A Learning-based Vehicle-Trajectory Generation Method for Vehicular Networking

Liang Zhao[1], Yufei Liu[1], Ahmed Al-Dubai[2], Zhiyuan Tan[2], Geyong Min[3], and Lexi Xu[4]
[1] School of Computer Science, Shenyang Aerospace University, China
[2] School of Computing, Edinburgh Napier University, UK
[3] College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK
[4] China Unicom, Beijing, China

***Abstract*—**With the rapid development of mobile applications, networking technologies have been constantly evolved to offer a more convenient way of sharing information and online-communication anytime and anywhere. Vehicular networks have the potential to become one of the important carriers of future mobile networks. The performance of current vehicular networks has been widely evaluated through simulation experiments due to the high cost and impracticality of other experimental approaches. The most paramount factors of vehicle networks are the authenticity of simulative evaluation, where the mobility of the vehicles is the first significant feature (i.e., the nodes of the vehicular network) that must be properly considered. However, generating the corresponding real mobility datasets has always been a big challenge although it is vital to the simulations of vehicular networks. Therefore, in this paper, we propose a learning-based generation method that can be used to build the vehicle-trajectory data for variety of vehicle densities. Firstly, with analyzing the road bayonet data, we obtain the hidden pattern between road traffic and time. Secondly, we deploy Vissim (a well-known traffic simulator) to generate the experimental data by considering the urban functional areas for the origins of vehicles. The generated experimental data are learned by Extreme Learning Machine (ELM), and the weight matrix of the parameters is obtained, which presents the impact of the experimental parameters on the simulation results. We prove the effectiveness of our method by comparing the generated vehicle-trajectory datasets with the vehicle density predicted by the weight matrix and the realistic traffic flow model.

Keywords*—vehicle mobility, dataset generation, vehicular networks, Vissim.*

I. INTRODUCTION

With the increase in the volume of vehicular traffic and its serious environmental impact, the most concerned issue for transportation departments in most nations and regions is traffic management. With the rapid development of the recent wireless vehicular communication capabilities, an important part of the Intelligent Transportation System (ITS) is to enable efficient traffic management. As a fundamental element of ITS, vehicular networking technology has attracted much attention from academia and industry [1]. Vehicular network establishes the communication platform between vehicles, which improves traffic efficiency and brings reliable safety and multiple conveniences for the drivers and thereby makes the experience of travellers more comfortable [2]. For example, it is applied to support the safety driving by obtaining information on other vehicles (such as speed, direction, and position) and observing road condition. In both academia and industry, the performance evaluation of vehicular networking technologies should be taken under large-scale scenarios, which makes the real field tests of vehicular networks costly and impractical. Hence, simulation is the current optimal choice for the validation of vehicular network architecture and protocols. In this context, due to the dynamic and road-constrained nature of vehicles, the trajectory of vehicles is the key to achieving the close-to-real performance results in the simulation. Till now, it is still infeasible to collect dense vehicular trajectories where floating car data and car electric plate data are both sparse. The trajectory of vehicles has shown the big impact on the evaluation of routing protocols of vehicular networks where small changes of the trajectories or incomplete representation of vehicle traffic could cause the huge waves are shown in different performance metrics [3]-[5]. Therefore, generating real-world vehicle-trajectory has attracted tremendous research efforts.

There is great substantial progress in the quality of vehicle-trajectory in the studies of vehicular networks over the last few years including the random mobility models employed [3], the stochastic mobility of realistic road topology [6], and the microscopic vehicular model [7]. The trajectories generated from the above models are input in professional simulation environments. Also, in parallel with the evolution of vehicle-trajectory, vehicle traffic dataset in the real world have grown in number and scale.

In this work, we propose a learning-based generation method of the vehicle-trajectory dataset with considering the bayonet data and Extreme Learning Machine (ELM) [8] [9], which breaks the dependence of mobile simulation on the investigation report [10]. To make the experimental data more representative and realistic, the impact of urban functional areas on vehicle travel is considered when we set the origin point of vehicles. Besides, we propose a new scheme to tune the parameters of traffic simulation tools, e.g., Vissim. This method provides a feasible way to generate fixed-width vehicle trajectory data, which is no longer cumbersome and more practical. The weight matrix of parameter and simulated traffic flow model obtained by this method can be used to generate mobile trajectory datasets of different road topologies and vehicle densities and does not need to be modeled again. Also, the vehicle-trajectory dataset obtained by our method facilitates the simulation of communication protocols.

Our main contributions can be summarized as follows:

- A novel learning-based vehicle-trajectory generation method is proposed.
- We considered the urban function area when setting the origin point of the vehicle.
- Compared with real data, the proposed method achieves promising performance.

The rest of the paper is organized as follows. The related work is discussed in Section II. In Section III, vehicle-trajectory generation method is given in detail. Section IV describes the simulation experiment and the performance results. Conclusions are presented in Section V.

978-1-5386-7238-9/18/$31.00 ©2019 IEEE

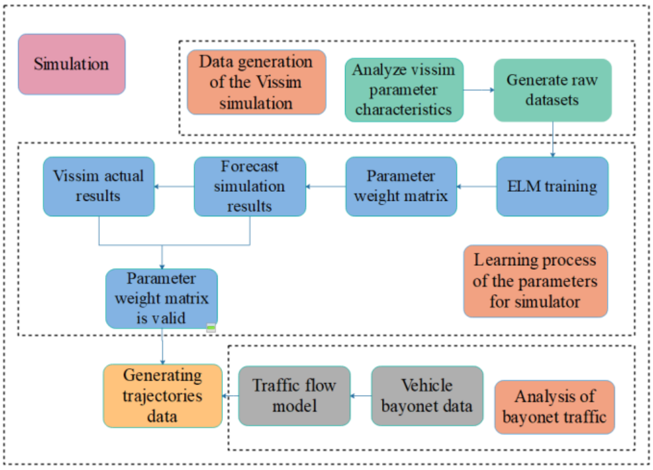

Fig. 1 Flow diagram of datasets generation methodology

II. RELATED WORK

Nowadays, with the progress of modern science and technology, there is more road detection equipment deployed to collect data. As a consequence, a massive variety of mobility trace of vehicles is stored, where R&D has made many efforts to apply these data.

Socio-traffic surveys represent a momentous source of information for obtaining vehicle-trajectory datasets. Existing study has made many efforts to generate large-scale vehicle-trajectory datasets that compass very large urban areas, and are realistic also from a macroscopic point of view. In [11], the authors propose the framework for vehicular mobility scenario generation, namely, En Route. By applying the framework to traffic camera dataset, they model the traffic demand of a large-scale urban scenario (city of London). Origin-Destination (OD) matrix is employed to identify the traffic demand, including the start time, the coordinates of origin, and destination of each vehicle trip in the simulation area. Currently, the stochastic traffic demand model and the shortest path-based algorithm are most commonly methods to establish traffic demand. However, these methods may lead to unrealistic vehicle flow and deviation of simulation results. The authors in [12] adopt the precise road traffic information gathered from the flow counters and the OD matrix model to simulate the vehicle-trajectory in Luxembourgian. According to the official traffic flow data, the real traffic data is denser than the generated vehicle-trajectory data in some regions.

The real-world car traces fetched directly from retrieving vehicle positions of GPS receivers has been employed for research, e.g., Floating Car Data (FCD). In [13], the authors analyze the FCD of Beijing, China, by dividing the urban functional areas. They apply the Gravity Model to predict the OD matrix of vehicles. As a result, they reproduce the scenario by using the simulation tool, Simulation of Urban Mobility (SUMO) [14] to generate the mobility data of urban vehicles. The generated vehicle movement data is compared with the traffic in Beijing. The results show that the method produces closed-to-reality vehicle-trajectory in most areas, but only not for the region in areas such as railway stations or bus stations. In [15], the authors generate the trajectory of social vehicles that last for 24 hours in Köln, where the traffic flow for each road is set by considering the micro-driver behavior and the macro traffic flow. Then they use the Gawron's algorithm to balance the traffic flow. Finally, they demonstrated that this dataset considering microscopic behavior has a significant impact on network performance assessment. However, if there is no accurate official data provided by the government, it is impossible to generate dataset by their method.

Most of the previously discussed vehicle-trajectory datasets are synthetically generated by macroscopic traffic data into a microscopic mobility simulator. However, vehicle-trajectory prediction of objects has gradually become active research field and has been widely supported in moving objects databases. In [16], aiming at the disadvantages of the existing predictive trajectory algorithm, an improved algorithm based on Hidden Markov model-based Trajectory Prediction (HMTP) is proposed, called HMTP∗, which captures the parameters required by the real-world scenarios regarding objects with dynamically changing speed. Also, a vehicle-density-based method of trajectory division is employed to improve the efficiency of vehicle-trajectory prediction. Extensive experiments are conducted to demonstrate the effectiveness of the proposed algorithm. The results of these experiments prove that the algorithm has a higher positioning accuracy than the general algorithm. However, this approach cannot be employed to predict the future location of moving nodes due to the spatiotemporal characteristic of the trace data. The trajectory of the dynamic object is predicted by employed Markov chain [17]. To compute the K-order transition matrices and predicted dynamic paths from constructing a Markov chain, the historical information is transmitted into directed connected graphs. However, this method heavily depends on the massive data of history trajectories. Thus, the feasibility and accuracy cannot be guaranteed.

By analyzing and discussing the existing mobility trace, we summarize the general regulation of excellent vehicle-trajectory datasets. Overall, the ultimate vehicular mobility trace for vehicular network simulation should feature all of the following.

- The integrity of vehicle traffic should be included in the vehicle trajectory data.
- In the case of higher time granularity, an order-of-second level of precision tracking of the position of each vehicle is a least.
- The realistic performance of the micro-behavior of individual drivers should be represented, and their interaction with other drivers and the road conditions are also reflected.

The above survey shows that current trajectory obtained through real-world tracking cannot meet the requirements of the first two rule, as they are limited to subsets of the vehicle-trajectory datasets, e.g., exhibit reduced temporal detail, or update the position once a minute or a few minutes. These datasets are today mainly employed for the performance evaluation of high latency or opportunistic mobile communication networks. However, for a more general use case, significantly higher granularity and penetration rates are needed. However, it is easy to foresee that the public disclosure of such mobile datasets will be hindered because of privacy concerns and market rules, similar to today happening of logs collected by mobile network operators. The limitation of real-world mobile trace forces us to resort to generated synthetic vehicle-trajectory datasets. In fact, at the cost of computational complex, the generation of such trajectory datasets can meet any volume of traffic and time granularity. Thus, the first two requirements above can be fulfilled.

The interaction of driver and driver, the driver and the road environment are considered in the currently generated vehicle-trajectory model. Moreover, most of the trajectories are generated by injecting macro data into the micromotion simulators. Couple with the availability of the real-world road map service, the third constraint can be easily met. In [18] detailed theoretical research of microscopic mobility modeling is proposed, and recommended readers for reference. The vehicle-trajectory dataset we introduce in the next sections satisfies the above constraints and is a new method to generate vehicle-trajectory data employed micromotion simulators.

III. Establish Simulation Model

The current mobility-model based vehicle-trajectory generation cannot meet the simulation needs of the vehicular network and vehicular networks. More trajectories of private vehicles are required as they represent the real movements of urban vehicles. However, due to privacy and security reasons, the data of private vehicles are quite difficult to obtain. Therefore, the development of vehicular network is hindered by the lack of sufficiently trajectory dataset. In this paper, we propose a generation method of the vehicle-trajectory dataset based on road bayonet data and ELM.

As shown in Fig. 1, our generation method of vehicle-trajectory dataset mainly includes three parts, the analysis of bayonet traffic, the data generation of the Vissim simulation, and the learning process of the parameters for the simulator. We adopt the bayonet data of several lanes near Jingtian Road in Shenzhen, China. By analyzing the data, the traffic flow is obtained of these roads. It is clear that the closer average traffic volume generated by the simulation is to the real one, the more accurately the generated vehicle-trajectory dataset can reflect the real traffic in the area. In Subsection B, a large amount of experimental data is generated according to different parameters. The generated experimental data is trained by applying the neural network learning algorithm, ELM to determine the weight of each parameter based on the experimental results. In this way, we set the parameters of Vissim simulation based on the weight matrix of the parameter to get the dataset of vehicle-trajectory which mirrors reality.

A. Establish Traffic Model

The road bayonet data of Shenzhen is considered, where we analyze the vehicle traffic statistics for several lanes, which contains all the vehicles information passing through the lane in the time range. The Shenzhen Ministry of Transport through carries on an automatic and systematical counting of the traffic flow through a set of counting devices on main road and motorway. As of 2018, traffic is counted in both directions at 5 different roads. The traffic volume changes with time also can be reflected by the bayonet data. The license plate number, time of passing the bayonet, and driving lane number are included in it. To eliminate data anomalies due to the operational errors, the data storage anomalies, and the missing data (duplicate data, error data, and incomplete data), we pre-process the data.

Then we analyze the statistics on these data. The number

Fig. 2 A road topology

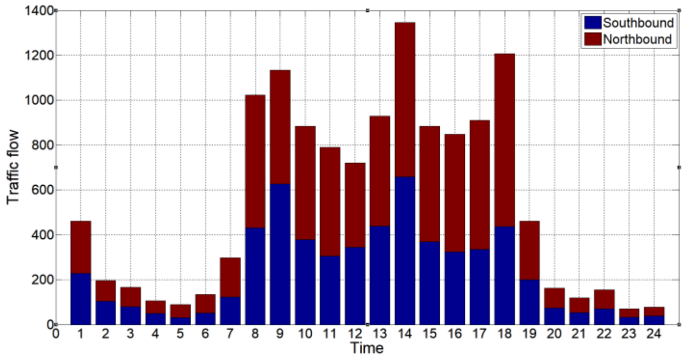

Fig. 3 Traffic flow of Jingtian Road one day

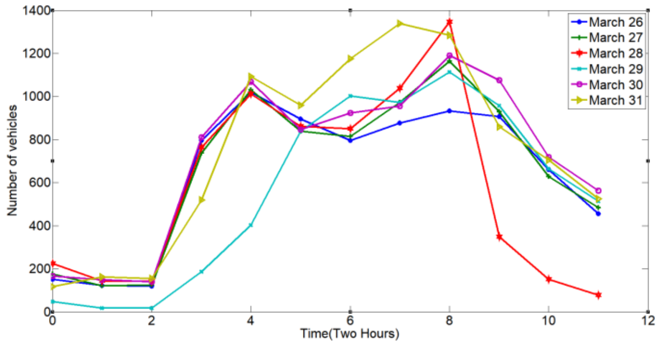

Fig. 4 Traffic flow of Jingtian Road for six days

of vehicles driving on the road at a different time is also distinct, which is important to the vehicle-trajectory dataset. For example, the traffic flow in the morning is the least, while around 4 pm is the most intensive traffic time. We divide a day into 12 timeslots, which means every two hours is a timeslot. We count the number of vehicles passing through the bayonet at each timeslot. Fig. 2 shows a traffic flow chart of Jingtian Road for six days. We observe that during these days, the traffic flow curve of Jingtian Road tends to be the same. According to the results of the bayonet data, it is obvious that the traffic flow of different levels of roads varies greatly. However, at each timeslot, the change in traffic volume still follows the rules above. Statistical information on the traffic flow of Southbound and Northbound Jingtian Road is shown in Fig. 3. The traffic flow of different directions of Jingtian Road tends to be the same. The traffic flow at different timeslots of these roads will be employed to construct the real-world traffic model.

After defining the traffic flow model by analyzing the bayonet data, we now describe how Origin and destination are chosen. From the perspective of macro traffic, urban functional areas can be referred to regions containing

TABLE I. FORMAT OF PARAMETER

Number of attributes	parameter name	Notes
1	number of effective lanes	INT
2	input flow	INT
3	input time	INT (0, 300)
4	input ratio	INT (0,1)
5	average traffic flow	FLOAT

surrounding buildings with a special function. Also, urban functional areas are closely related to human mobility patterns, i.e. when people arrive or leave a place, and where they come from, and where they leave for. The area general can be divided into three functional areas, such as industrial, commercial, or residential, according to different functions and feature. As mentioned above, geographical zones' surface and function type are obtained from OSM data. Through the analysis and modeling of urban mobile data, the probability for choosing an origin or destination is effected by three parameters, including the weight of its region function type, the weight of its attractive area, the weight of its prosperity.

Next, we mainly consider choosing the appropriate place to be the origin of the vehicle. The probability for choosing a region as origin of vehicles is greatly influenced by the surface and type of the geographical region. First, the attractiveness of each region function type can be set to the default value. Additionally, the surface of each zone is also computed as it will be a parameter of its attractiveness. However, the functional type and surface of a region cannot fully determine its attractiveness in the topology area. For instance, two industrial regions of equal functional type and surface may not be equally attractive if one is located on a metropolis center and another is located on a region with low population density. Therefore, we define an extra weight that applies to the attractiveness of different region, which we define as prosperity weight. Prosperity place represents more intensive vehicle travel in this place. The attractiveness of an area within the simulation topology is determined by these three parameters, and the attractive area will become the origin of the simulated vehicle travel. The urban functional areas we divided are shown in Fig. 4. Red represents residential areas, green represents commercial areas, and purple represents industrial areas. Functional area information will be used as the basis for Vissim to generate vehicle trajectory data. The next section will explain how to generate Vissim simulation data.

B. Generate Vissim Simulation Data

Vissim is a microscopic, time-interval and driving behavior-based simulation tool for the traffic modeling of urban traffic and public transportation operations. It can generate visual traffic conditions online, or output various statistics such as travel time and queue length. As a microscopic traffic simulation model, Vissim includes a car-following model [20] and a lane change model.

The accuracy of the traffic simulation model mainly relies on the quality of the traffic flow model. Unlike other low complex models which apply the continuous speed and a car-following model, the car-following model used by Vissim is a psycho-physiological driving behavior model. The basic idea of the Vissim is, once the rear driver finds the distance between his/her vehicle and the preceding vehicle is less than his/her psychological (safe) distance, the

Fig. 5 Function areas

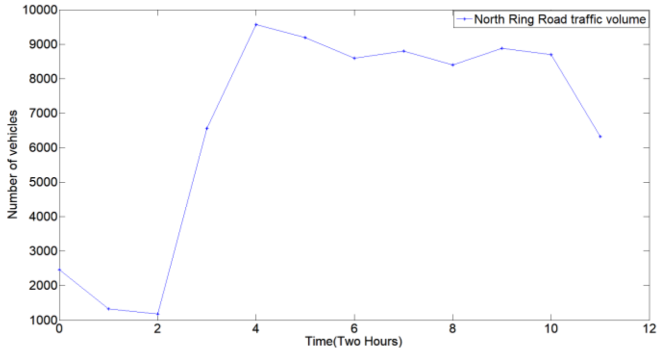

Fig. 6 North Ping Road traffic model

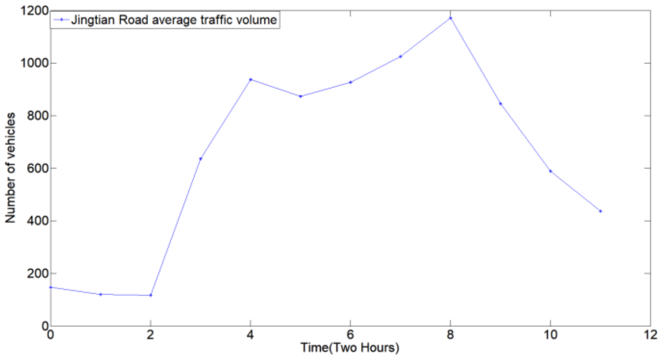

Fig. 7 Jingtian Road traffic model

rear driver then begins to slow down. Since the driver of the rear vehicle cannot accurately determine the speed of the preceding vehicle, the speed of the rear vehicle is then lower than the speed of the preceding vehicle for a timeslot. When the distance between the front and rear vehicles reaches another psychological (safe) distance, the driver of the rear from that area is added to the Vissim workspace. After vehicle begins to accelerate slowly. From this cycle, an iterative process of acceleration and deceleration is formed.

Next, we would like to present how Vissim generates a vehicle-trajectory dataset. Firstly, we construct a road topology. To make our road topology closer to the reality, the relevant areas from the real map are considered to construct the road topology in the simulation. After determining the relevant simulating area, the road topology setting up the corresponding scale, the road segment is constructed according to the road topology. We correct the road topology so that it can match to the real world. Different simulation parameters are set to generate a large amount of simulation data for subsequent ELM training. Fig. 5 is a road topology we constructed. The format of these parameters is shown in Table I. Input flow is the number of vehicles generated from the road per hour. The number of lanes with input flow divided by the total number of lanes is the input ratio. Average traffic flow is the average of the other lane traffic except for the input lane. We use average traffic flow as a performance metric of simulation results. The degree of similarity between average traffic flow and the bayonet traffic flow represents the fidelity of the simulation. To determine the weight of each parameter, we train many simulation data to get the combined effect of these parameters on the experimental results.

In addition to the parameters mentioned above, we also should consider how to select the origin of a trip of vehicle. By analyzing the data of the real-world map, we divide the simulation area into different functional areas. As mentioned above, we manually set the attraction value of each functional area, and most vehicular travels often occur near the attractive functional area. Hence, when setting up the origin of the vehicle in Vissim, those attractive functional areas are considered.

C. Determining Parameter Weights

The traditional single-hidden-layer feed forward neural network consists of an input layer (IL), a hidden layer (HL) and an output layer (OL) [21]. The IL and the HL, the HL and the OL neurons are fully connected. Among them, the input layer has n neurons, corresponding to n input variables; the hidden layer has 1 neuron; the output layer has m neurons, corresponding to m output variables. The connection weight between the input layer and the hidden layer is W, while the threshold of the hidden layer neurons is b. To achieve better learning performance, this learning algorithm requires many iterative learning steps. This consumes high computation and results in low efficiency. On this basis, ELM is a new algorithm for a single hidden layer feedforward neural network (SLFN). The traditional feedforward neural network has the disadvantages of slow training speed, easy to fall into local minimum and is sensitive to the selection of learning rate. In contrast, the ELM algorithm randomly generates the input layer connection weight and the hidden layer connection weight and the thresholds of the hidden layer neurons. Moreover, there is no tuning required during the training process in which we only need to set the number of neurons of the hidden layer. Hence, a unique optimal solution can be obtained. Compared with the traditional training methods, ELM has the advantages of faster learning speed and better generalization performance. In particularly, ELM mainly has the following steps.

- It first determines the number of neurons in the hidden layer, and randomly sets the connection weight w of the input layer and the hidden layer and the threshold b of the hidden layer neuron.
- It then elects an infinitely differentiable function as the activation function of hidden layer neurons and then calculating the hidden layer output matrix H.
- Finally, it calculates the weight of output layer β.

It is worth mentioning that the related research results [22] [23] show that many nonlinear activation functions can be used in ELM (such as sigmoid function, sine function, and compound function) as well as the non-differentiable functions.

Next, we can interpret the reason for using ELM to construct a weight matrix of parameter as in follow. The volume of real-time vehicle traffic on each road changes over time. Therefore, we cannot use only trajectory data with fixed-density to describe the state of traffic flow at all time of the day. For example, if we test protocols without the mobile trajectory data of various densities, it may lead to overly optimistic results. Testing routing protocols may have different results when we simulate the protocol over the vehicle-trajectory data of different vehicle sparseness. Therefore, we should generate different trajectories of vehicle sparsity over different timeslots to represent more comprehensive regional traffic data. However, Vissim cannot directly generate vehicle-trajectory with a specified density. The Vissim simulation parameters determine the type of generated vehicle-trajectory data. Hence, we need to get the relationship between Vissim parameters and simulation results to avoid duplication of work. The generation method of vehicle-trajectory proposed herein can generate the movement trajectories of different vehicle densities for the different timeslots. This method requires accurate bayonet data to perform pre-modeling which determines the traffic density of the road for each timeslot. After analyzing the characteristics of Vissim, we identify several parameters that have great influences on simulation results. We apply ELM to determine the parameter weight matrix, which is employed to study how these parameters affect the simulation results. Then, the relevant parameters are set in the Vissim by using the generated weight matrix of parameter, while the trajectory data corresponding to the vehicle density is obtained. The trained weight matrix could be applied to predict the vehicle density and be used to generate the vehicle-trajectory dataset.

IV. Experiment

This section mainly includes three parts: 1) processing bayonet data, 2) determining the weight matrix of the parameter, and 3) comparing and analyzing experimental results. Firstly, we process the bayonet data to build a real traffic model. Then, a large amount of Vissim simulation data is used to determine the weight matrix of the parameter. This weight matrix of the parameter can be used to predict Vissim simulation results and build Vissim simulation models. Finally, we generate vehicle-trajectory data based on the real traffic model and the weight matrix of the parameter. We upload most of the relevant code for building this model to the GitHub open source website, https://github.com/liuyufei1119/Traffic-mobile-data.git. The code is mainly divided into two categories, one is the code we use to process the bayonet data, and the other is the code that runs in Matlab [24]. The second type of code includes code that processes the training dataset and native ELM code.

A. Process Bayonet Dataset

The vehicle bayonet data we used is the traffic monitoring data that is collected from local roads in Shenzhen such as Fulong Road, Xinzhou Road and some main roads of Beihuan Road. The equipment of data acquisition stores all the information of the vehicle through the bayonet during this time, including the license plate number, the elapsed time, the monitoring point name and the lane number. However, a large amount of incorrect or malformed data could be collected where sometimes data loss may occur in case of fault collection devices. The dirty data could have a huge impact on data processing results. Therefore, before analyzing the data, we should clean the original data, remove the dirty data and keep the correct data for building real traffic model. The data for each monitoring point is stored in a separate file, and we classify the data

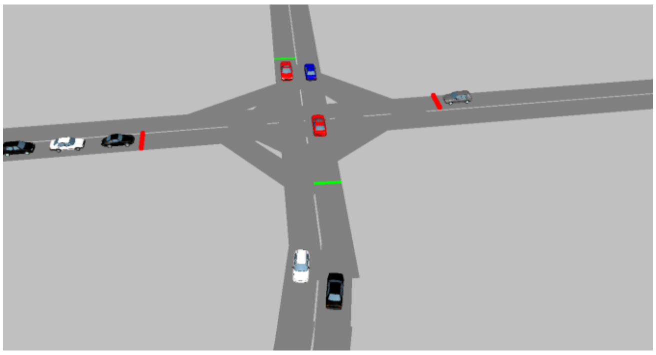

Fig. 8 Simulation screenshot

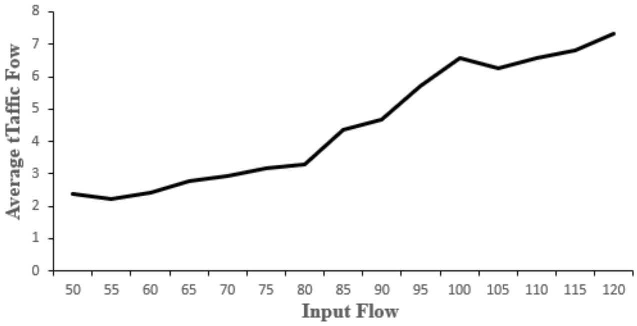

Fig. 9 The effect of input flow on the results

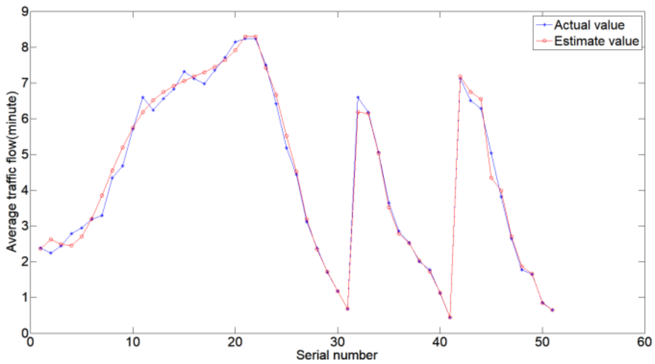

Fig. 10 Comparison of test results

TABLE II. THE EFFECT OF THE NUMBER OF HIDDEN LAYER NEURONS ON THE EXPERIMENT

Number of hidden layer neurons	Train deviation	Test deviation
20	0.0749	0.0754
200	0.0345	0.0374
500	0.0275	0.0288
1000	0.0269	0.0276
1500	0.0275	0.0284
2000	0.0274	0.0285
2500	0.0274	0.0281
3000	0.0276	0.0301
3500	0.0276	0.0306
4000	0.0276	0.0313

based on the time when the vehicle passes the monitoring point. For the sake of observation, the statistics of the number of vehicles in each road are calculated while two hours are counted as a timeslot. For some unidirectional multi-lane roads, we again classify the vehicle data in the timeslot according to the lane number, which gives the traffic volume data for each lane per timeslot. Due to the

TABLE III. SIMULATED VEHICLE FLOW MODEL

Number of attributes	Input time	Predicted average traffic flow
1	290	8.10
2	280	7.86
3	270	7.36
4	260	6.81
5	250	6.35
6	240	5.98
7	230	5.63
8	220	5.24
9	210	4.80
10	200	4.41
11	190	4.16
12	180	4.08
13	170	4.15
14	160	4.23
15	150	4.15
16	140	3.78
17	130	3.14
18	120	2.40
19	110	1.83
20	100	1.59

breakdown of the detection equipment, the erroneous vehicle data of March 25 is too large to be used. Thereby, we deploy the average traffic data for the remaining six days as the real traffic model of the road. Fig. 6 shows the six-day real traffic model of Jingtian Road. It can be seen from this figure that the traffic volume of the road changes relatively, with the maximum traffic flow, starts from 16:00 to 17:00 and the smallest from 4:00 to 5:00. This traffic flow curve serves as a model to generate vehicle-trajectories using Vissim. The relationship of traffic volume over time will be used as a real-world traffic flow model and will be one of the evaluation criteria for generating vehicle-trajectory datasets.

We have a total of 9 sets of data, one of which is abnormal due to damage to the test equipment. Also, five of the remaining eight sets of data belong to highways or national highways, and the traffic volume of these roads is highly dense. Fig. 7 shows the traffic flow of the North Ring Road. There are also three groups of data for ordinary city lanes. Compared with highways, vehicles in ordinary city lanes are slower and have a lower traffic volume. In this method, urban area simulation is mainly considered, so only these three sets of data are used as the real traffic model. Fig. 8 is a traffic diagram of an intersection during the simulation. In addition to the parameters mentioned above that have a greater impact on the experimental results, some parameters have less impact, such as the composition of the traffic flow. For these parameters, we generally use the default value of Vissim.

B. Determine Weight Matrix of Parameters

After getting the real traffic model, this subsection describes how to generate the weight matrix of Vissim parameter. Firstly, we generate the initial experimental data by applying Vissim. Different Vissim parameters are set to generate the experimental data, which can develop a diverse set of experimental data. We select two regions on Google

Map as road topology maps for generating experimental data. Variable-control methods are used to generate raw data, which makes the data intuitive and clear. To generate the continuous experimental data, only one parameter was changed per experiment compared to the previous experiment. This continuous data helps us to make statistics, and it also visually shows the impact of this parameter on the experimental results. Fig. 9 is experimental data obtained by continuously changing the input flow while other parameters are unchanged. It can be seen that the average traffic volume increased as the input traffic increased. The input flow is linear with the average traffic flow when other parameters are constant within a certain range. In this way, we generate a lot of experimental data.

The weight matrix of the parameter is trained by ELM below, and its execution environment is Matlab2014a. The activation function of ELM we choose the "Sigmoid" function. All input attributes must be normalized to [-1,1] before training the data with ELM. We also apply the number of effective lanes, input flow, input time, and input ratio as input data, and the average traffic volume as output data. For the number of hidden neurons in ELM, we consider the optimal value from the incremental experiments. The initial number of hidden neurons is set as 20. Finally, the best results are summarized in Table II. When the number of hidden layer nodes is 1000, the training result of ELM is optimal.

After the training of ELM, a parameter weight matrix is learned. Fig. 10 is the comparison of test results. ELM training deviation is 0.0269 where test deviation is set as 0.0276. The weight matrix of the parameter can be used to predict the simulated vehicle flow of Vissim, or the value of each parameter according to this matrix can be derived from the vehicle density for the different timeslots. In this way, the corresponding vehicle-trajectory can be generated according to the real traffic model of different regions at different times.

C. *Experiment Comparison and Analysis*

According to the generated weight matrix of the parameter, we can predict the Vissim simulation results. Comparing the predicted results with the Vissim simulated results to analyze the effectiveness of the weight matrix of the parameter. We acquired a suitable area of Beijing on Google Maps to build a Vissim road topology. After constructing the road topology, the values of the Vissim simulation parameters are also required for simulation. We get the traffic density of the timeslot we want to simulate by real traffic model. Next, we will talk about how to use Vissim to generate a vehicle-trajectory dataset close to real traffic model. It is necessary to set the appropriate parameters for Vissim to obtain the vehicle-trajectory data corresponding to different vehicle density. The appropriate origin of vehicle travel needs to be considered before setting these simulation parameters. As shown in Fig. 3, we select the origin of the vehicle based on the divided functional areas. We give priority to those attractive functional areas as the origin of vehicle travel. According to the size of the simulation area, there are ten points in the simulation as the origin of vehicle travel. Hence, the road next to the ten most attractive functional areas is selected as the origin. After setting the origin of vehicle travel, we consider setting Vissim simulation parameters to carry out simulation experiments.

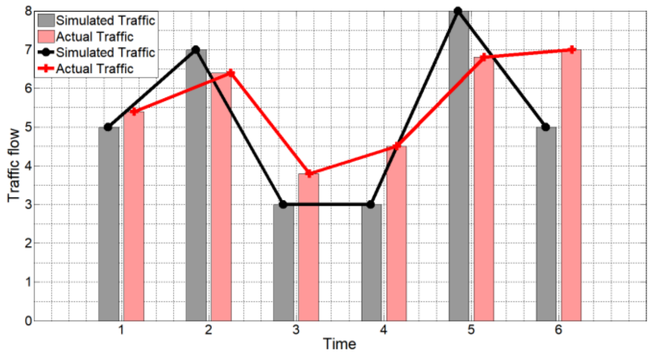

Fig. 11 Average vehicle flow contrast

TABLE IV. STATISTICAL RESULTS OF MEASURE POINTS

Measure	from	to	Number of Vehicle
1	300	360	5
2	300	360	7
3	300	360	3
4	300	360	3
5	300	360	8
6	300	360	5

TABLE V. JINGTIAN ROAD DATA COMPARISON

Real traffic flow	Predicted traffic flow	Simulated vehicle flow
1.23	1.59	1.58
3.6	3.78	4.02
4.9	4.80	4.76
5.3	5.24	5.12
7.3	7.36	7.52
7.8	7.86	7.76

Now we can determine these simulation parameters by applying our constructed weight matrix of parameters. Firstly, a simulated vehicle flow model needs to be modeled. Here we focus on one of the models, that is, the simulation temporal model. The experimental data employed to build this simulation model are generated by a series of experiments, with the following parameters. The number of effective lanes is set as 10. The input ratio is the ratio of the number of effective lanes to the total number of lanes, and its value is 0.233. The initial input time is set to 300, while the initial input flow is set to 150, reduced the input time from 300 to 50 with the other parameters unchanged. Then the weight matrix of parameters is predicted to this set of parameters, as shown in Table III, this is part of the simulated vehicle flow model we generated. To compare the experimental results with the real traffic conditions, the traffic flow counter is placed on three main roads in the road topology. Two counters are placed on each road to count the traffic flow in two directions and record the traffic flow lasting one minute from 300 seconds to 360 seconds after the start of the simulation. In Table IV, The counter mainly records the number of vehicles passing through the monitoring points in this minute. We compare the vehicle flow data recorded by the counters with the real vehicle flow data. As shown in Fig. 11, counter number 1 and 2 represent traffic flow data collected by counters in different directions of a lane, and so on. Vehicle-trajectory data generated by simulation show roughly the same volume of traffic at monitoring points as the real data. The average traffic flow of these six counters is 5.17, and the average traffic flow of the card data is 5.65. This represents that the vehicle trajectory data we generate can truly reflect the real-world traffic situation in the region.

According to Fig. 2, we randomly select the traffic-flow density corresponding to the six timeslots of Jingtian Road. The traffic flow density is the average number of vehicles passing through the section within one minute. Table V is the result of our experiment comparison. If we reduce the amount of change in the input time in the simulated vehicle flow model, we can get better experimental results.

Using our method, we can easily set the parameters of Vissim to get the movement trajectory datasets of a different time or different density. After many experiments, we found that the traffic flow will be unevenly distributed at the beginning of the simulation. Because Vissim only generates traffic at a fixed location, the vehicle will only move at the birth point if the simulation time is too short. The vehicle-trajectory data generated by the simulation about 300s can have good balance. In general, this method can generate movement trajectory datasets with different vehicle densities according to different timeslot, and can also predict simulation results according to Vissim parameters, so this method has strong practicability.

V. Conclusion

The lack of vehicle-trajectory dataset hinders research in vehicular networks. To address this problem, we propose a method for generating vehicle-trajectory dataset using Vissim based on bayonet data. Comparing with the data of the clamp port, it is proved that the vehicle trajectory data generated by our method is authentic. An important reason for the authenticity of the experimental data is that the impact of urban functional areas on vehicle travel is taken into account when we set the origin point of vehicles. Our methods mainly include establishing real traffic model, generating Vissim simulation data and generating weight matrix of parameter. It can be employed to construct simulated vehicle flow model and generate vehicle-trajectory datasets, demonstrating the effectiveness of our approach by comparing it to real Vissim experimental data. In our approach, the vehicle uses the default method of the simulation tool to select the driving route, which may result in a vehicle with a purposeless repeating trajectory. In the future, we will consider adding weights to each road to let the vehicle choose its route, and we will consider using more parameters to measure the effectiveness of the generated vehicle-trajectory dataset. For example, we will add the congestion level of each lane to the standard for measuring the simulation results, which will make our dataset generation method better. Also, the traffic flow model of each road requires more accurate measurement data to the model, and it is not currently available. However, in the future, we intend to acquire and use such data.

Acknowledgment

This work is partly supported by the National Science Foundation for Young Scientists of China (61701322), the Key Projects of Liaoning Natural Science Foundation (20170540700) and the Liaoning Provincial Department of Education Science Foundation (L201630).

References

[1] A. Sharma, R. Chaki and U. Bhattacharya, "Applications of wireless sensor network in Intelligent Traffic System: A review," *2011 3rd International Conference on Electronics Computer Technology*, Kanyakumari, 2011, pp. 53-57.

[2] D. Kombate and Wanglina, "The Internet of Vehicles Based on 5G Communications," *2016 IEEE International Conference on Internet of Things (iThings)*, Chengdu, 2016, pp. 445-448.

[3] F. Bai, N. Sadagopan, A. Helmy, "The IMPORTANT framework for analyzing the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks", Elsevier Ad Hoc Networks, vol.1, 2003.

[4] M. Fiore, J. Härri, "The Networking Shape of Vehicular Mobility", ACM MobiHoc, Hong Kong, PRC, May2008.

[5] W. Viriyasitavat, F. Bai, O.K. Tonguz, "Dynamics of Network Connectivity in Urban Vehicular Networks", IEEE Journal on Selected Areas in Communications, vol.29, no.3, pp.515–533, 2011.

[6] A. K. Saha, D. B. Johnson, "Modeling Mobility for Vehicular Ad Hoc Networks", ACM VANET, Philadelphia, PA, USA, Oct.2004.

[7] S. Jaap, M. Bechler, L. Wolf, "Evaluation of Routing Protocols for Vehicular Ad Hoc Networks in City Traffic Scenarios", IEEE ITSC, Vienna, Austria, Sep.2005.

[8] W. Li, Q. Chen, T. Dong, L. Wei and Q. Zhang, "Traffic Signs Classification Based on Local Characteristics and ELM," *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, 2017, pp. 127-130.

[9] G. Huang, Q. Zhu and C. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, Budapest, 2004, pp. 985-990.

[10] L. Codeca, R. Frank and T. Engel, "Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research," *2015 IEEE Vehicular Networking Conference (VNC)*, Kyoto, 2015, pp. 1-8.

[11] R. Ketabi, B. Alipour and A. Helmy, "Poster abstract: En route towards trace-based simulation of vehicular mobility," *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Atlanta, GA, 2017, pp. 1020-1021.

[12] Y. Pigné, G. Danoy, and P. Bouvry, "A vehicular mobility model based on real traffic counting data," in *Communication Technologies for Vehicles. Springer,* 2011, pp. 131–142.

[13] X. Kong *et al*., "Mobility Dataset Generation for Vehicular Social Networks Based on Floating Car Data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3874-3886, May 2018.

[14] S. Haddouch, H. Hachimi and N. Hmina, "Modeling the flow of road traffic with the SUMO simulator," *2018 4th International Conference on Optimization and Applications (ICOA)*, Mohammedia, 2018, pp. 1-5.

[15] S. Uppoor, O. Trullols-Cruces, M. Fiore and J. M. Barcelo-Ordinas, "Generation and Analysis of a Large-Scale Urban Vehicular Mobility Dataset," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1061-1075, May 2014.

[16] S. Qiao, D. Shen, X. Wang, N. Han and W. Zhu, "A Self-Adaptive Parameter Selection Trajectory Prediction Approach via Hidden Markov Models," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 284-296, Feb. 2015.

[17] Q. Peng, Z. Ding, and L. Guo, "Prediction of trajectory based on Markov chains," Compu. Sci., vol. 37, no. 8, pp. 189–193, 2010.

[18] M. Fiore, "Vehicular Mobility Models", in S. Olariu, M. Weigle (editors), Vehicular Networks: From Theory to Practice, Chapman & Hall/CRC, 2009.

[19] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in Proceedings of the18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012, pp. 186–194.

[20] J. Leng, Y. Zhang and M. Sun, "VISSIM-Based Simulation Approach to Evaluation of Design and Operational Performance of U-turn at Intersection in China," *2008 International Workshop on Modelling, Simulation and Optimization*, Hong Kong, 2008, pp. 309-312.

[21] H. Nguyen, L. Kieu, T. Wen and C. Cai, "Deep learning methods in transportation domain: a review," *IET Intelligent Transport Systems*, vol. 12, no. 9, pp. 998-1004, 2018.

[22] G. Huang, G. B. Huang, S. Song and K. You, "Trends in extreme learning machines: a review", *Neural Networks, 61*(C), 2015, pp. 32-48.

[23] M. R. Daliri, "A Hybrid Automatic System for the Diagnosis of Lung Cancer Based on Genetic Algorithm and Fuzzy Extreme Learning Machines," *Journal of Medical Systems,* 2012, 36 (2), pp. 1001-1005.

[24] N. Li and Y. Sun, "VISSIM Parameter Calibration Based on Traffic Characteristics Distribution at Signalized Intersections," *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, Enshi, China, 2018, pp. 150-153.