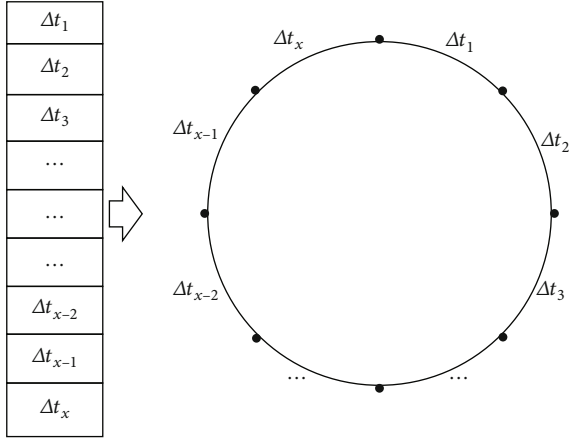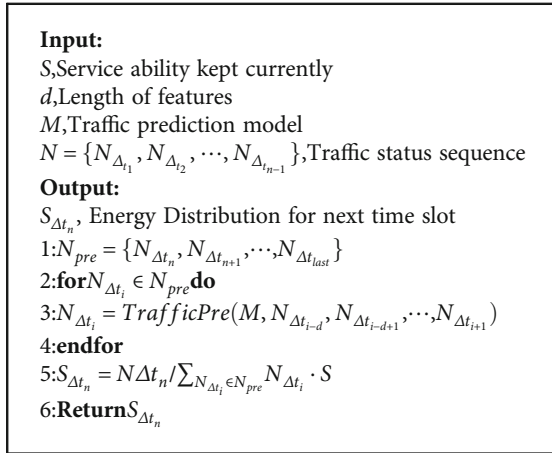
FIGURE 4: Method of sliding window.

Input:
S,Service ability kept currently
d,Length of features
M,Traffic prediction model
$N = \{N_{\Delta_{t_1}}, N_{\Delta_{t_2}}, \cdots, N_{\Delta_{t_{n-1}}}\}$,Traffic status sequence
Output:
$S_{\Delta t_n}$, Energy Distribution for next time slot
1:$N_{pre} = \{N_{\Delta t_n}, N_{\Delta t_{n+1}}, \cdots, N_{\Delta t_{last}}\}$
2:**for**$N_{\Delta t_i} \in N_{pre}$**do**
3:$N_{\Delta t_i} = TrafficPre(M, N_{\Delta t_{i-d}}, N_{\Delta t_{i-d+1}}, \cdots, N_{\Delta t_{i+1}})$
4:**endfor**
5:$S_{\Delta t_n} = N\Delta t_n / \sum_{N_{\Delta t_i} \in N_{pre}} N_{\Delta t_i} \cdot S$
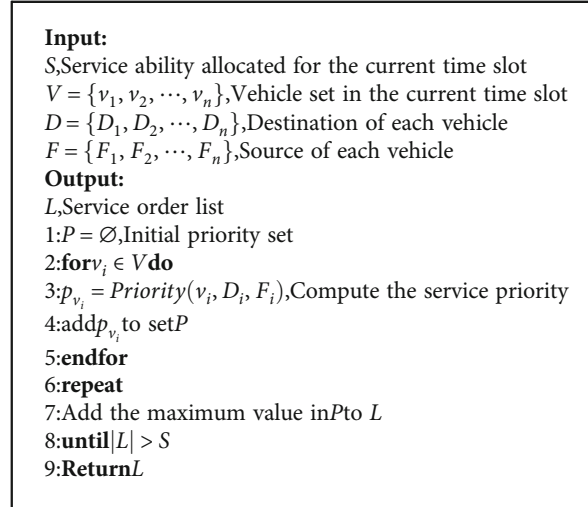6:**Return**$S_{\Delta t_n}$

ALGORITHM 1: Service Ability Distribution Algorithm.

(3) ensures that for each RSU, the energy cost for its communication with vehicles should be no more than the energy it is charged. Constraint (4) ensures if an RSU can communicate with a vehicle; the vehicle should be in the RSU's coverage range.

3.3. Main Idea. In order to maximize the service number of vehicles and the throughput, all the vehicles in the network should be connected to an RSU at any time, and then, the service number is maximum intuitively. Thus, we should figure out the number of vehicles in the network, based on which the total energy required to satisfy the communication between all vehicles and RSUs can be calculated.

As the energy cost of each RSU is relative to the number of vehicles covered by each RSU, the first step is to predict the number of vehicles passed by each RSU. Based on this, the required energy of each RSU can be obtained. However, the amounts of energy in each RSU are not the same, which are determined by surrounding environments. When the energy of an RSU cannot afford the energy cost on the vehicles passed by, a scheduling algorithm should be designed. A reasonable communication scheduling algorithm should be designed to save energy or reduce energy waste, especially when the energy of an RSU cannot afford the energy cost

Input:
S,Service ability allocated for the current time slot
$V = \{v_1, v_2, \cdots, v_n\}$,Vehicle set in the current time slot
$D = \{D_1, D_2, \cdots, D_n\}$,Destination of each vehicle
$F = \{F_1, F_2, \cdots, F_n\}$,Source of each vehicle
Output:
L,Service order list
1:$P = \varnothing$,Initial priority set
2:**for**$v_i \in V$**do**
3:$p_{v_i} = Priority(v_i, D_i, F_i)$,Compute the service priority
4:add$p_{v_i}$to set$P$
5:**endfor**
6:**repeat**
7:Add the maximum value in$P$to L
8:**until**$|L| > S$
9:**Return**L

ALGORITHM 2: Priority-based RSU access algorithm.

on communications. The main idea of the algorithm is to tune energy cost over all time periods and then to select more vehicles to access.

4. Deep Learning-Based Communication Scheduling Algorithm

4.1. Deep Learning-Based Traffic Prediction Mechanism. As Section 3 shows, the energy on each RSU is limited and each time of communication between an RSU and a vehicle costs energy. In this subsection, our concern is on how to predict the future traffic according to the current traffic status for each RSU. Here, we use the LSTM model to predict future traffic. Compared with the RNN model, LSTM can tackle a longer sequence better to avoid the affection of short memory. In order to make the traffic data fit the LSTM model, the traffic data should be preprocessed first. However, due to the sparsity of the training data of each RSU and the similarity of traffic-evolving rules on many RSUs, we cluster RSUs to several classes using the clustering method. At last, we use the LSTM model to train the traffic-evolving data in different classes to get a prediction model for each cluster of RSUs. Next, we will introduce the deep learning-based traffic prediction mechanism in the following parts.

4.1.1. Data Preprocessing Method. Usually, the traffic flow changes with the time and reveals similar evolving rules day by day. Thus, we focus on every day as an independent object to study the rules of traffic flow.

First, a day can be divided into many equal time slots Δt, such as an hour or a quarter.

For each RSU, the amount of traffic flow in each time slot forms a value sequence for a day, shown as Figure 3, where $N_{\Delta t_x}$ means the total number of vehicles at time slot Δt_x covered by an RSU. However, there may be noise in the data for some days, which is not the common phenomenon. For example, a traffic accident may cause a fast rising of the traffic flow under an RSU, which happens very
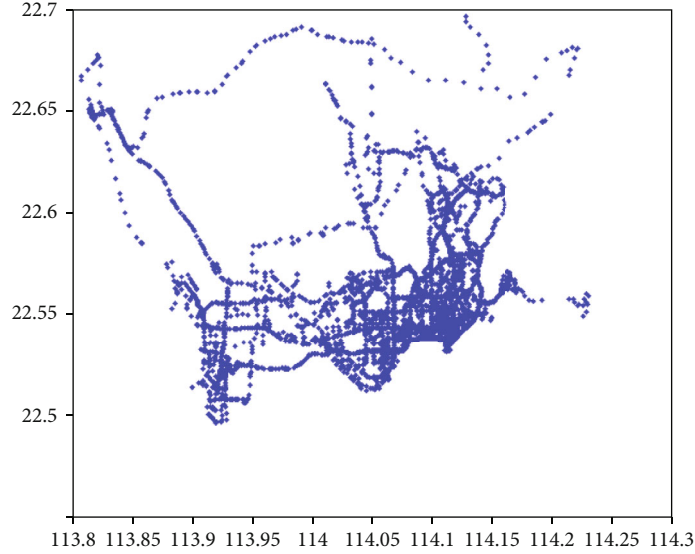
FIGURE 5: Trajectory snapshot.

TABLE 1: Simulation scenario description.

Area size	$50\,km \times 50\,km$
Duration	8 days
Number of RSUs	1536
Coverage of RSUs	$1\,km \times 1\,km$
Number of trajectories	975391

occasionally. Therefore, we get the average traffic of the same time slot on each day, shown as the bottom of Figure 3, which can reveal the original traffic status for each slot.

4.1.2. RSU Clustering Method. There are three advantages for RSU clustering. First, the amount of RSUs in a city is usually linearly related to the size of the city and there may be thousands of RSUs in a city. If we train a model for each RSU, it will cost too much computing resource, which is also not necessary. In addition, it is found that traffic flows on many RSUs reveal some similar evolving patterns, which can be tackled together. At last, it can solve the data sparsity problem as there may not be enough data to be trained for some RSUs in practice. When some RSUs are clustered into one class, the data on this class of RSUs can be shared together, so that the training set for this cluster of RSUs is extended.

After data preprocessing, the evolving rules for a cluster of RSUs are denoted as a sequence $(\bar{N}_{\Delta t_1}, \bar{N}_{\Delta t_2}, \bar{N}_{\Delta t_3}, \cdots, \bar{N}_{\Delta t_x})$. Here, this sequence is used as the feature vector for model training. As mentioned before, the time slot Δt can be chosen as a quarter, an hour, or longer. In this paper, it is needed to catch the evolving rule from a microscopic time perspective but the length of the sequence cannot be large considering the computing consumption. Besides, there should be obvious up or down trending for the sequence value; rather than that, most of the values in the sequence are closed to zero. Thus, Δt is set to a quarter, so that the sequence length, namely, the number of dimensions of the feature vector, is $60/15 \times 24 = 96$.

So far, each RSU has a special feature vector, which will reveal individual characteristic. Next, all these feature vectors are used as the input source to cluster all the RSUs. But, it is hard to get a good result if we directly cluster such high-dimensional feature vectors with general clustering methods [24]. In order to solve the problem of high-dimensional data clustering, many approaches have been proposed. Clustering with a subset and fuzzy clustering are typical representatives of these methods [25, 26]. Moreover, the deep learning-based clustering algorithm is also used to cluster high-dimension data, in which a deep learning strategy is used to learn low-dimensional representation of high-dimensional data. Then, the ordinary clustering algorithm can be used after the low-dimensional representation is obtained. Although a better clustering effect can be obtained using these clustering methods, they cluster high-dimensional data in an abstract way and there are still challenges in data visualization and clustering rationality.

T-distributed stochastic neighbour embedding (t-SNE) is a new dimension reduction and visualization technique for high-dimension data [27], which is used to reduce the dimensionality of each RSU's traffic flow so that all RSUs can be clustered well. T-SNE is developed from SNE that has a "crowding problem." Different from SNE, the Euclidean distance is converted to joint probability to express the similarity between data points in t-SNE (Euclidean distance is converted to conditional probability in SNE). As shown in equations (6) and (7), p_{ij} and q_{ij} are the joint probability of the original high-dimensional data and low-dimensional data after dimensionality reduction, respectively.

$$p_{ij} = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma^2\right)}{\sum_{k\neq l}\exp\left(-\|x_k - x_l\|^2/2\sigma^2\right)}, \qquad (6)$$
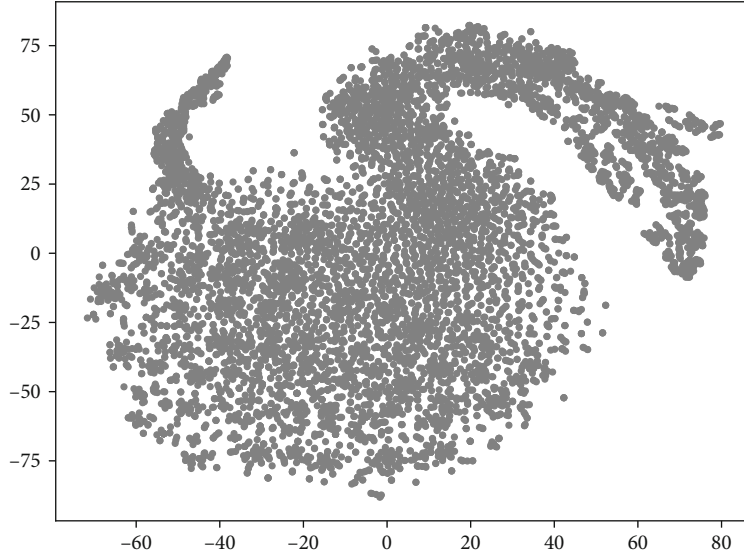
FIGURE 6: The visualization of each RSU feature vector.

$$q_{ij} = \frac{\exp\left(-\left\|y_i - y_j\right\|^2\right)}{\sum_{k \neq l}\exp\left(-\left\|y_k - y_l\right\|^2\right)}. \tag{7}$$

The difference in the joint probability distribution of the original data and the data after dimensionality reduction will be very small, even $p_{ij} = q_{ij}$. Based on this idea, Kullback-Leibler (KL) divergence is used to measure the difference between two distributions, shown in equation (8), where P is the joint probability distribution in the high-dimensional space (the original data), Q the joint probability distribution in the low-dimensional space (the data after dimensionality reduction), and C the cost or the sum of KL divergence difference value between two distributions. The smaller C is, the better the dimensionality reduction effect is.

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{8}$$

As shown in equation (9), for the purpose of minimizing C or the sum of the KL divergence, the gradient descent method is used so that a better dimensionality reduction effect can be obtained. Finally, the desired low-dimensional data will be obtained by continuous iterative solutions.

$$\frac{\delta C}{\delta y_i} = 4\sum_j \left(p_{ij} - q_{ij}\right)\left(y_i - y_j\right). \tag{9}$$

By using t-SNE, the low-dimensional representation of the traffic flow in each RSU can be obtained. Then, the K-means algorithm can be used for clustering RSUs, in order to determine a proper K value. The elbow method is used to fix the best number of clusters K.

Its main idea is to choose the optimal K according to the varying trend of the sum of squared errors (SSE) in a cluster. SSE is defined as equation (10).

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2. \tag{10}$$

Here, C_i is the ith cluster, p is the sample vector of C_i, and m_i is the center of cluster C_i. When K is smaller than the actual cluster number, the value of SSE descends fast with K increasing. However, when a proper K is found, the value of SSE tends to be stable with K growing. Therefore, the inflection point is the best K value.

4.1.3. Deep Learning-Based Traffic Prediction. Once all the RSUs are clustered, the future traffic flow should be predicted based on the history data of RSUs. As real-time prediction is hard to achieve, we use the LSTM model to train the dataset offline.

First in the learning stage, we use the vectors from each RSU class as the training dataset individually. However, if the 96-dimension features of the vector are used as the training set, it will cost too much computing resource. Intuitively, the current traffic conditions are usually affected by the last few hours of traffic conditions. Thus, we use a method called the sliding window with size n to cut off the feature vector. So, in the learning stage, the consecutive n elements in a vector compose the feature sequence and the following element is the tag.

In the prediction stage, the consecutive n elements in a vector are used to predict the following element, shown as the left part in Figure 4. Another problem is how to predict the first n elements. As the traffic flow is changing periodically, the vector can be transformed to a clock-like vector, shown as the right part in Figure 4. For example, to predict the first element, we can use the last n elements of the vector as input.
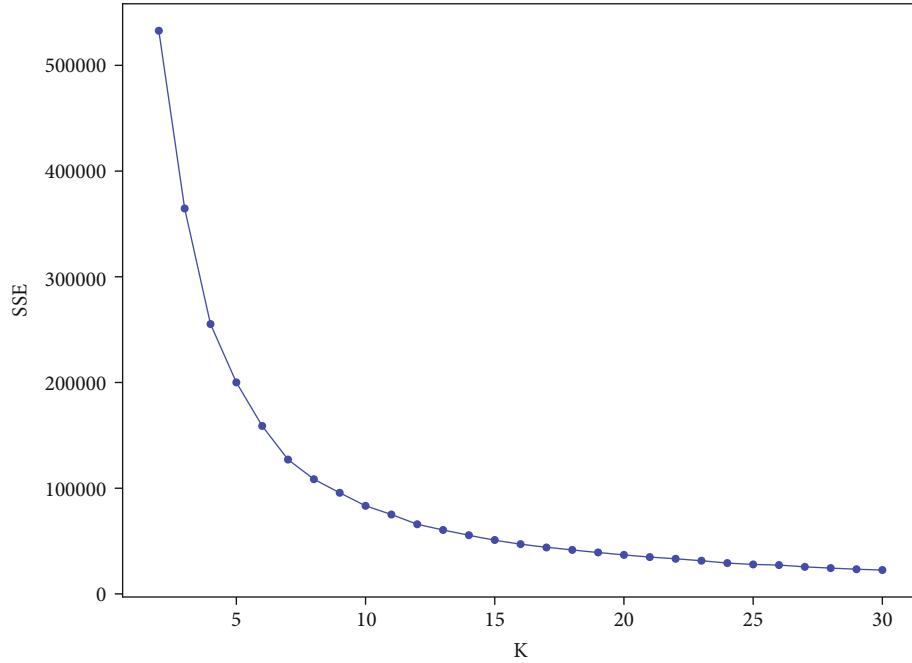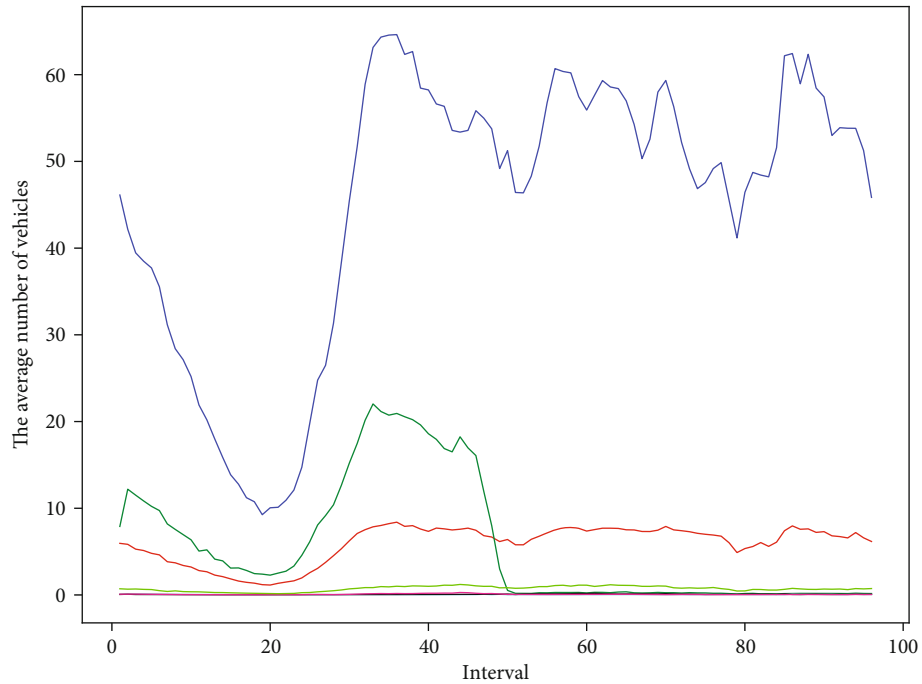
(a) SSE values vs different K values

(b) Clustering result of $K = 7$

Figure 7: Result of clusters.

4.2. Scheduling Algorithm. There are two parts in the RSU communication scheduling algorithm, namely, service ability distribution algorithm and the priority-based RSU access algorithm. When a new period starts, such as end of daytime, the energy of each RSU has already been charged during daytime. From then on, the energy is continually discharging until the next period. We assume that the energy charged during daytime will be used in the next period, rather than in the current period. Therefore, the energy

should support the RSU to work for the entire period. Without a proper scheduling, the energy might be exhausted quickly, so that the vehicles traveling during the end of the period may not be served no matter its priority.

Therefore, we propose the service ability distribution algorithm to schedule the energy distribution for each RSU. The main idea of the Algorithm 1 is sketched below. First, when the energy harvested during daytime is determined, the total service ability in this period of the RSU will
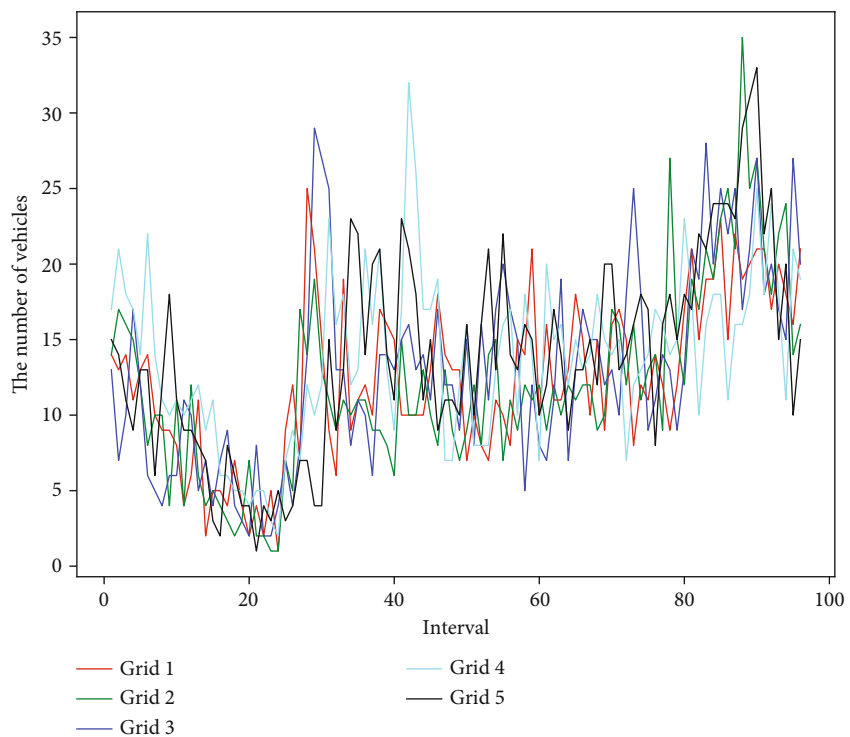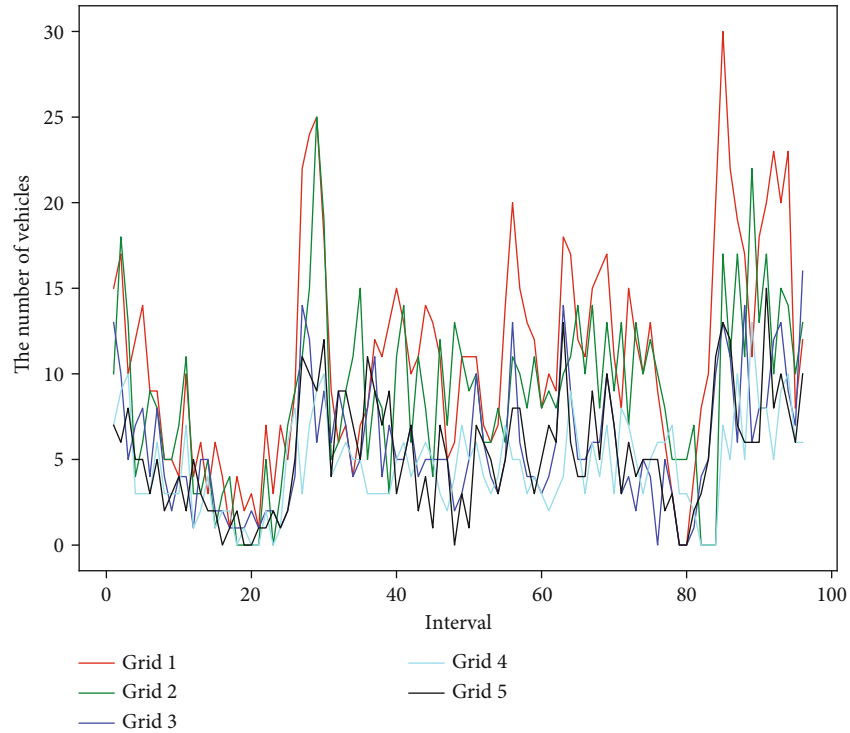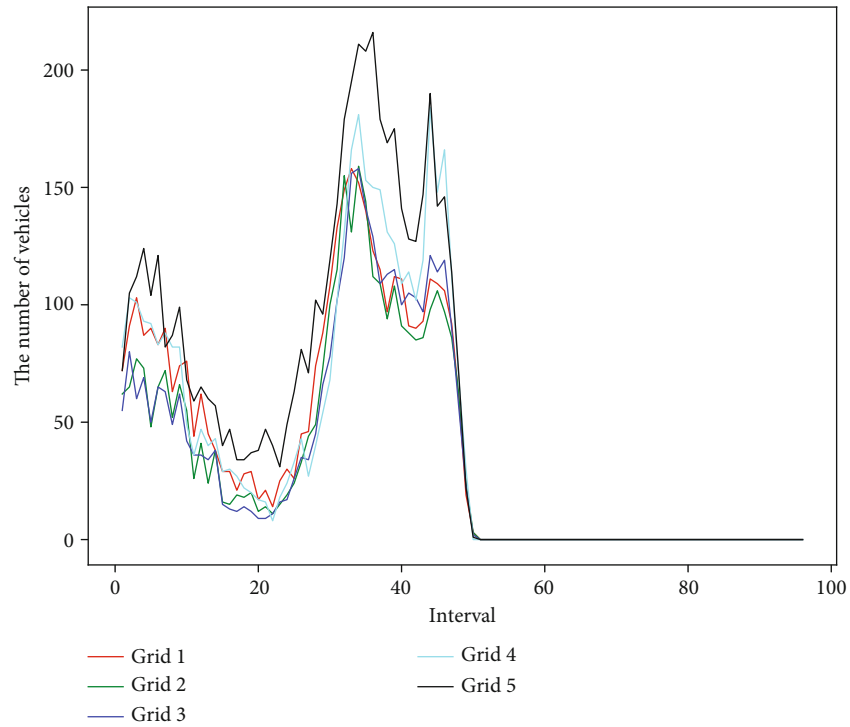
(a) Cluster 1
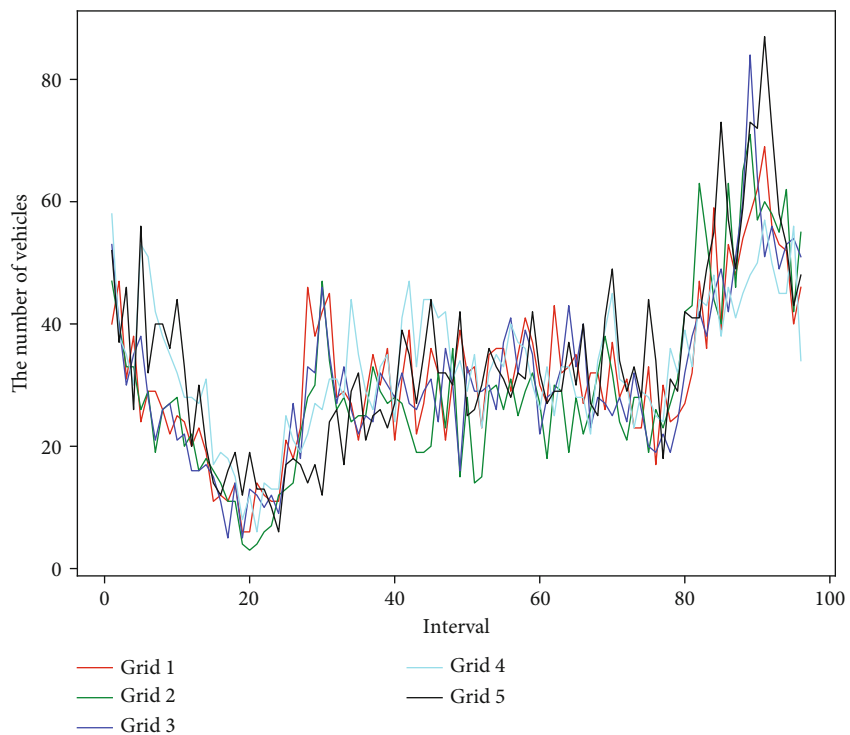
(b) Cluster 2

FIGURE 8: Continued.

(c) Cluster 3

(d) Cluster 4

FIGURE 8: Examples of clusters.

be determined, too. Next, we will assign the total service ability to each time slot according to the predicted number of vehicles in each time slot. In order to predict the number of vehicles, we use the above traffic prediction algorithm to get the traffic flow from the following slot to the ending slot of the period. Then, the service ability for each time slot is assigned according to the total service ability multiplied by the ratio of the current slot traffic flow to the total traffic flow

Table 2: Loss for each cluster under various feature vector lengths.

Class	Length		
	5	10	15
1	3615	3608	3594
2	836	835	834
3	525133	439265	436616
4	27915	27420	27247
5	1282	1282	1282
6	857	857	856
7	63173788	60481962	59170782

Table 3: The average MSE loss of different models ($\times 10^4$).

L	N				
	16	32	64	128	256
1	6.856	5.958	4.453	4.159	4.149
2	6.973	5.761	4.436	4.438	4.146
3	8.71	6.409	5.121	4.779	5.132
4	9.945	7.108	5.433	6.115	5.375
5	12.049	8.158	6.573	7.048	6.394

Table 4: The prediction error under different batch sizes.

B	C				
	16	32	64	100	200
1	4.429	4.429	**4.429**	4.43	4.43
2	4.267	4.132	4.018	3.921	**3.838**
3	**7.124**	9.693	11.752	13.366	14.736
4	14.454	14.235	14.051	13.891	**13.767**
5	12.779	11.944	11.227	10.607	**10.063**
6	9.976	9.892	9.809	9.729	**9.65**
7	**12.739**	15.575	18.252	20.728	23.129

of the period. Once the service ability on each time slot for an RSU is assigned, it is necessary to determine the service vehicles and the service order.

Here, we propose a conception of service priority, defined as formula (11) (where we assume that the destination of each vehicle is known).

$$\text{Service priority} = \frac{1 + \text{original distance}}{1 + \text{destination distance}}. \quad (11)$$

It means that the longer the time that the vehicle has travelled, the higher the priority it has, because it may have more data to submit. Besides, if a vehicle has a long way to its destination, it may have more chances to communicate with RSUs, so that the priority is small. Here, the distance can be Euclidean distance or Manhattan distance. Based on this algorithm, when several vehicles are in the coverage of an RSU. The RSU calculates the priorities of all vehicles and provides service in the descending order of the priority, shown as Algorithm 2.

5. Simulations

5.1. Simulation Scenario. We evaluate our mechanism in a scenario with real vehicle trajectories, which come from the taxis in Shenzhen city, and a snapshot of the taxi location at a time point is shown in Figure 5. In this scenario, we assume that all the RSUs are uniformly distributed with the coverage range of $1 \, \text{km} \times 1 \, \text{km}$. Then, it can be considered that any area in Shenzhen is covered by an RSU. The parameters of the scenario are shown in Table 1.

The simulation is composed of three parts, namely, RSU clustering, traffic flow prediction, and RSU communication scheduling process. First, in the RSU clustering process, a day is divided to 96 time slots with each 15 minutes, so the clustering is based on the dataset of 96-dimensional vectors as we introduced in IV-A2. Then, the best K value is chosen by comparing varying SSE values, and in the following part, this value is used to deploy the simulation. Third, the trajectory dataset lasting for seven days is selected as the training dataset and the dataset for the other day is the testing set. Fourth, we choose different lengths of feature vectors to train the models in order to compare the accuracy of prediction.

At last, in the RSU communication scheduling process, we compare the number of successfully accessed vehicles between our scheduling algorithm and other scheduling mechanisms when the energy is not sufficient for all the vehicles' communications.

5.2. Simulation Result

5.2.1. RSU Clustering. As described in IV-A1, the dimensionality of original traffic flow data needs to be reduced before RSUs can be clustered with t-SNE. For the convenience of data visualization, the 96D data is reduced to 2D in our simulation and the visualization of the original traffic flow is shown in Figure 6. Based on the dimensionality reduction result with t-SNE, the K-means algorithm can be performed. As shown in Figure 7(a), the SSE value decreases with K increasing. When K is 7, the SSE starts to descend slowly. Although the SSE value still falls down, the falling gradient becomes smaller. Therefore, it is believed that the best K value is 7. After the clustering, all the RSUs are divided to 7 clusters.

Then, in Figure 7(b), the clustering result is described with the average value of all the vectors in each cluster. It is found that 3 clusters are shown clearly and each of them has a special evolving rule. The other clusters are not very clear as most of elements in the vector are very small, even almost 0. In Figure 8, we take four clusters as examples, each of which contains several 96-dimensional vectors of RSUs. It is found that the vectors in a cluster show similar evolving trending, which can prove the correctness of our cluster mechanism.

5.2.2. Deep Learning-Based Traffic Prediction. In the deep learning-based prediction algorithm, the length of feature vectors is set to 5, 10, and 15. Table 2 shows the losses varying with various feature lengths for each cluster, in which the value is the Euclidean distance between the predicted vector and the original vector. When the length of feature is 5, the
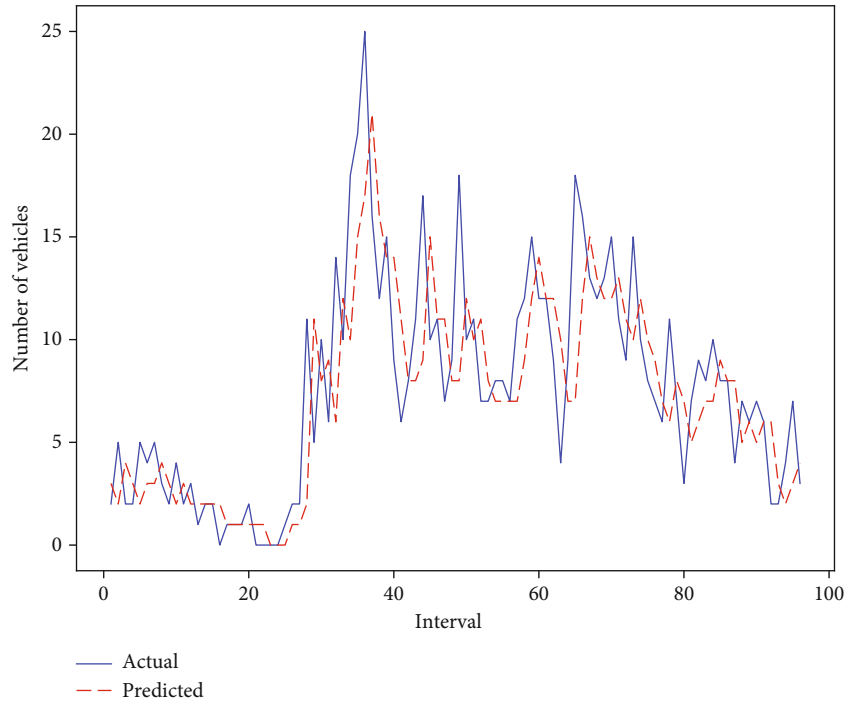
(a) Grid 1
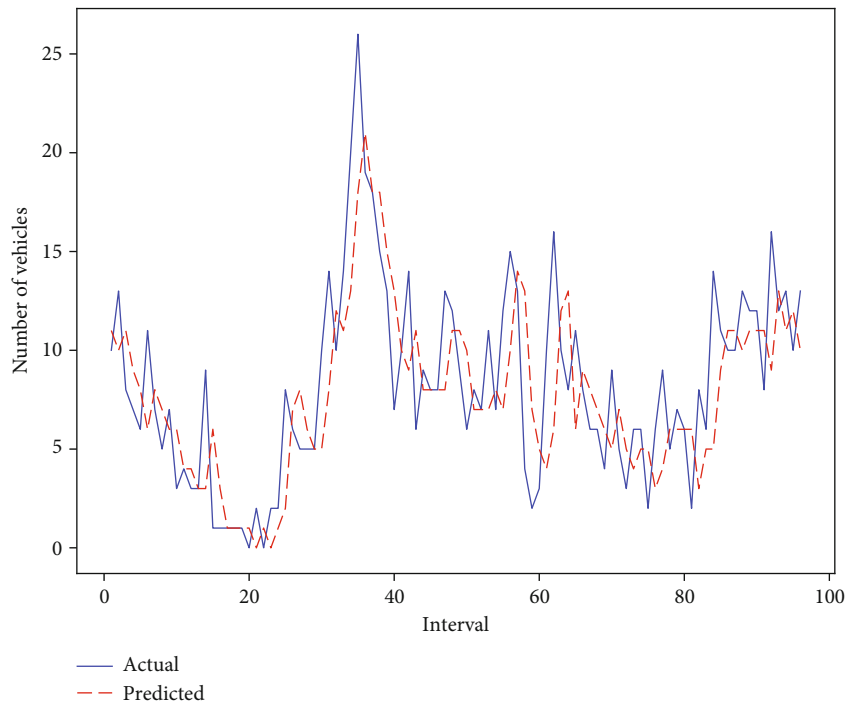
(b) Grid 2

FIGURE 9: Continued.
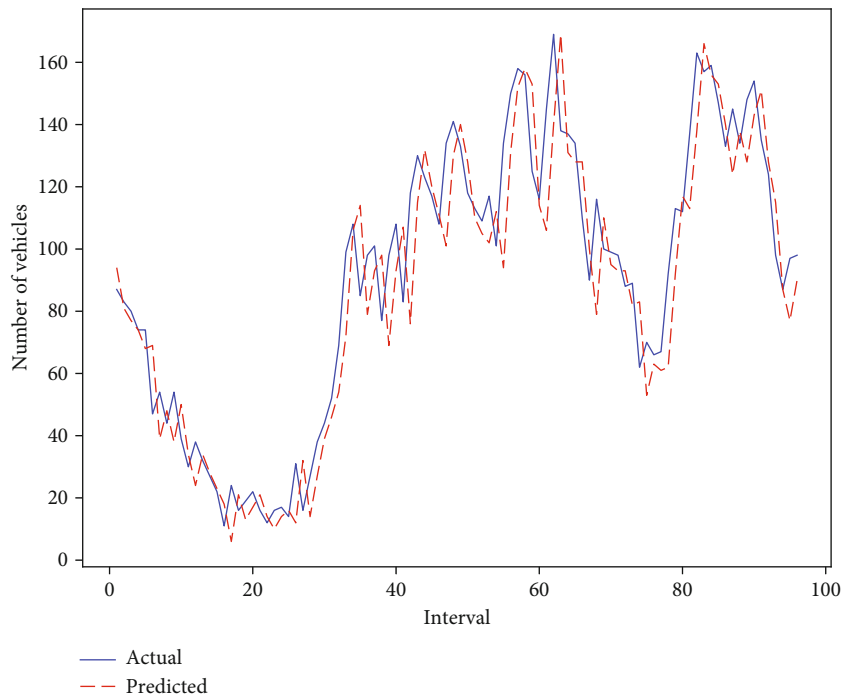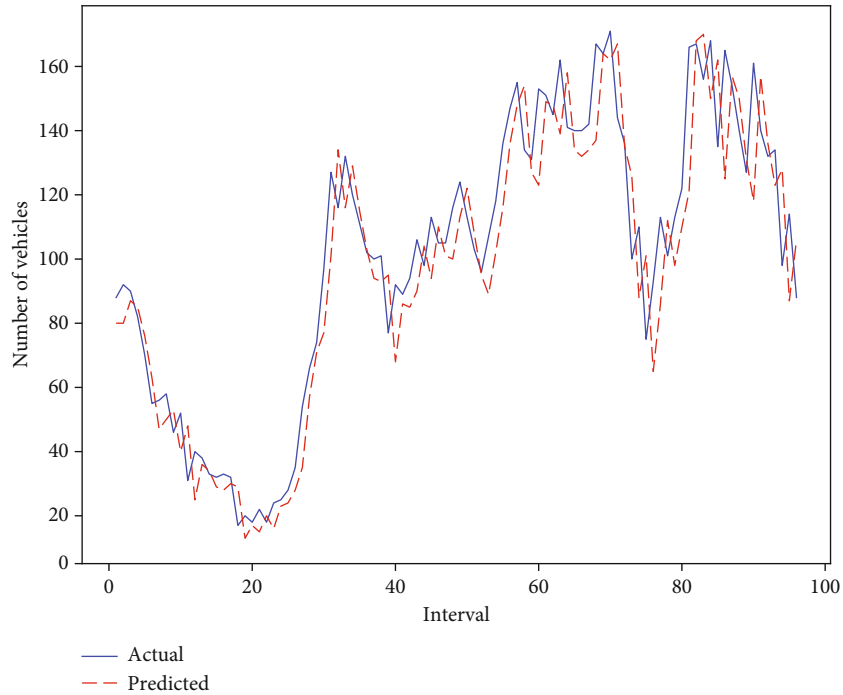
(c) Grid 3

(d) Grid 4

FIGURE 9: Comparison between predicted vectors and original vectors.

loss is the highest compared to the others. For the length of the feature vector being 10 and 15, the losses for each cluster are almost the same. However, considering the computing resource exhausted, we choose the feature length as 10. In order to choose the best prediction model for RSUs in each class, the hyperparameters of the LSTM network are tuned by changing the number of layers and neurons of each layer at first. The activation function and loss function are set as "tanh" and "mse," respectively. Dropout and early stop are used to prevent overfitting, where the value of "dropout" is 0.2. The average MSEs of different models are shown in Table 3, in which "N" represents the number of neurons in each layer and "L" means the number of layers. As shown in Table 3, the model has less MSE loss when the layer is 1

TABLE 5: The energy consumption with different scheduling algorithms under two scenarios.

	FCFS	Random	Average	Priority
50%	429352	415540	415540	**414513**
75%	644000	622174	622174	**601213**

TABLE 6: The number of served vehicles with different scheduling algorithms under two scenarios.

	FCFS	Random	Average	Priority
50%	6355	6349	6193	**6396**
75%	6407	6371	6351	**6412**

and neuron is 256 in most cases. Therefore, the 1-layer LSTM model with 256 neurons in each layer is chosen to train the final prediction model in the flowing simulations. Moreover, the batch size also affects the prediction accuracy of training models. Thus, the models with different batch sizes are trained so that the best prediction model can be found in each class. The average prediction error with different batch sizes of each class is shown in Table 4, where "B" and "C" denote "batch size" and "class," respectively. The smallest error is shown in bold, and the corresponding model is selected as the final prediction model. Next, we show the comparison between the prediction vector and the original vector in four grids as examples. In Figure 9, four RSUs are randomly selected from four clusters, which reveals that each prediction model fits well for the corresponding cluster.

5.3. *Scheduling Algorithm.* In this part, we simulate the scenario that the RSUs do not have enough power to work through the entire period. Table 5 shows the energy consumption when the service ability is half and three quarters of requirement on different scheduling algorithms, in which "FCFS" denotes "first come first service," "random" means an RSU randomly chooses some vehicles to serve, "average" denotes the service ability of each interval is equal for each RSU, and "priority" represents that an RSU serves the vehicles according to the proposed scheduling algorithm in this paper. It can be seen that RSUs consume minimal energy in the two scenarios with the proposed scheduling algorithm.

Besides, the number of served vehicles under two scenarios are shown in Table 6. It is obvious that RSUs serve the maximized number of vehicles using minimized energy consumption with the proposed algorithm. Therefore, the performance of our proposed algorithm is proved.

6. Discussion and Conclusion

RSUs are playing a more and more important role in the IoVs. In terms of energy and deployment cost, renewable energy-powered RSUs will be used widely in the future. In this paper, we study the energy scheduling problem on renewable energy-powered RSUs. We propose a deep learning-based communication scheduling algorithm for RSUs in the IoVs, which is composed of RSU clustering, deep learning-based traffic prediction, service ability distribution algorithm, and priority based RSU accessing algorithm. At last, we conduct extensive simulation and the simulation results indicate that our algorithm can achieve a higher fairness rate while keeping a proper performance than the no scheduling algorithm.

Data Availability

The original data is the GPS data of taxis in Shenzhen, China, which were continuously sampled by the GPS devices during time period 2011/04/18–2011/04/26. Considering the large amount of original data, the processed data is submitted at the website https://http://github.com/pultoW/trajectory. If there is any question, you can contact the author via ljt_it@163.com.

Disclosure

A part of this manuscript has been related in the Conference "2020 92nd IEEE Vehicular Technology Conference, Oct. 4, 2020" with technique parts extended, and the simulation has been done with extended algorithms and mechanisms proposed in this paper [28].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] C. H. Ou, B. Y. Wu, and L. Cai, "GPS-free vehicular localization system using roadside units with directional antennas," *Journal of Communications and Networks*, vol. 21, no. 1, pp. 12–24, 2019.

[2] C. Wen and J. Zheng, "An RSU on/off scheduling mechanism for energy efficiency in sparse vehicular networks," in *2015 international conference on Wireless Communications & Signal Processing (WCSP).*, pp. 1–5, Nanjing, China, 2015.

[3] E. Benalia, S. Bitam, and A. Mellouk, "Data dissemination for Internet of vehicle based on 5G communications: a survey," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, article e3881, 2020.

[4] C. Wang, X. Cheng, J. Li, Y. He, and K. Xiao, "A survey: applications of blockchain in the Internet of vehicles," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, 16 pages, 2021.

[5] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 1–590, 2016.

[6] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[7] C. Wang, J. Li, Y. He, K. Xiao, and H. Zhang, "Destination prediction-based scheduling algorithms for message delivery in iovs," *IEEE Access*, vol. 8, pp. 14965–14976, 2020.

[8] M. M. Najm, M. Patra, and T. Venkatesh, "Cost-and-delay aware dynamic resource allocation in federated vehicular clouds," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6159–6171, 2021.

[9] W. S. Atoui, M. A. Salahuddin, W. Ajib, and M. Boukadoum, "Scheduling energy harvesting roadside units in vehicular ad hoc networks," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Montreal, QC, Canada, 2016.

[10] Q. Ibrahim, "Design, implementation and optimisation of an energy harvesting system for vehicular ad hoc networks' road side units," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 298–307, 2014.

[11] S. Mostofi, A. Hammad, T. D. Todd, and G. Karakostas, "On/off sleep scheduling in energy efficient vehicular roadside infrastructure," in *2013 IEEE International Conference on Communications (ICC)*, pp. 6266–6271, Budapest, Hungary, 2013.

[12] G. A. Audu, S. Bhattacharya, A. Muhtar, B. Qazi, and J. M. H. Elmirghani, "Reliability and quality of service of an off-grid wind powered roadside unit in a motorway vehicular environment," *Vehicular Communications*, vol. 9, pp. 176–187, 2017.

[13] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, "Mobile edge computing-enabled Internet of vehicles: toward energy-efficient scheduling," *IEEE Network*, vol. 33, no. 5, pp. 198–205, 2019.

[14] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[15] A. A. Hammad, T. D. Todd, G. Karakostas, and D. Zhao, "Downlink traffic scheduling in green vehicular roadside infrastructure," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 1289–1302, 2013.

[16] B. L. Nguyen, D. T. Ngo, M. N. Dao, Q. T. Duong, and M. Okada, "A joint scheduling and power control scheme for hybrid I2V/V2V networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15668–15681, 2020.

[17] W. S. Atoui, W. Ajib, and M. Boukadoum, "Offline and online scheduling algorithms for energy harvesting RSUs in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6370–6382, 2018.

[18] A. Khezrian, T. D. Todd, G. Karakostas, and M. Azimifar, "Energy-efficient scheduling in green vehicular infrastructure with multiple roadside units," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 1942–1957, 2015.

[19] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4592–4601, 2017.

[20] R. Atallah, C. Assi, and M. Khabbaz, "Deep reinforcement learning-based scheduling for roadside communication networks," in *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 1–8, Paris, France, 2017.

[21] Y. J. Ku, P. H. Chiang, and S. Dey, "Real-time QoS optimization for vehicular edge computing with off-grid roadside units," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11975–11991, 2020.

[22] L. Dai, T. Chen, Y. Zhai, and G. Wang, "Energy-efficient distributed packet scheduling optimization strategy in cooperative vehicle infrastructure systems," *Wireless Communications and Mobile Computing*, vol. 2021, 11 pages, 2021.

[23] X. Wang, Z. Ning, X. Hu et al., "Future communications and energy management in the Internet of vehicles: toward intelligent energy-harvesting," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 87–93, 2019.

[24] Y. Chen, S. Tang, N. Bouguila, C. Wang, J. du, and H. L. Li, "A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data," *Pattern Recognition*, vol. 83, pp. 375–387, 2018.

[25] R. Elankavi, R. Kalaiprasath, and D. R. Udayakumar, "A fast clustering algorithm for high-dimensional data," *International Journal Of Civil Engineering And Technology (Ijciet)*, vol. 8, no. 5, pp. 1220–1227, 2017.

[26] J. Wang, N. Mao, X. Chen, J. Ni, C. Wang, and Y. Shi, "Multiple histograms based reversible data hiding by using FCM clustering," *Signal Processing*, vol. 159, pp. 193–203, 2019.

[27] W. Li, J. E. Cerise, Y. Yang, and H. Han, "Application of t-SNE to human genetic data," *Journal of Bioinformatics and Computational Biology*, vol. 15, no. 4, article 1750017, 2017.

[28] C. Wang, J. Li, X. Cheng, Y. He, L. Sun, and K. Xiao, "LSTM-based communication scheduling mechanism for energy harvesting RSUs in IoVs," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 1–5, Victoria, BC, Canada, 2020.