

Trajectory-prediction based relay scheme for time-sensitive data communication in VANETs

Zilong Jin^{1,2*}, Yuxin Xu¹, Xiaorui Zhang¹, Jin Wang³, and Lejun Zhang⁴

¹ School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

[e-mail: zljn@nuist.edu.cn, 13375228989@163.com, zxr365@126.com]

² Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing 210044, China

³ School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410000, China

[e-mail: jinwang@csust.edu.cn]

⁴ College of Information Engineering, Yangzhou University, Yangzhou 225127, China

[e-mail: zhanglejun@yzu.edu.cn]

*Corresponding author: Zilong Jin

Received February 11, 2020; revised April 20, 2020; revised June 16, 2020; revised July 22, 2020; accepted August 3, 2020; published August 31, 2020

Abstract

In the Vehicular Ad-hoc Network (VANET), the data transmission of time-sensitive applications requires low latency, such as accident warnings, driving guidance, etc. However, frequent changes of topology in VANET will result in data transmission failures. In order to improve the efficiency of VANETs data transmission and increase the timeliness of data, this paper proposes a relay scheme based on Recurrent Neural Network (RNN) trajectory prediction, which can be used to select the optimal relay vehicle to transmit data. The proposed scheme learns vehicle trajectory in a distributed manner and calculates the predicted trajectory, and then the optimal vehicle can be selected to complete the data transmission, which ensures the timeliness of the data. Finally, we carry out a set of simulations to demonstrate the performance of the algorithm. Simulation results show that the proposed scheme enhances the timeliness of the data and the accuracy of the predicted driving trajectory.

Keywords: Vehicular Ad-hoc Network, Optimal Relay Vehicle Selection, Recurrent Neural Network, Time-sensitive, Trajectory Prediction

A preliminary version of this paper was presented at ICONI 2019, and was selected as an outstanding paper. This work was supported by the National Natural Science Foundation of China (Grant No. 61602252, 61802197, 61972207), the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20160967), and the Project through the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institution.

1. Introduction

The Vehicular Ad-hoc Network refers to an open mobile Ad-hoc network composed of vehicles, vehicles and infrastructures, vehicles and pedestrians communicating with each other in a traffic environment. The VANET combines wireless communication between vehicles and positioning technology into the vehicle's sensor assembly to provide over-the-horizon advance sensing, which benefits the safety of the driver and the surrounding environment, increases driving comfort and reduces environmental impact. Therefore, VANET plays a significant role in connecting vehicles to share important driving data in autonomous driving, making applications such as accident warning, assisted driving, road traffic information inquiry, inter-vehicle communication and Internet access services can be provided to vehicles. Autonomous driving technology is an essential part of the future vehicle, which enables vehicles to understand their driving conditions and road environment and respond to emergencies in time. These advantages often require the ability to exchange urgent data in time and relay it to the data center provides computing and scheduling services so that we can get help from it. In the VANET, vehicles communicate in a cooperative manner to facilitate communication between vehicles and base stations [1]. But due to the high-speed mobility of vehicles, network topologies often change [2], and this will cause communication delays for many new service platforms aimed at improving the quality of VANET transmissions. Therefore, autonomous driving also faces severe challenges, such as being unable to respond in time to unexpected speed changes and sudden changes in road conditions. However, existing communication protocols in VANETs are difficult to cope with communication requirements so far. With the rapid development of autonomous driving technology, reducing communication delay becomes a hot issue in VANET. The communication delay is usually caused by the unstable driving condition of the vehicle. Therefore, predicting vehicle trajectory in advance can effectively reduce communication delay.

In VANET, all vehicles on a road follow traffic laws, thus the movement of a vehicle is predictable [3-5]. In addition, public transportation facilities, such as buses, metro, etc., all follow certain rules of driving. For example, the bus must arrive at a predetermined station at a certain time. Other commercial vehicles, such as taxis, buses, etc., pick up and drop off customers in a certain area. At the same time, the popularity of Global Positioning System(GPS) facilitates the collection of driving data. The movement pattern of these vehicles can be learned from these driving data, and it is good for choosing the appropriate relay vehicle to transmit data. At present, many researchers related to mobility prediction have been proposed. Some researchers presented predictive models, and others developed applications that predict vehicle mobility, which makes significant contributions to driving, safety, and vehicle communications [6-14]. In [6], the authors proposed an intuitive and effective regional transformation model to describe vehicle mobility between regions divided by urban intersections based on two urban vehicle trajectories. The model predicts the time a vehicle will arrive at a position, which helps to understand the traffic pressure in one area and provides a more time-saving route for other drivers. But the predicted limit value fluctuates too much and the accuracy fluctuates greatly. In [15-16], the work of mobility prediction in Mobile Ad-hoc Network(MANET) is discussed. Although more and more researchers focus on vehicle mobility prediction, few of them adopt machine learning to predict vehicle mobility. For the transmission of time-sensitive data in autonomous driving applications, there is a certain trade-off between computational overhead and communication efficiency. And in order to meet the two requirements at the same time, we can learn the driving history of the

vehicle which helps predict its driving trajectory and set an optimized communication strategy.

Based on the above considerations, a relay vehicle selection algorithm is proposed in this paper, in which the most suitable vehicle to relay data can be selected. Firstly, we read historical data of vehicles including direction, speed, position, etc., to learn the driving pattern based on RNN. Secondly, the future trajectory of the vehicles is predicted. We choose a vehicle whose future trajectory passes by the base station with the shortest time as the relay vehicle through the proposed relay vehicle selection algorithm. Finally, the selected vehicle will relay emergency data to the base station and complete communication in the shortest time. In order to prove the performance of the proposed algorithm, a set of simulations is performed. And the simulation results show that the predicted trajectory is basically consistent with the actual trajectory, and the time that the base station receives the data is also greatly reduced.

The remainder of this paper is organized as follows. The related work is discussed in Section 2. In Section 3, the network model and problem definition are described. The proposed relay vehicle selection algorithm is introduced in Section 4. In Section 5, the simulation results of the proposed scheme are shown. Section 6 is the conclusion.

2. Related Work

At present, the predictability of vehicle mobility is mainly divided into predicting its future position, trajectory and travelling time [5], as shown in Fig. 1. In [17], the authors proposed a vehicle trajectory prediction method that integrates environmental perception and user preference. They used the SSEM algorithm to deal with the user preference and employed the optimal variational gaussian mixture model to represent the complex environment, so that the prediction is realized. This paper focuses on the prediction of future trajectories. There are two main methods to solve this problem. The first method is to predict the future trajectory through time series. It uses machine learning methods to build the model, and then predicts future trajectories by training the existing driving data, as shown in 2.1 [8-10]. The second method is to predict the future trajectory of the vehicle by its previous position, direction, velocity and acceleration [11-14]. This method measures the change in vehicle state through physical motion equations, as detailed in 2.2.

Based on the predicted trajectory, suitable relay vehicles can be selected. Many people focus on relay selection studies. In [18], the authors designed a relay selection method based on distance, and developed an analytical model based on competitive delay and packet delivery rate (PDR). In [19-21], the authors used machine learning methods such as Markov chain as the basis for relay selection. And in [22-25], the authors proposed the algorithm for moving target recognition and detection based on the data in the RFID tags, which can better identify and classify the data.

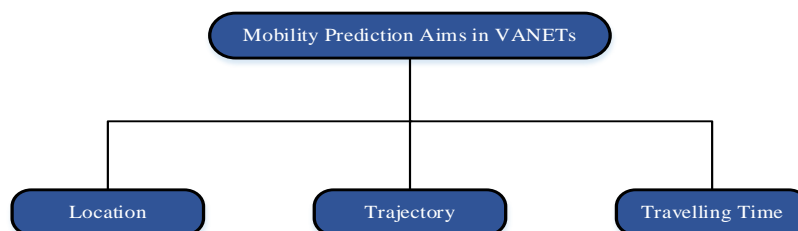


Fig. 1. Type of research for mobility prediction

2.1 Predicting Trajectory Through Machine Learning

Machine learning is to study how computers simulate or implement human learning behaviors to acquire new knowledge or skills, and reorganize existing knowledge structures, which continuously improves their performance. The neural network, a kind of machine learning method, is a mathematical model that simulates the behavioral characteristics of animal neural networks and performs distributed parallel information processing. In [15], a competitive neural network is used to extract the mobility pattern of the vehicle, and the extracted pattern is utilized to predict the future direction of the vehicle. This is a novel mobility prediction model using self-organizing maps (SOMs).

In order to reduce the impact of resource mobility on vehicle cloud performance, a novel solution based on Artificial Neural Network (ANN) mobility prediction model to predict future trajectory is proposed in [16]. The authors utilized the performance of the vehicle cloud to reduce the impact of sudden changes in vehicle position based on the ANN mobility prediction model. The model has three layers: the input layer, the hidden layer and the output layer. The method can change the output to a linear function and the NN to continuous linear regression. Acceptable performance can be obtained by adjusting NN hyper-parameters, such as learning rate, hidden layers, and the number of neurons in each layer. Finally, the cost of different schemes is compared to prove the performance of the proposed scheme. The formula for the percentage of cost is as follows:

$$OH = \frac{T - N \times T_j}{T} \times 100\% \quad (1)$$

where OH is total overhead percentage, T is total simulation duration, N is the number of jobs accomplished during simulation, T_j is job duration.

2.2 Predicting Trajectory by Physical Motion Equation

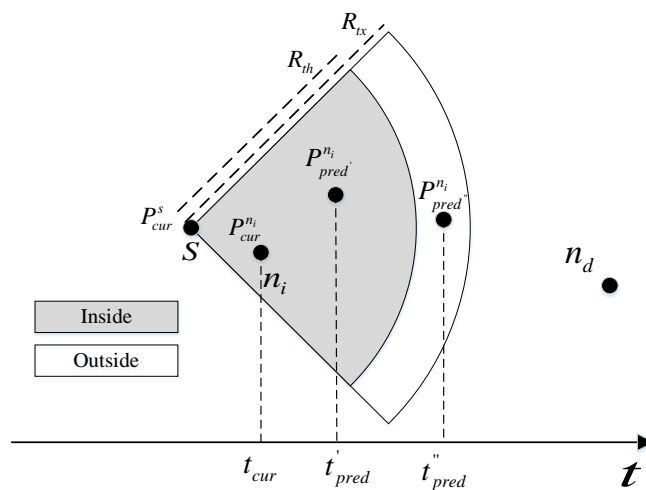


Fig. 2. Predicting trajectory by physical motion equation

Park *et al.* [26] proposed a method for the routing improvement problems, which used the mobility data contained in the beacon to select a reliable intermediate vehicle, as shown in Fig. 2. The mobility data contains the position $P_{cur}^{n_i}$, the velocity $V_{cur}^{n_i}$, the acceleration $a_{cur}^{n_i}$ and the direction $\theta_{cur}^{n_i}$ at t_{cur} . The formula for predicting the position can be got as follows:

$$P_{pred}^{n_i} = P_{cur}^{n_i} + \int_{t_{cur}}^{t_{pred}} (a_{cur}^{n_i} t + v_{cur}^{n_i}) dt \quad (2)$$

The sender vehicle transmits $P_{pred}^{n_i}$ to all neighbor vehicles. Each neighbor vehicle will calculate the distance d between its actual position and predicted position at t_{pred} . If d is less than threshold ε , the node will be selected as a candidate vehicle, which can be regard as an alternate vehicle for the relay vehicle.

Then, the mobility state value $V_{MS}^{n_i}$ can be got as follows by using different criteria:

$$V_{MS}^{n_i} = D(P_{pred}^{n_i}, n_d) + MS \quad (3)$$

where $D(P_{pred}^{n_i}, n_d)$ is Euclidean distance between $P_{pred}^{n_i}$ and the position of destination vehicle n_d and MS is the mobility state function. Finally, the smallest $V_{MS}^{n_i}$ is the relay vehicle.

Although these articles studied the dynamic motion of the vehicle and made predictions for the next moment, they cannot continue to predict the future trajectory of the vehicle. And the trajectory prediction method based on deep learning can predict the trajectory of the vehicle more accurately than the method based on traditional machine learning.

3. Network Model and Problem Definition

The network model and the prediction problem of the future trajectory are introduced in detail in this section. We focus on the communication model and the routing protocol in the network model, and then the prediction problem is described in the city traffic model, whose solution will be given in the next section.

3.1 Network Model

Due to the frequent changes in the Vehicle-to-Vehicle(V2V) network topology, it is assumed that there is a delay-tolerant communication protocol between vehicles, with certain communication delays and intermittent connections. It is consistent with the frequent disconnection and connection of V2V networks, which allows vehicles to carry and forward the received data.

Assuming that each vehicle is equipped with GPS, it can ensure that all vehicles can accurately locate themselves. On the other hand, all vehicles can clearly know the relative position of each base station with GPS, which is expressed in longitude and latitude.

In the considered network, a cooperative communication strategy between V2V and Vehicle to Base Station (V2B) is considered. The communication range of the base station and the vehicle is assumed to be R_b and R_v , respectively. Once the distance between the vehicle and

the base station is less than R_b , they can directly communicate with each other. Otherwise, the vehicle can only communicate with other vehicles when the distance between vehicles is less than R_v . So, the vehicle carrying data will forward data to other appropriate vehicles until the vehicle enters the communication range of the base station. Then, the vehicle will forward the data to the base station. Through this cooperative communication strategy, the data can be finally uploaded to the data center. In our network, routing at the Bundle layer is adopted. The routing protocol is based on the "storage-carriage-forward" mechanism. The specific process is shown in Fig. 3.

The three-way handshake is used as a confirmation mechanism so that after receiving the complete data, the backbone network will respond to the source vehicle a confirmation message. That is to say, when the source vehicle receives a confirmation message from the backbone network, it stops transmitting data and deletes it from the buffer.

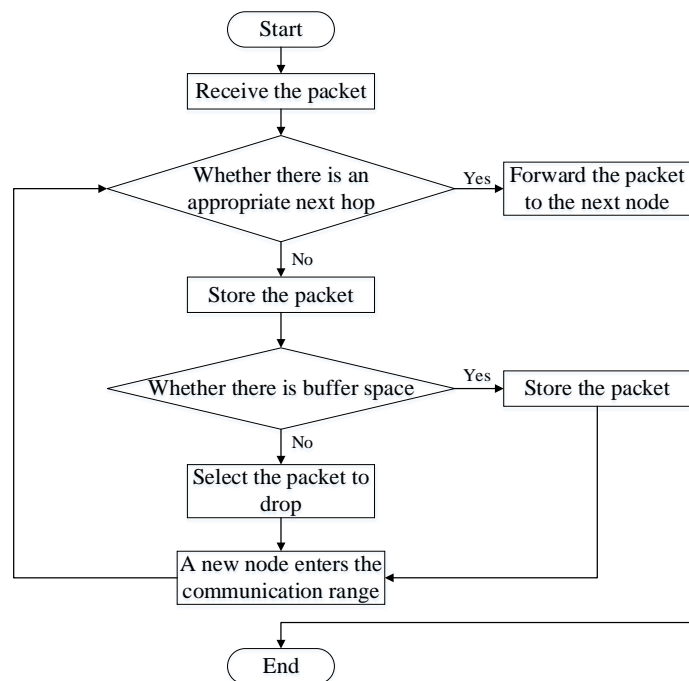


Fig. 3. Packet processing flowchart in Bundle protocol

3.2 Problem Definition

In this paper, a city traffic model is considered. n vehicles are randomly distributed on the three lanes: left-turn lane, straight lane, and right-turn lane, and two base stations are located nearby. In Fig. 4, a case is illustrated that the source vehicle C_0 needs to upload a set of emergency data to the backbone network. From the GPS information, the nearest base station B_0 is located on the left side of the next intersection, but the source vehicle does not pass through the road section. If C_0 misses B_0 , C_0 has to carry the data to the next base station B_1 , which will take a long time to store data in the buffer. In this case, the best way is to transfer the data to B_0 as soon as possible instead of carrying it and waiting. Therefore, choosing the

appropriate relay vehicle which is moving towards the B_0 can significantly reduce the transmission delay.

Once the data is transmitted to the base station through the relay vehicle, after confirming the integrity of the data, an *ACK* message will be sent to the source vehicle to tell it that its emergency data has been received, and the data temporarily stored in the buffer can be dropped.

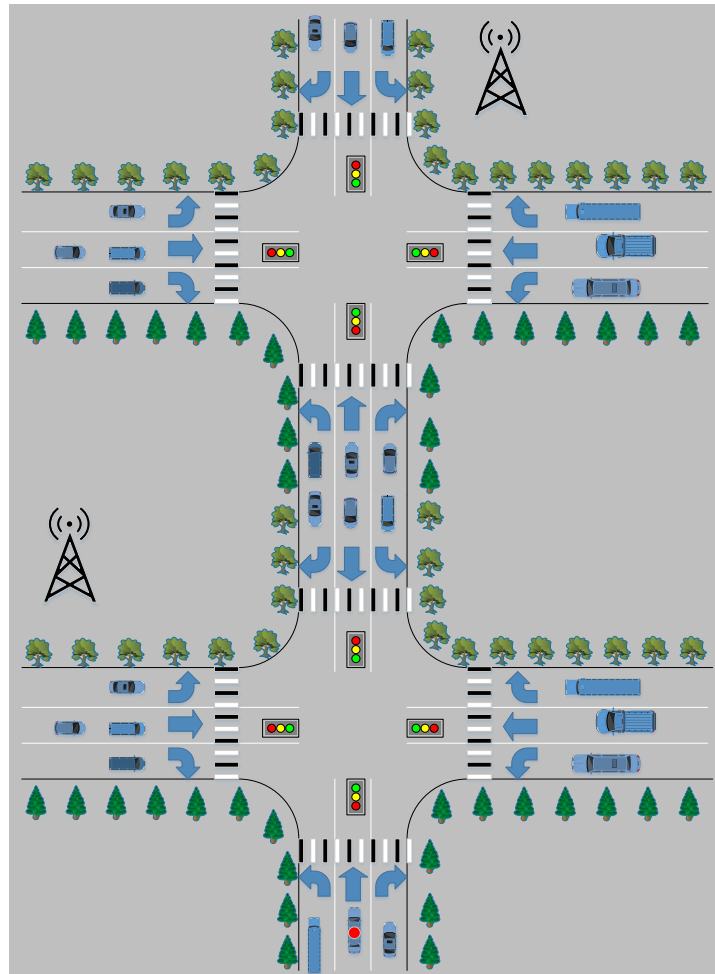


Fig. 4. The scenario proposed in the experiment

4. Proposed Method

In a DTN, if the source vehicle C_0 cannot directly transfer data to the nearest base station B_0 within a required time, it will store and carry the data until a new base station appears, which is shown as the B_1 in **Fig. 4** [27]. Supposing that the time when the source vehicle C_0 transfers data to B_1 and B_0 are t_0 and t_1 , respectively. And it can be seen that t_0 is greater than t_1 obviously. To transfer data to B_0 , a vehicle that will pass through B_0 must be selected as the relay vehicle. Because RNN exhibits good performance in predicting an object movement with continuous time-series data and its prediction effect outperforms other neural networks, we use RNN to

learn the previous driving data and make a prediction of the future trajectory. And then an optimal relay vehicle selection algorithm is proposed to select the vehicle with the shortest time consumption as the relay vehicle. The algorithm takes the driving data as the input and the minimum transmission delay as the optimization object to get the most suitable relay vehicle. As shown in Fig. 5, the relay vehicle is selected and it helps source vehicle C_o to relay data to the nearest base station B_o , which is on the left side of the next intersection shown in Fig. 4. Therefore, the timeliness of urgent data can be greatly improved and the communication delay can be reduced.

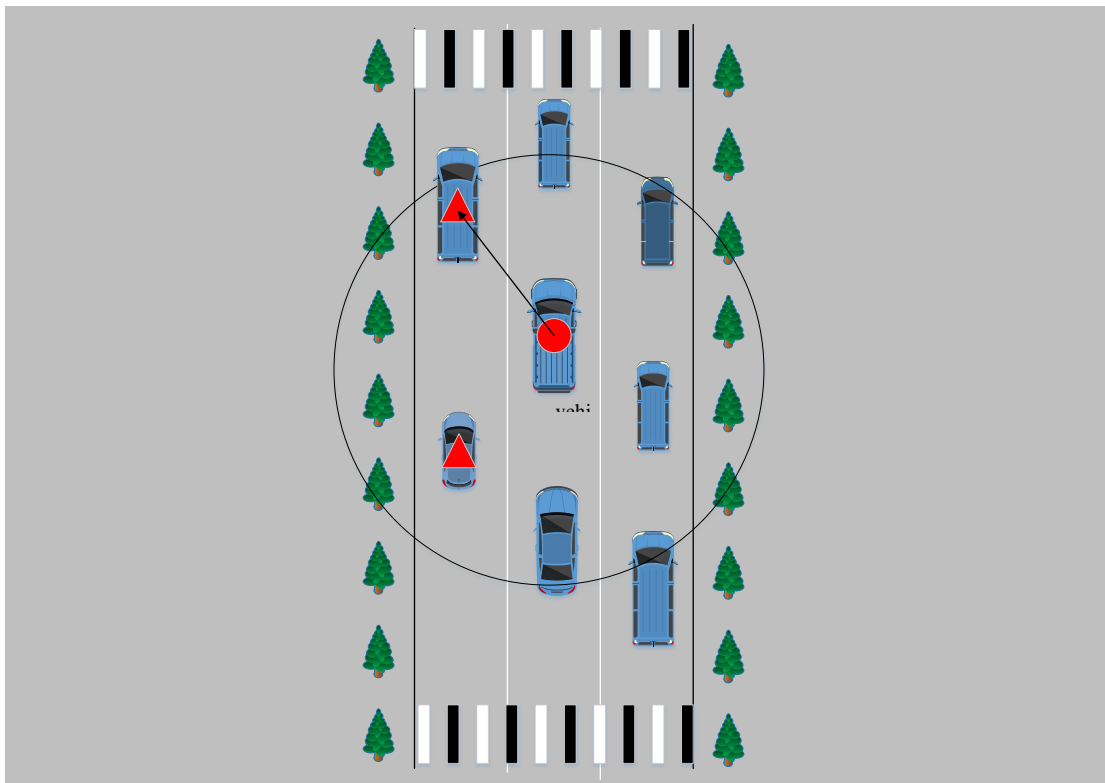


Fig. 5. Select relay vehicle within one-hop range

4.1 Recurrent Neural Network Based Trajectory Prediction

4.1.1 Recurrent Neural Network

Recurrent Neural Network is a kind of neural network dedicated to processing time-series data samples. The driving data of a vehicle is a set of time-series data, including the position, speed and direction of continuous-time points. So we utilize RNN to discover the correlation between time-series data. Each layer of RNN not only outputs to the next layer but also outputs to a hidden state. In our network, the future trajectory of the vehicle is related to the current position, speed and moving direction. Therefore, we can use these information of the vehicle in continuous time to form multiple sets of sequences. Then we predict the trajectory of the vehicle at the next moment by observing the correlation between the sequence before and after. When the speed of the vehicle is very fast, it will be farther away from the current position at

the next moment. On the contrary, if the speed is very low, it will be close to the current position at the next moment.

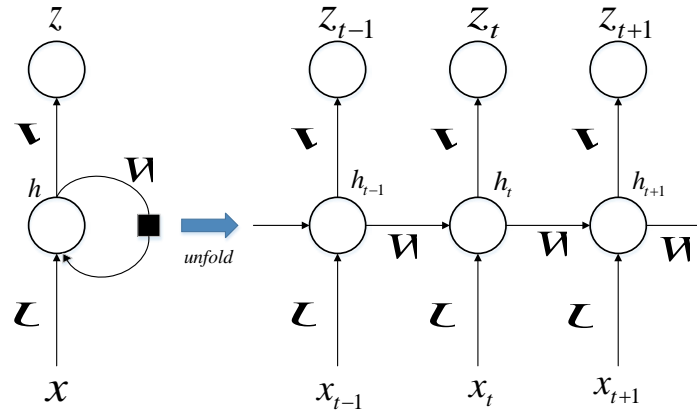


Fig. 6. A simple RNN structure

As shown in **Fig. 6**, the left part is a simple RNN structure, and the right part is the effect of expanding the whole network in form of time series. In this Recurrent Neural Network, the input at time t consists of two parts, one is the driving data x at the current time t , the other is the output h at time $t-1$. It can be seen that the prediction at time t is affected by all the driving data before t . We use the connections between neurons recurrently to form Recurrent Neural Network.

The recursion formula of RNN is as follows:

$$h_t = \tanh(Ux_t + Wh_{t-1} + b_h) \quad (4)$$

$$z_t = \text{softmax}(Vh_t + b_z) \quad (5)$$

where h_t is hidden layer vector, U is weighting matrix in the input layer, x_t is input vector, W is weighting matrix in the hidden layer and b_h is bias vector. z_t is output vector, V is weighting matrix in the output layer, b_z denotes bias vector. RNN is good at dealing with long-term dependencies. But if the interval is too long, the weighting matrix will be multiplied by the previous output, which will cause a gradient explosion problem. Thus, Long Short-Term Memory (LSTM) network, a special kind of RNN, is employed to solve such problem.

4.1.2 Long Short Term Memory Network

An RNN consisting of LSTM units is called an LSTM network. LSTM inherits the advantages of the RNN model and can solve the problem of the vanishing gradient. LSTM adds three sigmoid functions (i.e., forget gate, input gate and output gate) to each small unit. The following describes how LSTM solves long-term dependency problems in the driving data of vehicles with three gates in a neuron.

(1) Forget Gate

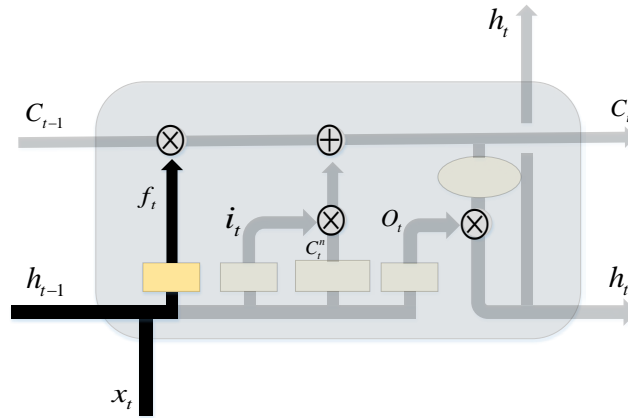


Fig. 7. Forget gate

The forget gate layer in neuron determines which information will be discarded from the state information. As shown in Fig. 7, the forget gate will read h_{t-1} at the previous unit and x_t at current time, which consists of longitude, latitude, direction, and speed of the vehicle at time t . Then the formula for memory cells f_t can be got as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{6}$$

where f_t means the forget gate at the time t , σ is the logistic sigmoid function, a threshold function for NN. W_f denotes weight matrices in the forget gate, h_t is the hidden vector at time t , b_f denotes bias vectors in the forget gate, $x = [x_1, x_2, \dots, x_n]$ is an input sequence. f_t is a number between 0 and 1. Here, 1 means completely remembered and 0 means completely forgotten. After C_{t-1} receives f_t , it will determine how much driving data will be forgotten from itself.

(2) Input Gate

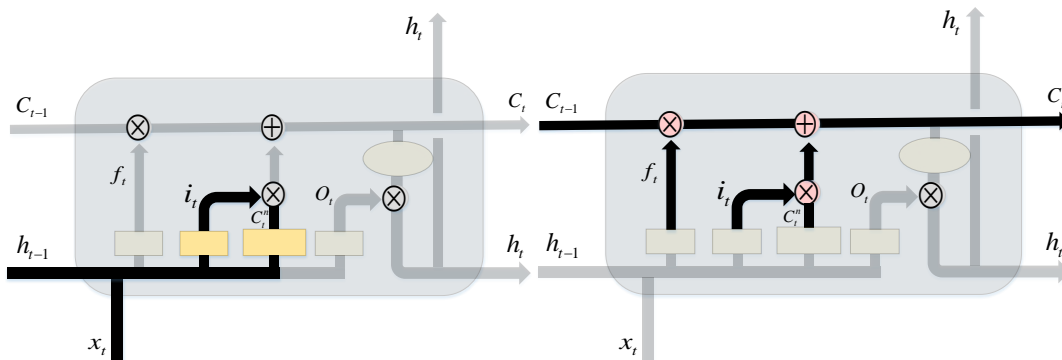


Fig. 8. Input gate

The input gate is used to add new information to the state information, which is shown in **Fig. 8**. The same with the forget gate, the current i_t and the new state information C_t^n are obtained from the input x_t and h_{t-1} , which are shown in Eq. (7) and (8), respectively.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (7)$$

$$C_t^n = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (8)$$

where i_t means the input gate at the time t and determines how much driving data will be remembered in the cell state, W_i denotes weight matrices in the input gate, b_i denotes bias vectors in the input gate. W_c is weight matrices in the state information, b_c is bias vectors in the state information.

Then the current new state information C_t is obviously calculated by the following formula:

$$C_t = f_t * C_{t-1} + i_t * C_t^n \quad (9)$$

where C_t is state information at time t , f_t means the forget gate at the time t .

(3) Output Gate

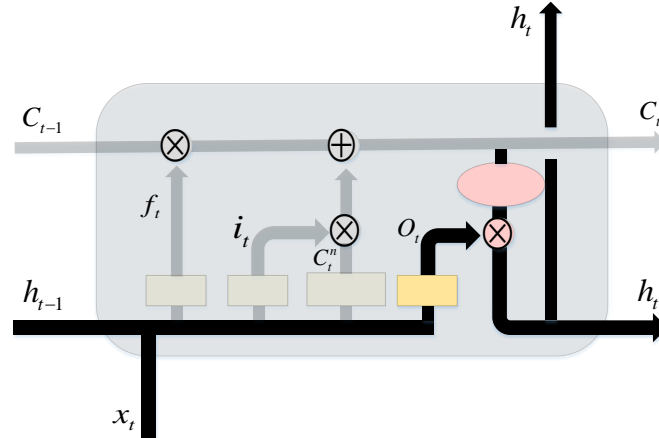


Fig. 9. Output gate

Finally, the value of the output needs to be determined by LSTM. This is decided by the output gate as shown in **Fig. 9**. The output gate adds h_{t-1} at the previous unit and x_t to a sigmoid function. Then o_t can be got as shown in Eq. (10), which determines how much information needed to output from the state information:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (10)$$

where o_t means the input gate at the time t , W_o denotes weight matrices in the output gate, b_o denotes bias vectors in the output gate. Then we multiply o_t with the output of the sigmoid gate $\tanh(C_t)$. Finally, the output information at time t can be shown in Eq. (11).

$$h_t = o_t * \tanh(C_t) \quad (11)$$

4.2 Learning and Training

As described in 4.1, LSTM can train long-term dependency information. Additionally, it can solve the vanishing gradient problems. So LSTM is suitable for predicting the trajectory of the vehicle.

When the source vehicle needs to upload emergency data to a data center, the vehicle first judges whether it will pass through the nearest base station based on the GPS information. If not, it will take a long time to transmit the data and reduce the timeliness of the emergency data, which shows that selecting the appropriate relay vehicle will facilitate the rapid upload of data. In this paper, we utilize the driving data of vehicles, which can be obtained easily by the GPS information and regard taxis as relay vehicles.

In summary, we use the RNN trajectory prediction method to predict the trajectory of all taxis within the one-hop range of the source vehicle. Since all taxis are equipped with GPS, it will record the driving data within 24 hours, including longitude, latitude, direction, and speed data. Firstly, we select these four features as the learning objects of the RNN. The values of the four features at time t and the output at time $t-1$ are taken as input data. Then, the forget gate determines which information will be discarded from the state information C_t by the value of f_t , the input gate determines how much information will be added to the state information C_t by the value of i_t . Meanwhile, the state information C_t will be updated. Finally, the state information is processed through \tanh and multiplied by the output of the sigmoid function, and the output gate will only output the part determined. In general, Recurrent Neural Network training model can be shown in Fig. 10.

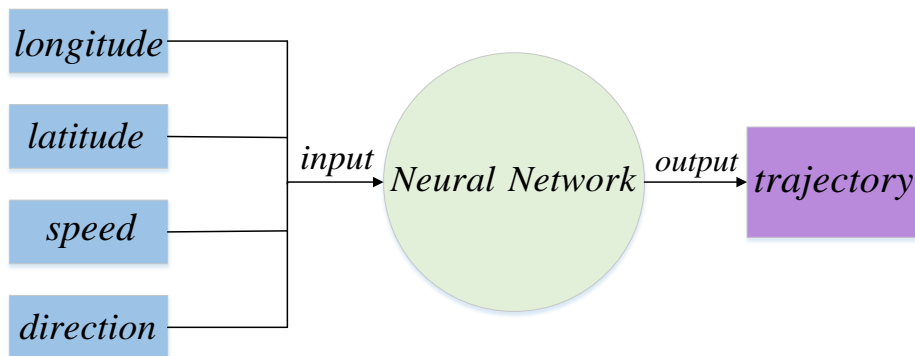


Fig. 10. Recurrent Neural Network training model

4.3 Relay Vehicle Selection Algorithm

As mentioned in 4.2, the predicted trajectory is composed of consecutive time positions, so the trajectory can be represented as a collection of positions, which is represented by longitude and latitude. The set can be formalized mathematically as follows:

$$Vehicle_m = \{(log_{m1}, lat_{m1}), (log_{m2}, lat_{m2}), \dots, (log_{mn}, lat_{mn})\} (n > 0, 0 < m \leq M) \quad (12)$$

where m is the ID of the vehicle, n is the time point, M is the total number of vehicles.

Meanwhile, each position corresponds to a point in time, so the set of time points of the vehicle can be expressed as:

$$T_m = \{t_{m1}, t_{m2}, \dots, t_{mn}\} \quad (13)$$

When the vehicle enters the communication range of a base station, the data communication can be performed. In order to numerically analyze, the communication range of a base station is expressed as a set of position points. The set is defined as Eq. (14):

$$BS_i = \{(log_{i1}, lat_{i1}), (log_{i2}, lat_{i2}), \dots, (log_{ij}, lat_{ij})\} (m, n > 0) \quad (14)$$

Where i denotes the ID of the base station and j is the number of the position.

Next, it is necessary to make a judgment on the relationship between (log_{mn}, lat_{mn}) and BS_i . If

$$(log_{mn}, lat_{mn}) \in BS_i, \quad (15)$$

we put the time point of vehicle m that satisfies Eq. (15) into a new set as follows:

$$T_k = \{t_1, t_2, \dots, t_z\} (z \leq n, T_k \in T_m). \quad (16)$$

Then, if the vehicle m is selected as the relay vehicle, the minimum time it takes to transmit the data to the base station can be obtained in Eq. (17):

$$t_{min}^m = \min(T_k) \quad (17)$$

The above calculation process is performed on all vehicles within the one-hop range of the source vehicle, and the minimum time required for all vehicles to transmit data to the base station is obtained. After grouping them into a new collection, each item of the set is the shortest time it takes for the vehicle to pass the data.

$$T_{min} = \{t_{min}^1, t_{min}^2, \dots, t_{min}^m\} \quad (18)$$

$$t_{min} = \min(T_{min}) \quad (19)$$

Finally, we will select the vehicle corresponding to t_{min} as the relay vehicle, which completes the communication task of transmitting emergency data to the base station. And the time it takes to pass data is ensured to be the shortest of all vehicles within a hop range. The specific algorithm can be seen in Algorithm 1:

Algorithm 1 Relay vehicle selection algorithm

Input: $Vehicle_m = \{(log_{m1}, lat_{m1}), (log_{m2}, lat_{m2}), \dots, (log_{mn}, lat_{mn})\}$,

$$T_m = \{t_{m1}, t_{m2}, \dots, t_{mn}\},$$

$$BS_i = \{(log_{i1}, lat_{i1}), (log_{i2}, lat_{i2}), \dots, (log_{ij}, lat_{ij})\}$$

- 1 For ($m=1, m \leq M, m++$)
- 2 If $(log_{mm}, lat_{mm}) \in BS_i$ then
- 3 Record the time point corresponding to the position that satisfies the condition into a new set:
 $T_k = \{t_1, t_2, \dots, t_z\}$
- 4 Get the shorest time for vehicle $m : t_{min}^m = \min(T_k)$
- 5 Search the minimum time for each vehicle : $T_{min} = \{t_{min}^1, t_{min}^2, \dots, t_{min}^m\}$

Output: The vehicle which has the minimum t_{min}^m

By summarizing, the entire flow chart of the proposed relay vehicle selection algorithm is shown in Fig. 11:

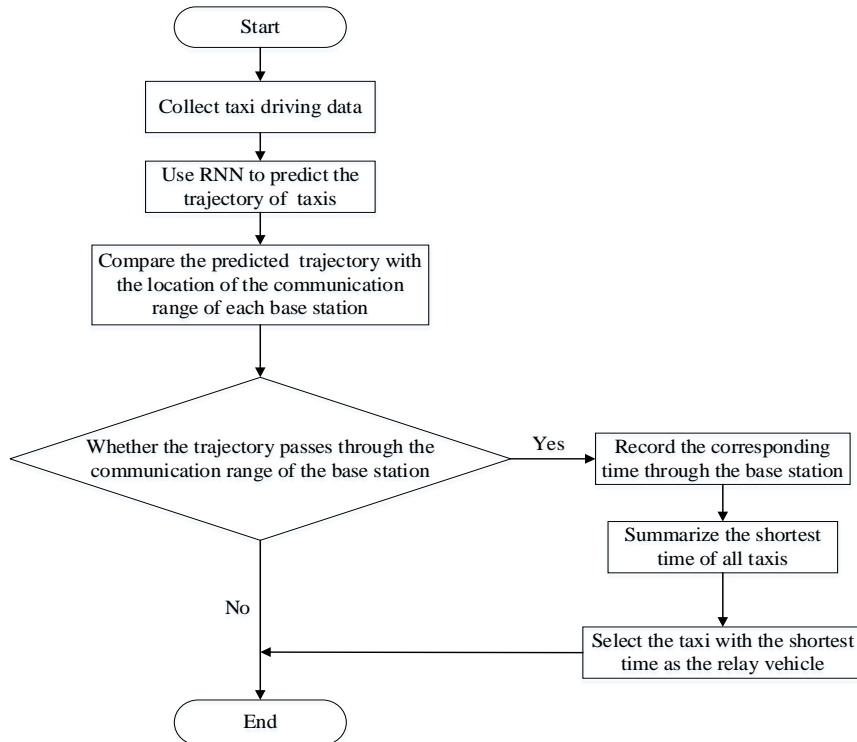


Fig. 11. Flow chart of the proposed relay vehicle selection algorithm

5. Simulation

5.1 Data Processing

In order to accurately predict the vehicle's trajectory, we need to collect the vehicle's driving data, including longitude, latitude, speed, direction. In reality, the collection of data on private vehicles is unrealistic because it involves personal privacy. We use the data set of taxis to replace, because it is relatively more transparent. The data set comes from the Smart City Research Group of Hong Kong University of Science and Technology [28]. It contains the driving trajectories of 4316 groups of taxis within 24 hours.

First, in order to eliminate the problem of accuracy reduction caused by the proportions of the different fields in the data set and eliminate the dimensional impact between the indicators, we use Min-max normalization to resolve comparability between data metrics and limit the results between 0 and 1. The Min-max normalization is as follows:

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (20)$$

Where $\max(x)$ is the maximum value of the sample data and $\min(x)$ is the minimum value of the sample data.

Then the four characteristic values are regularized to make the data fall within a certain statistical interval. The LSTM neural network is used to train the first 80% data set. Then it predicts where the vehicle will arrive in the future and compares it to the actual arrival position. Root mean square error (RMSE) is used to evaluate the accuracy of the prediction. When the value of RMSE is smaller, the accuracy of the prediction model describing the experimental data will be higher. The formula is shown in Eq. (21).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_{obs,i} - X_{model,i})^2}{n}} \quad (21)$$

Where $X_{obs,i}$ is the observed value, $X_{model,i}$ is the actual value.

5.2 Simulation Results

There are 200 vehicles in the experiment, each vehicle randomly selected 10 groups driving data from the 4316 groups of data without repetition. We use the first 8 groups of data for training and the 9-th group and the 10-th group for testing. The position at $t=0$ of the 9-th or 10-th group is regarded as the starting position of the vehicle in the experiment, and the position of the first vehicle is regarded as the origin of the coordinate system. The east-west direction and the north-south direction are taken as the x-axis and the y-axis, respectively. Then other vehicles are mapped to the coordinate system according to their position.

To demonstrate the performance of the method, one vehicle is selected randomly from 200 vehicles at a time as the source vehicle. When the source vehicle needs to relay data, it will broadcast the relay request within the communication range. Then, the vehicle receiving the relay request immediately reads the driving data of the first three hours in the memory and inputs them into LSTM prediction model to predict the driving trajectory. Finally, each vehicle sends its predicted trajectory to the source vehicle. And the source vehicle selects the

vehicle with the shortest time as the relay vehicle according to the proposed relay vehicle selection algorithm. Each experiment simulation is carried out 100 times and takes the average value as the final result to verify the superiority of the algorithm.

Fig. 12 shows the effect of trajectory prediction using LSTM, where the blue line is the real value and the orange line is the predicted value. As shown in **Fig. 12**, the 9-th group and the 10-th group are used for testing after the model training is completed. Then we use the first 80% data of the 10-th group as input and compare the predicted value of the output with the real value. **Fig. 12(a)** is the result on time and longitude and **Fig. 12(b)** is the result on time and latitude. **Fig. 12** indicates that the predicted trajectory is basically consistent with the actual trajectory, which demonstrates the accuracy of our proposed method.

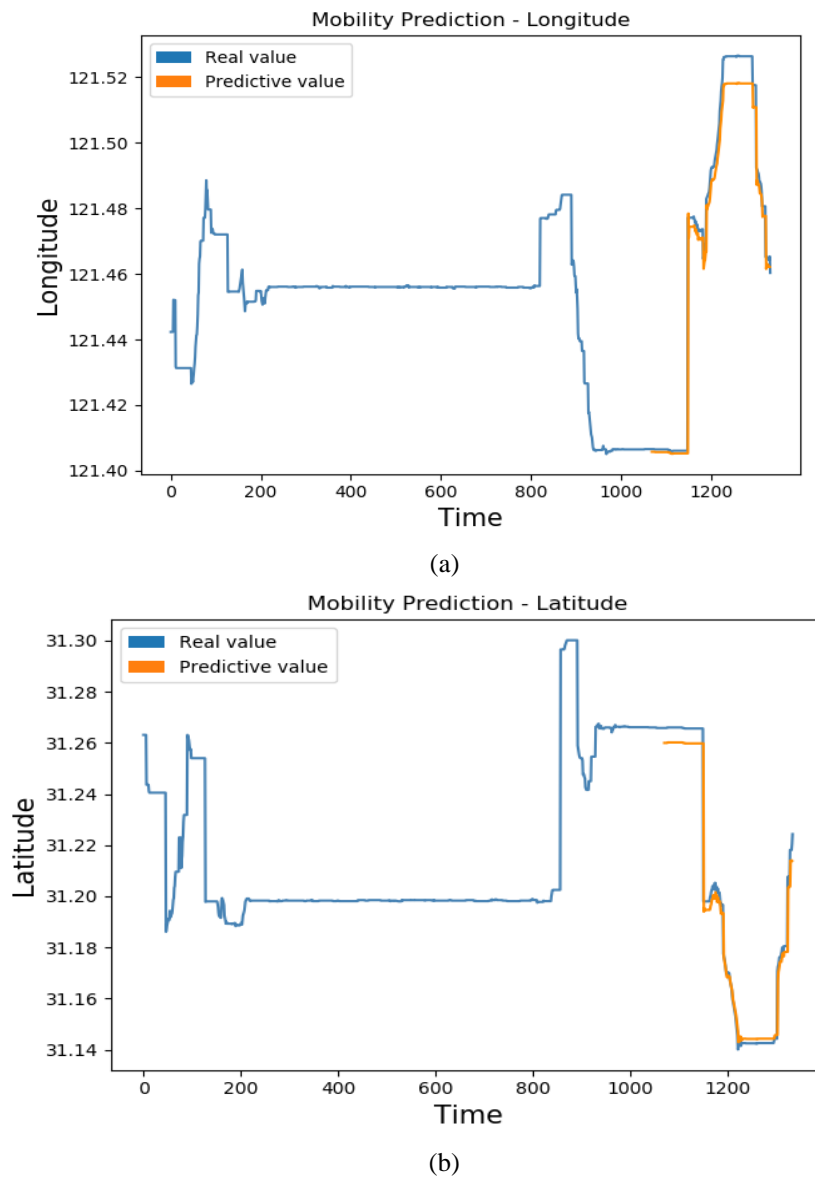


Fig. 12. The predicted result of trajectory based on LSTM.
(a) longitude, (b) latitude

Fig. 13 shows the spiraling trajectory of the vehicle, where the longitude and latitude are taken as the x-axis and the y-axis, respectively, and the time t is taken as the z-axis. This is because the data of the vehicle comes from the taxi, and the taxi is likely to stay in an area for a long time due to the traffic jam, resulting in the spiraling trajectory. In **Fig. 13(a)**, the blue dashed line is the real driving trajectory, and the green line is the predicted trajectory. The comparison result of actual driving trajectory and the predicted trajectory shows that the predicted trajectory is good. For a more detailed observation, we take out the part from 1200s to 1260s of **Fig. 13(a)**, as shown in **Fig. 13(b)**. It can be seen that the predicted trajectory and position are very accurate, but there is a little time delay.

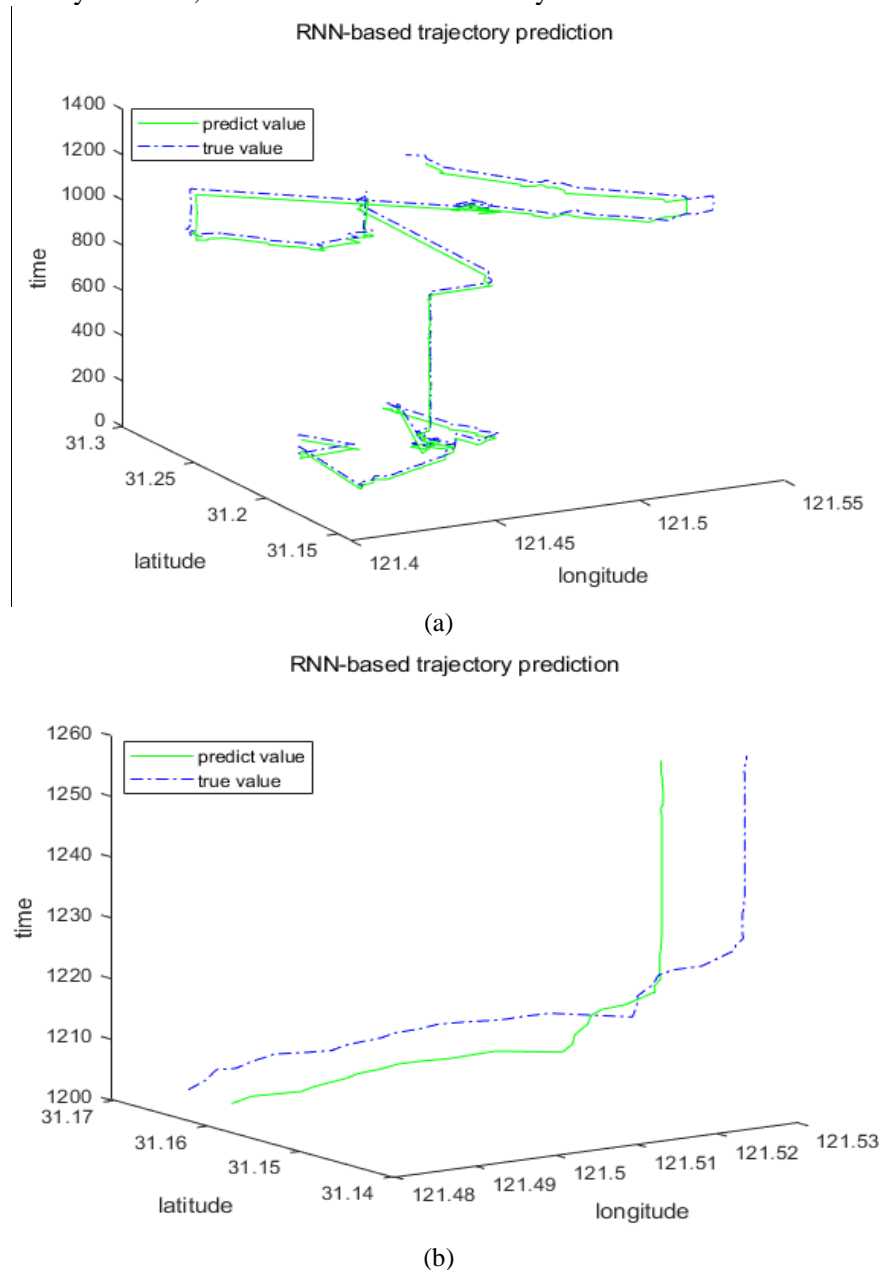


Fig. 13. The result of RNN-based trajectory prediction. (a) from 0s to 1260s, (b) from 1200s to 1260s

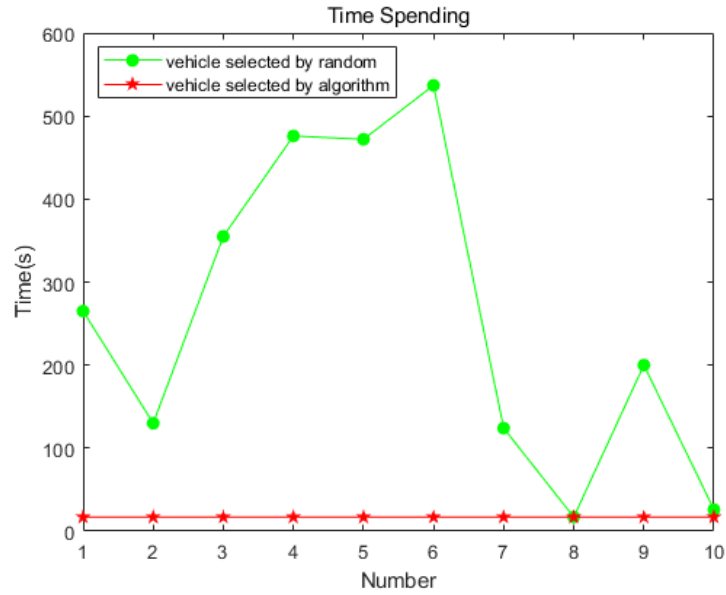


Fig. 14. Time spent passing data to the base station

Fig. 14 shows the time it takes for the vehicle to transmit data to the base station. Assuming that there are 10 taxis in the one-hop of the source vehicle, and they are distributed in the road randomly. The performance of the proposed algorithm is compared with a random method, which randomly selects a vehicle as the relay vehicle, and the comparison experiment conducts 10 times. In **Fig. 14**, the red line is the vehicle chosen according to our proposed method, and the green line is the vehicle chosen according to the random method. As can be seen, the vehicle selected by the random method has only 20% chance of choosing a vehicle with a shorter time and 80% chance of choosing a vehicle with a longer time than our proposed method. Therefore, the relay vehicle obtained by the relay vehicle selection algorithm can effectively reduce the network communication delay and improve the timeliness of the emergency data.

Table 1. The RMSE of longitude and latitude

	Longitude	Latitude
Train Score	0.016 RMSE	0.010 RMSE
Test Score	0.005 RMSE	0.003 RMSE

As shown in **Table 1**, the RMSE of the relay vehicle selected by the algorithm is very low. The RMSE for training is only 1%-2%, and the RMSE for testing is less than 1%. Therefore, the prediction of our proposed algorithm is accurate. Compared with carrying and forwarding data to BS, the optimal relay vehicle selection algorithm we proposed can effectively reduce the delay.

6. Conclusion

In this paper, we proposed a relay vehicle selection algorithm based on RNN trajectory prediction to reduce the communication delay in VANTE and improve the timeliness of data. The proposed method adopts a distributed scheme for vehicle trajectory prediction. When a vehicle needs to relay data, it broadcasts the relay signal within the communication range.

Then, the vehicle receiving the relay signal immediately reads the driving data of the first three hours in the memory and inputs them into the prediction model to predict the driving trajectory. In addition, each vehicle forwards them to the source vehicle after getting the predicted driving trajectory, and the source vehicle selects the optimal relay vehicle according to the relay vehicle selection algorithm. Finally, the source vehicle forwards the data to the relay vehicle and then to the base station. And the base station replies an *ACK* message to complete the communication when the relay vehicle completes transmitting the data.

The simulation results show that the proposed method has high accuracy for vehicle mobility prediction, and it can reduce communication delay and improve the timeliness of data.

References

- [1] S. Hung, X. Zhang, A. Festag, K. Chen, and G. Fettweis, "Virtual Cells and Virtual Networks Enable Low-Latency Vehicle-to-Vehicle Communication," in *Proc. of IEEE Global Communications Conference*, pp. 1-7, December, 2017. [Article \(CrossRef Link\)](#).
- [2] H. Oh, J. Yoo, C. Kim, and S. Ahn, "A novel mobility management for seamless handover in vehicle-to-vehicle/vehicle-to-infrastructure(V2V/V2I) networks," in *Proc. of International Symposium on Communications and Information Technology*, pp. 259-260, October, 2009. [Article \(CrossRef Link\)](#).
- [3] S. Al-Sultan, M. Al-Doori, and H. Al-Bayatti, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications*, vol. 37, no. 1, pp. 380-392, January, 2014. [Article \(CrossRef Link\)](#).
- [4] N. Aljeri and A. Boukerche, "Movement prediction models for vehicular networks: An empirical analysis," *Wireless Networks*, vol. 25, no. 6, pp.1505–1518, 2019. [Article \(CrossRef Link\)](#).
- [5] I. Halim and H. Fahmy, "Prediction-based protocols for vehicular ad hoc networks: Survey and taxonomy," *Computer Networks*, vol. 130, no. 1, pp. 34-50, October, 2018. [Article \(CrossRef Link\)](#).
- [6] Y. Li, D. Jin, P. Hui, Z. Wang, and S. Chen, "Limits of predictability for large-scale urban vehicular mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2671-2682, December, 2014. [Article \(CrossRef Link\)](#).
- [7] Y. Zhu, R. Jiang, J. Yu, and Li. Zhi, "Geographic routing based on predictive positions in vehicular ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1-9, August, 2014. [Article \(CrossRef Link\)](#).
- [8] G. Xue, Y. Luo, J. Yao, and M. Li, "A novel vehicular position prediction based on mobility patterns for routing in urban VANET," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1-14, December, 2012. [Article \(CrossRef Link\)](#).
- [9] H. Feng, C. Liu, Y. Shu, and O. Yang, "Position prediction of vehicles in VANETs using a Kalman filter," *Wireless Personal Communications*, vol. 80, no. 2, pp. 543-559, August, 2015. [Article \(CrossRef Link\)](#).
- [10] J. Zhang, M. Ren, H. Labiod, and L. Khoukhi, "Link duration prediction in VANETs via AdaBoost," in *Proc. of IEEE Global Communications Conference*, pp. 1-6, December, 2017. [Article \(CrossRef Link\)](#).
- [11] H. Yang, M. Yu, and X. Zeng, "Link available time prediction based GPSR for vehicular ad hoc networks," in *Proc. of IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 293-298, May, 2017. [Article \(CrossRef Link\)](#).
- [12] G. Mosqueda, G. Santos, and R. Villarreal, "Reliable freestanding position-based routing in highway scenarios," *Sensors MDPI*, vol. 12, no. 11, pp. 14262-14291, December, 2012. [Article \(CrossRef Link\)](#).

- [13] S. Cha and K. Lee, "Position prediction for grid-based geographical routing in vehicular ad hoc networks," in *Proc. of International Conference Grid and Distributed Computing*, pp. 35-41, January, 2011. [Article \(CrossRef Link\)](#).
- [14] H. Menouar, M. Lenardi, and F. Filali, "Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks," in *Proc. of Vehicular Technology Conference*, pp. 2101-2105, November, 2007. [Article \(CrossRef Link\)](#).
- [15] A. Bohlooli and K. Jamshidi, "A GPS-free method for vehicle future movement directions prediction using SOM for VANET," *Applied Intelligence*, vol. 36, no. 3, pp. 685-697, April, 2012. [Article \(CrossRef Link\)](#).
- [16] A. Rossi, G. Barlacchi, M. Bianchini, and B. Lepri, "Modelling Taxi Drivers' Behaviour for the Next Destination Prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-10, June, 2019. [Article \(CrossRef Link\)](#).
- [17] Z. Xia, Z. Hu, and J. Luo, "UPTP Vehicle Trajectory Prediction Based on User Preference Under Complexity Environment," *Wireless Personal Communications*, vol. 97, no. 3, pp. 4651-4665, September, 2017. [Article \(CrossRef Link\)](#).
- [18] D. Cao, Z. Bin, J. feng, and L. bao, "A robust distance-based relay selection for message dissemination in vehicular network Wireless Networks," *Wireless Networks*, vol. 26, pp. 1755-1771, 2020. [Article \(CrossRef Link\)](#).
- [19] S. He, K. Xie, K. Xie, C. Xu, and J. Wang, "Interference-aware Multi-source Transmission in Multi-radio and Multi-channel Wireless Network," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2507-2518, 2019. [Article \(CrossRef Link\)](#).
- [20] L. Fan, L. Bin, and F. Chao, "Mobile relay deployment in multihop relay networks," *Computer Communications*, vol. 112, no. 1, pp. 14-21, August, 2017. [Article \(CrossRef Link\)](#).
- [21] C. Feng, S. Arshad, S. Zhou, D. Cao and Y. Liu, "Wi-multi: A Three-phase System for Multiple Human Activity Recognition with Commercial WiFi Devices," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7293 - 7304, May, 2019. [Article \(CrossRef Link\)](#).
- [22] J. Su, Y. Chen, Z. Sheng, Z. Huang, and A. Liu, "From M-ary query to bit query: a new strategy for efficient large-scale RFID identification," *IEEE Transactions on Communications*, pp. 1-13, January, 2020. [Article \(CrossRef Link\)](#).
- [23] J. Wang, Y. Gao, X. Yin, F. Li and H. Kim, "An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1-9, December, 2018. [Article \(CrossRef Link\)](#).
- [24] J. Su, Z. Sheng, A. Liu, Z. Fu, and Y. Chen, "A time and energy saving based frame adjustment strategy (TES-FAS) tag identification algorithm for UHF RFID systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 2974-2986, 2020. [Article \(CrossRef Link\)](#).
- [25] J. Su, Z. Sheng, V. Leung, and Y. Chen, "Energy efficient tag identification algorithms for RFID: survey, motivation and new design," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 118-124, February, 2019. [Article \(CrossRef Link\)](#).
- [26] K. Park, H. Kim, and S. Lee, "Mobility State Based Routing Method in Vehicular Ad-Hoc Network," in *Proc. of 2015 IEEE International Conference on Mobile Services*, pp. 473-474, August, 2015. [Article \(CrossRef Link\)](#).
- [27] S. Tornell, C. Calafate, and J. Cano, "DTN protocols for vehicular networks: An application oriented overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 868-887, December, 2014. [Article \(CrossRef Link\)](#).
- [28] <https://www.cse.ust.hk/scrg/>
- [29] D. Gavalas, C. Konstantopoulos, and G. Pantziou, "Mobility prediction in mobile ad-hoc networks," *Next Generation Mobile Networks and Ubiquitous Computing*, pp. 226-240, 2011. [Article \(CrossRef Link\)](#).
- [30] R. Amin, D. Ripplinger, and D. Mehta, "Design Considerations in Applying Disruption Tolerant Networking to Tactical Edge Networks," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 32-38, October, 2015. [Article \(CrossRef Link\)](#).

- [31] K. Zhang, Y. Zhang, and S. Wan, "Research of RSSI indoor ranging algorithm based on Gaussian - Kalman linear filtering," in *Proc. of 2016 IEEE Advanced Information Management*, October, 2016. [Article \(CrossRef Link\)](#).
- [32] W. Su, S. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, pp. 3-30, January, 2001. [Article \(CrossRef Link\)](#).



Zilong Jin received the B.E. degree in computer engineering from the Harbin University of Science and Technology, China, in 2009, and the M.S. and Ph.D. degrees in computer engineering from Kyung Hee University, South Korea, in 2011 and 2016, respectively. He is currently an Assistant Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, China. His research interests include wireless sensor networks, mobile wireless networks, and cognitive radio networks.



Yuxin Xu received his B.S. degree in information management and information system from the Binjiang College, Nanjing University of Information Science & Technology, China, in 2018. Now he is a master student in School of Computer and Software, Nanjing University of Information Science & Technology. His main research interests include internet of vehicles and edge computing.



Xiaorui Zhang was born in 1979. She received her Ph.D. degree in Instrument Science and Technology Department of Southeast University, Nanjing, China, in 2010. She also received her B.S. and M.S. degrees from Henan University of Science and Technology, Henan, China, in 2004 and 2007, respectively. Now she is a Professor and supervisor for master student in School of Computer and Software, Nanjing University of Information Science & Technology. Her research interests include the virtual reality, human-computer interaction, and image processing.



Jin Wang was born in 1979. He received the B.S. degree in electronic engineering and the M.S. degree in information engineering from the Nanjing University of Posts and Telecommunications, and the Ph.D. degree in computer engineering from Kyung Hee University, in 2010. He is currently a Professor with the Changsha University of Science and Technology. Since 2002, he has been engaged in wireless sensor networks, MANET, VANET research, and had accumulated and systematic research works in network energy-saving routing algorithms, data fusion, computational intelligence, intelligent medical treatment, and other fields.



Lejun Zhang received the M.S. degree in computer science and technology from the Harbin Institute of Technology and the Ph.D. degree in computer science and technology from Harbin Engineering University. He is currently a Professor with Yangzhou University, China. His research interests include computer networks, social network analysis, and information security.