Toward Predicting Stay Time for Private Car Users: A RNN-NALU Approach

Qibo Zhang, Fanzi Zeng, *Member, IEEE*, Zhu Xiao, *Senior Member, IEEE*, Hongbo Jiang, *Senior Member, IEEE*, Amelia C. Regan, *Member, IEEE*, Kehua Yang, and Yongdong Zhu

***Abstract*—Predicting the stay time of private cars has various applications in location-based services and traffic management. Due to the associated randomness and uncertainty, achieving the promising performance of stay time prediction is a challenge. We propose an RNN-based encoder model to solve this problem, which consists of three components, i.e., an encoder module, an exception module, and an MLP dropout. First, we encode the stay behaviour into hidden vectors at a specific time to avoid the effect of time sparsity. The encoder module utilizes a multilayer perceptron (MLP) to learn spatiotemporal features from the historical trajectory data, such as the inherent relationship between the stop points and corresponding stay time. We proved a linear relationship problem that cannot be ignored in the stay time prediction problem. In particular, we have added basic arithmetic logic units to the network framework to find linear relationships. By reconstructing the basic arithmetic and logical relations of the network, we have improved the ability of the neural network to handle linear relations and the extrapolation ability of the neural network. Our method can remember the number patterns seen in the training set very well and infer this representation reasonably. Moreover, we utilize the dropout technique to prevent the prediction model from overfitting. We perform extensive experiments based on a large-scale real-world private car trajectory dataset. The experimental results demonstrate that our method achieves an RMSE of 0.1429 and a MAPE of 55.8533%. Furthermore, the results verify the effectiveness and advantages of the proposed model when compared with the benchmarks.**

***Index Terms*—Stay event, private car, neural network, human mobility.**

Manuscript received July 13, 2021; revised November 12, 2021; accepted April 1, 2022. Date of publication April 5, 2022; date of current version June 24, 2022. This work was supported in part by the National Natural Science Foundation of China under Grants U20A20181 and 61772401, in part by the Humanities and Social Sciences Foundation of Ministry of Education under Grant 21YJCZH183, in part by the Key R&D Project of Hunan Province of China under Grant 2022GK2020, in part by the Hunan Natural Science Foundation of China under Grant 2022JJ2059, in part by the Funding Project of Zhejiang Lab under Grant 2021LC0AB05, and in part by the Open Project of Guangming Laboratory under Grant GML-KF-22-22. The review of this article was coordinated by Prof. Sukumar Kamalasadan. *(Corresponding authors: Fanzi Zeng; Zhu Xiao; Yongdong Zhu.)*

Qibo Zhang, Fanzi Zeng, Zhu Xiao, Hongbo Jiang, and Kehua Yang are with the College of Computer Science and Electronic Engineering, Hunan University, Hunan 410082, China (e-mail: zhangqibo@hnu.edu.cn; zengfanzi@hnu.edu.cn; zhxiao@hnu.edu.cn; khyang@hnu.edu.cn).

Amelia C. Regan is with the Department of Computer Science, Institute of Transportation Studies, University of California, Irvine, CA 92697 USA (e-mail: aregan@uci.edu).

Yongdong Zhu is with Zhejiang Lab, Hangzhou 311121, China (e-mail: zhuyd@zhejianglab.com).

Digital Object Identifier 10.1109/TVT.2022.3164978

I. Introduction

The *STAY time*, as the term itself implies, indicates the length of time that people spend when they arrive and stay at a determined destination [1]. Predicting the length of stay time can effectively improve the quality of various location-based services [2]. For example, stay time plays an important role in recommending online points of interest (PoIs) since it directly affects sales [3]. Service providers can improve the accuracy of their marketing policies through better understanding the stay time of drivers. As part of an intelligent transportation system (ITS), stay time provides valuable information for traffic condition prediction. By predicting when vehicles in a stay state will start, the transportation agency can estimate road congestion in advance.

In this paper, we strive to predict stay time with special consideration of private car users. One observation that motivates our work is that people driving private cars to fulfill their travel needs has become a major daily activity, especially against the background of the continuous development of modern industrialization and urbanization [4], [5]. Additionally, we purposely use private car trajectory data [6], [7], since we can extract stay events [2] and stay times from this dataset, while other trajectory datasets, such as those of taxis and check-ins, do not contain any information on stay events and stay time [8].

Recently, concerns on the human mobility of private car users have emerged [2], [4], [9]. In particular, the recent development of neural networks provides optional solutions for predicting stay time due to their powerful modeling and training ability. For instance, J. Manweiler *et al.* used a machine learning algorithm to predict stay time at WiFi hotspots [1]. They used sensors (such as accelerometers and compasses) to estimate how long people stay in WiFi hotspots, without considering the spatiotemporal representation. J. Chen *et al.* integrated a decision tree with a recurrent neural network (RNN) to predict the stay time of vehicles via extracting spatiotemporal features [10]. However, it is not an easy task to achieve promising performance in predicting stay time [11]. This view is mainly derived from the fact that stay time usually involves randomness and uncertainty since many uncontrollable factors, such as subjective reasons (i.e., the user's mood) and objective reasons (i.e., the weather), have explicit impacts on stay time.

Staying behavior is sparse, and the travel is divided into two parts: driving and staying, which cause the time sparseness of the staying behavior. A private car user generally stays only

TABLE I
PCC With Different Characteristic

	Stop Lon	Stop Lat	Start hour	Start minute	Start time
Area.1	0.267	-0.160	0.609	-0.011	0.780
Area.2	0.035	-0.031	0.067	-0.001	0.074
Area.3	0.080	-0.062	-0.030	0.087	0.314

several hundred times per year, which is very sparse relative to the entire spatiotemporal domain. First, relative to the entire two-dimensional plane (latitude and longitude), the stay location is sparse. Second, relative to the entire time domain, hundreds of personal stay behavior data points are sparse every year [10].

In addition to the uncertainty of stay time, the following challenges impact the prediction of stay time. *i*) In addition to the spatial connection between stay events [2], in the time domain, the arrival time of stay events and different time intervals of adjacent stay events are highly related to the variance in stay time. This leads to the effect of time sparsity, which brings about challenges for achieving efficient spatiotemporal feature representation. *ii*) Although stay time has obvious randomness, it shows some degree of a linear relationship. According to people's needs, various stay behaviors have different stay times. The stay time can be divided into two parts. One part is the time spent according to the purpose of stay, and the other part is the time spent from the place of stay to the final place.

In Table I, 'Stop Lon' represents longitude of stop point, 'Stop Lat' is the latitude of stop point, 'Start hour' represents the hour and hour when the stay event starts, 'Start minute' represents the minute and hour when the stay event starts, 'Start time' resents the sum of the hour and minute when the stay event starts. Table I shows PCCs in three regions. We found that PCCs in different regions have different characteristics. For example, the PCC of Start time in Area.1 is 0.780, which has a strong linear relationship, but the PCC in Area.2 is only 0.074, and the linear relationship is almost non-existent. According to the characteristics of the linear relationship reflected in different regions, we can think that linear relationships exist on the issue of stay time prediction, and the research and discussion of linear relationships are significant.

When a user arrives at a place, the time spent each time for a certain purpose is similar. On the one hand, we look at the linear relationship from the temporal aspect, and different arrival times will linearly affect the stay time. For example, a user usually arrives home from work at 7 o'clock. If he/she arrives home at 9 o'clock for certain things, then his/her staying time may be reduced by two hours. On the other hand, we consider the linear relationship from the spatial aspect. The additional time consumed by different parking areas will change, including the linear relationship. The staying location being far away from the final destination will increase the stay time. Deep neural networks bring about powerful nonlinear fitting capabilities. However, nonlinear activation functions, which act as the basic structure in deep neural networks, make the network insensitive to these linear changes.

To address the abovementioned challenges, we design an RNN-based encoder model to resolve the problem of predicting stay time. Our proposed model consists of three components, i.e., an encoder module, an exception module and an MLP dropout. Specifically, we only encode one stay event into hidden vectors at a time, which avoids the effect of time sparsity. The encoder module utilizes a multilayer perceptron (MLP) to learn spatiotemporal features from the historical trajectory data, such as the inherent relationship between the stay points and the corresponding stay time [12]. The output of the MLP will be combined to generate the final spatiotemporal features by gated recurrent (GRU) cells. To prevent the model from overfitting, we utilize the dropout technique of the original MLP. Moreover, we build an exception module with neural arithmetic logic units (NALUs) [13] in the prediction model. The NALU layer enhances the neural network's ability to handle linear relationships by reconstructing the basic arithmetic logic relationship, which enables the proposed prediction model to have better predictive power. Finally, we integrate the output of the exception module to predict the stay time by MLP dropout.

The main contributions of this paper are summarized as follows.

- In order to predict the stay time, we built an encoder model based on RNN to capture the relationship between the stay point of a private car and its stay time. By doing so, we reduce the impact of sparsity and randomness of staying events. In addition, we use dropout techniques to reduce the overfitting of the prediction model.
- We proved a linear relationship problem that cannot be ignored in the stay time prediction problem. Since the existing conventional neural networks cannot cope with the prosperous linear relationship, we have improved the neural network's ability to handle linear relationships by reconstructing the basic arithmetic and logical relationships of the network and enhancing the extrapolation ability of the neural network.
- We conducted extensive experiments using a real-world private car trajectory data set. Experimental results show that, compared with advanced prediction network benchmark tests, our prediction model has good prediction capabilities and strong learning capabilities.

The remainder of this paper is organized as follows. In Section II, we review the related work. Then, we present our dataset, the stay event detection and the spatiotemporal data analysis of private car users in Section III. In Section IV, we introduce our prediction model. In Section V, we present the results of our experiments and evaluations. Finally, we conclude the paper in Section VI.

II. Related Work

In this paper, we investigate how to predict the stay time of private car users. This is closely related to human mobility and stay time. In this section, we review the research status of both topics.

Due to the needs of urban planning, social management, traffic forecasting and other applications, human mobility has been extensively studied [14] [15]. With the development of communication technology, it is becoming increasingly more common to use human resources and intelligent terminals to study human mobility models.

Researchers have begun to use big data to predict traffic flow [16]. Spatiotemporal analysis methods have been proposed for mobile phone activity data and for studying the relationship between network traffic and traffic flow [17], [18]. S. Jiang *et al.* [19] proposed three types of methods for extracting information from triangular mobile phone signals, describing different applications in spatiotemporal analysis and city modeling, and inferring the purpose of travel by using the geographic semantics of the travel destination. Some researchers [20] [21] have established mobility models to characterize the spatiotemporal pattern of human mobility, hoping to reproduce some characteristics of human movement. C. Song *et al.* [22] used personal trajectories recorded by mobile phones to discover some characteristics of human travel and conducted a quantitative analysis of the statistical characteristics of each human trajectory. M. C. Gonzalez *et al.* [23] proposed that human travel behavior has a highly temporal and spatial pattern and that human travel behavior has certain similarities. Some researchers have divided the communication network into different areas, used the data in the communication network to model the urban traffic distribution, and finally estimated traffic flow in the area [24], [25]. Other researchers have collected a large amount of mobile terminal and GPS data to infer people's points of interest and travel purpose [26], [27]. Soares [28] used a smartphone to detect the real-time travel mode of the intelligent transportation system.

Detecting traffic patterns is a common way to learn about the stay event. Haosheng Huang *et al.* [29] reviewed the research content of using mobile phone network data to detect traffic modes and got the conclusion that traffic mode is the key to travel behaviour research. The smartphone is a low-cost and high-efficiency device in the intelligent transportation system. In order to detect the movement and static state of vehicles, H.R.Eftekhari and M.Ghatee [30] has developed a new inference engine based on inertial measurement units to detect motorized mode. Further, to monitor and evaluate driving behaviour, a new system based on the inertial unit of the smartphone was developed, which uses the latitude and longitude data of the acceleration sensor to identify the driver's behaviour [31]. D. T *et al.* [32] provide a methodological framework for the comparative evaluation of driving safety efficiency based on Data Envelopment Analysis, they combined smartphone data with vehicle data to study vehicle driver efficiency. The vast majority of network traffic in the world comes from smartphones. Stratis Kanarachos *et al.* [33] has verified the role of smartphones as an integrated platform to monitor driver behaviour. Yu Cui *et al.* [34] uses smartphone GPS data to develop a comprehensive daily activity location scheduling model to capture known and unknown activities and build traffic simulators by modelling activities of different levels. O.Burkhard *et al.* [35] combines passive tracking of telephone providers and historical location data to develop a method of classification of transportation mode.

Understanding the time of the transportation system helps people choose the right road and reduces transportation costs and traffic uncertainty, and thus, time in intelligent transportation systems has been widely studied. Some scholars have focused on vehicle travel time. W.-H *et al.* [36] proposed a travel time prediction model, which predicted the high-speed travel time of each vehicle when the vehicles on the highway interfered with each other.

Researchers have become concerned with stay time. Some researchers have explored the effects of spatial differences and temporal changes on temporal patterns by using mobile phone data [37], [38]. J. Manweiler *et al.* [1] and S. Liu *et al.* [39] studied the stay time of mobile users. R. Low *et al.* [40] studied the parking activities of heavy trucks and used a generative adversarial network to predict the parking duration of commercial vehicles [41] K. S. Kung *et al.* [42] find that the maximum length of stay of city residents when commuting is related to some important daily human activities. Although the stay time is random and uncertain, in 2010, researchers published a study on the predictability of human movement in science. By measuring the entropy of the trajectory of anonymous users, it was found that users have 93% potential predictive power [43]. Y. Li [11] studied the predictability of the stay time of vehicles in different areas. Compared with human travel modes, modes of transportation have completely different characteristics. J. Chen *et al.* [10] used clustering and kernel density estimation to extract the spatiotemporal characteristics of stay events. Based on a deep neural network, a stay time predictor (STP) model was constructed to predict the stay time of stay events.

Our work validates the new idea of linear relationships in deep learning related to recent innovations in deep learning architectures. Many popular neural network architectures [44] [45] also advocate using linear links to reduce exploding/vanishing gradients or check the relationship between nodes. In connection, the linear relationship thinking in our article is also in line with a broader topic in machine learning, which attempts to identify the system's underlying structure in the form of behavioural control equations, which can reasonably infer the invisible part of the space. This is also a strong trend in recurrent networks, allowing the network to infer longer sequences than in training.Recent work [46], [47] attempts to use sorting to enhance LSTM, and they focus on using external memory modules to improve generalization capabilities to find sequences outside of system training.

In conclusion, most existing studies have qualitatively analyzed the influence of spatiotemporal factors on the spatial patterns of human mobility prediction. In this study, we focus on the temporal patterns within human mobility patterns. Via leveraging deep learning techniques, we aim to model the stay time pattern and combine historical spatiotemporal data to quantitatively predict the stay time using private car trajectory data.

III. Preliminary

In order to conveniently describe the concepts of stay event detection and prediction, this paper defines some critical definitions.:

Definition 1: stay event: a stay event refers to a situation where the vehicle stays somewhere temporarily, does not continue to move forward, and is in a stopped state.

TABLE II
TRAJECTORY TRIPS IN THE DATASET

Record ID	User ID	Start Time	Stop Time	Start Lon	Start Lat	Stop Lon	Stop Lat
23127	402500	2016-03-01 09:22:54	2016-03-01 09:24:00	114.093705	22.540422	114.093862	22.540712
318768	402500	2017-06-02 14:54:06	2017-06-02 14:58:55	114.054053	22.571113	114.064652	22.572648
852972	402500	2017-10-05 06:52:43	2017-10-05 06:58:15	114.076328	22.534818	114.060030	22.533915

Algorithm 1: Stay Event Detection from Trajectory Trips.

Input: S_{TI} retrieved from the private car trajectory dataset
Output: The record of the stay statue
function STAY EVENT DETECTION
  Extract S_{TI} from the dataset by User ID
  Sort S_{TI} by Record ID
  while $dist(S^{i}_{TI}, S^{i-1}_{TI}) \leqslant 300\,m$ **do**
    while $The\ duration \geqslant 120\,s$ **do**
      Record S^{i}_{TI} in stay statue of the User ID
    end while
  end while
end function

Definition 2: stay time: the vehicle changes from a moving state to a stationary state and then from a stationary state to a moving state. The duration of the static state is the stay time.

A. Trajectory Data and Stay Event Detection

Through our previous works [48], [49], we have obtained the trajectory dataset of large-scale real-world private cars from urban scenarios [4], [6]. When installing GPS/OBD devices in private cars, the collection of private car trajectory data was explained to the volunteers, and their consent was obtained. At the same time, when uploading data, the owner's real vehicle ID is anonymous, and the International Mobile Equipment Identity (IMEI) number is assigned to the GPS/OBD device as the unique ID of each vehicle for the purpose of privacy protection. In this dataset, the collected trajectory information is expressed in the form of a single trip. Each trip contains the record ID of the trip, vehicle ID, start and end times, start and end positions, mileage, etc. In addition, the driving status of each trip is recorded (such as speed, steering and current alarm information). For instance, the travel information (TI) contained in trajectory trips is presented in Table II.

Based on the trajectory dataset, we propose an algorithm to detect stay events from TI in the trajectory trips. As shown in Algorithm 1, the algorithm consists of the following steps. First, we extract the travel information of the same User ID in the dataset as S_{TI} = (Record ID, User ID, Start Time, Stop Time, Start Lon., Start Lat., Stop Lon., Stop Lat.) and sort them according to the Record ID. There are two reasons why the data are abnormal. The first reason is that the stay time being below a certain standard may be a special situation, such as flameout due to abnormal conditions. The second reason is the error of GPS. Our purpose of setting two values is to clean up any abnormal data in the dataset. Second, when we obtain trip information, we calculate the starting point of the current trip and the stopping distance of the previous trip. If the distance between the two trips does not exceed 300 m, then it is judged that the position information is not missing. Finally, we calculate the stop time of this trip and the start time of the next trip. If the stay time exceeds 120 s, then it is judged that the stay event of this trip is effective, and the current stay time is recorded. Earth's surface distance can be calculate by the Haversine formula as follows [50]:

$$dist(A, B) = 2 \times r \times \arcsin \sqrt{a} \tag{1}$$

$$duration = t_i - t_j \tag{2}$$

where

$$a = \sin^2 b + \cos(lat_i) * \cos(lon_j) * \sin^2(c)$$

$$b = \frac{lat_i - lat_j}{2}$$

$$c = \frac{lon_i - lon_j}{2}$$

where r is the radius of the earth, dist(A,B) is the distance between two S_{TI}:A(Lon_i, Lat_i, t_i), B (Lon_j, Lat_j, and t_j).

B. Spatiotemporal Analysis of Stay Time

After retrieving the stay events, we conduct a spatiotemporal analysis of stay time to study the connection between stay time, arrival time and stop points.

Usually, private cars have thousands of stay event records and thereby generate many stop points in their trajectories within a long period of time such as one year. Inspired from the findings in [4], we observe that most of the stop points are concentrated in several fixed areas.

Overall, the Fig. 1 shows that most stop points occur within a specific range of arrival times. Each private car owner's travel mode has its characteristics. This uniqueness is reflected in two aspects. The first is the uniqueness of parking time. As shown in Fig. 1, some users have a multimodal distribution in parking time, and some have a unimodal distribution. The second is the uniqueness of the number of stay events. The four users in our sample recorded the most parking times in three years, which were 7,066, 5,264, 4,013 and 3,969. Most users record less than a thousand times. To show the uniqueness, we selected the four users with the most records. Fig. 1(a) shows that the stay events usually occurs between 5 to 9 o'clock and 12 to 15 o'clock. In the stay event from 5 to 9 o'clock, the stay time tends to be concentrated in 10 minutes to 20 minutes. In the period from 12 to 15 o'clock, the stay time is more evenly distributed. Fig. 1(b) shows that the stay events usually occurs between 10 and 15 o'clock, and the stay time is mainly concentrated between 10

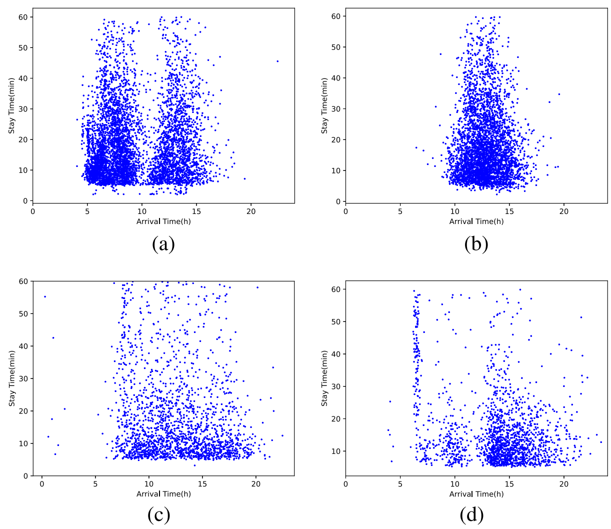

Fig. 1. Temporal distribution of stay events.

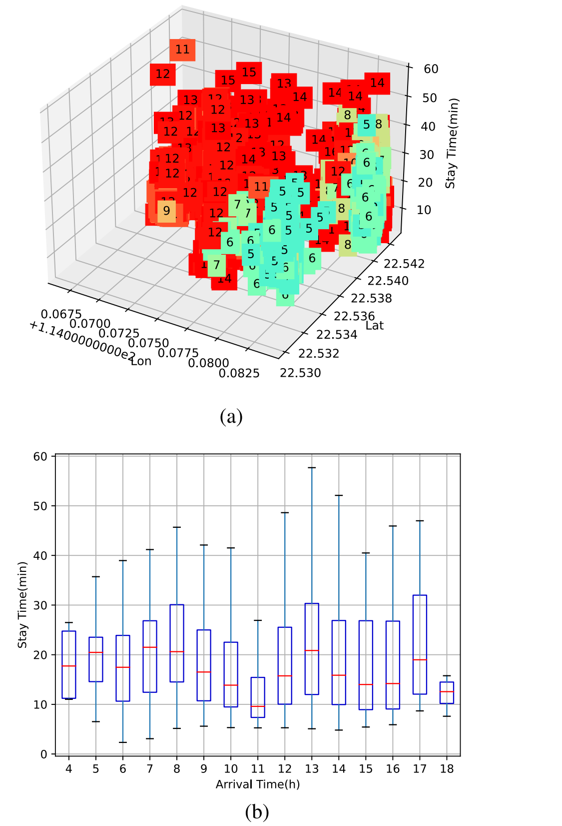

Fig. 2. Example of stay time distribution. Numbers in different colors represent different arrival times. In the same stop points, stay time changes according to arrival time.

and 30 minutes. Fig. 1(c) shows that the stay events are evenly distributed in the period from 7 to 18:00, and the stay time is evenly distributed in each period. Fig. 1(d) shows that the stay event occurred during the period from 12 to 18 o'clock. Between 12 and 18 o'clock, the stay time is mostly between 5 minutes and 20 minutes. It is worth noting that a lot of stay time also occurred at 7 o'clock, and the stay time at this time was evenly distributed between 5 minutes and 60 minutes. The above four users fully demonstrate the uniqueness of the stay event.

Fig. 2(a) illustrates the spatiotemporal distribution of stay time, in which the x-axis and y-axis represent the coordinates of stop points, and the z-axis provides the stay time of the private car user. The numbers in Fig. 2 represent the arrival time corresponding to the stay events. Figure 2 a shows two clusters with details in this small area. The stay time corresponding to the arrival time from 12 o'clock to 14 o'clock is the red part, and the stay time corresponding to the arrival time from 5 o'clock to 6 o'clock is shown in another colour. This shows that when the spatial characteristics are similar, the influence of the temporal characteristics on the stay time cannot be ignored. Furthermore, Fig. 2(a) shows the characteristics of the stay time corresponding to different arrival times in the same small area. In the same small area, when the arrival time is 4 o'clock, the stay time is within 10 to 20 minutes. After reaching 5 o'clock in time, the range of stay time begins to increase upwards and downwards. Until the arrival time is from 12:00 to 14:00, the range of stay time reaches the maximum. It is worth noting that when the arrival time is 11 o'clock, the increasing trend of the stay time range is interrupted, and the stay time is rapidly reduced. When the arrival time is 18:00, the stay time is shortened and concentrated. In summary, we found that only focusing on the spatial features in Fig. 2 and the staying time of users in an area is sparse and random. However, by combining the temporal and spatial characteristics, we found that the stay behaviour has an apparent clustering trend. The stay time pattern can also be reflected according to the arrival time. The above introduction fully illustrates that the temporal and spatial characteristics of the user's vehicle provide the possibility of predicting the stay time. In summary, we found that only focusing on the spatial features in Fig. 2 and the staying time of users in an area is sparse and random. However, by combining the temporal and spatial characteristics, we found that the stay behaviour has an apparent clustering trend. The stay time pattern can also be reflected according to the arrival time. The above introduction fully illustrates that the temporal and spatial characteristics of the user's vehicle provide the possibility of predicting the stay time.

The above information is indirectly included in the travel data. Our goal is to predict the stay time y_i by making use of the travel information S^i_{TI} of private cars. In the next section, we propose capturing the spatiotemporal features according to the changes in the arrival times and stop points of stay via constructing an RNN-based stay time prediction model.

IV. Methodology

M. C [23] proposed that human travel behavior has a highly temporal and spatial pattern and that human travel behavior has certain similarities. J. M [1] used a machine learning algorithm to predict stay time at WiFi hotspots. Rely on the powerful representation ability of neural networks for spatiotemporal features, J. Chen [10] integrated a decision tree with a recurrent neural network (RNN) to predict the stay time of vehicles via extracting spatiotemporal features. To obtain the spatiotemporal representation of stay events, we use a multilayer perceptron

(MLP) and gated recurrent units (GRUs) in the model to map the original data to the hidden feature space. After encoding through MLP and GRU, we obtain the spatiotemporal representation of the hidden layer space, and then, we use the fully connected layer to map the distributed feature representation into the sample space of hidden vectors. Our work validates the new idea of linear relationships in deep learning related to recent innovations in deep learning architectures. Many popular neural network architectures [44] [45] also advocate using linear links to reduce exploding/vanishing gradients or check the relationship between nodes. In connection, the linear relationship thinking in our article is also in line with a broader topic in machine learning, which attempts to identify the system's underlying structure in the form of behavioural control equations, which can reasonably infer the invisible part of the space. This is also a strong trend in recurrent networks. We use the NALU layer to reconstruct the basic arithmetic logic of the hidden vectors in the sample space to enhance the processing ability of the linear relationship of the model. Neural networks began to pay attention to the linear relationship between input data. Through the calculation of the linear relationship, the long-term stay event is connected with other stay events. At this time, the long stay incident was also affected. It is considered a meaningful event and will not become an abnormal situation ignored by the neural network.

Finally, to prevent overfitting, we use an MLP with a dropout function to decode the hidden vector into the final prediction value of the stay time.

A. Gated Recurrent Unit

A gated recurrent unit (GRU) is a variant of a recurrent neural network (RNN). The unit introduces a gating mechanism to avoid the vanishing gradient. It is simpler than other RNN variants (such as LSTM). The output of the hidden layer h_t of the GRU is calculated as the following function:

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \tag{3}$$

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi}) \tag{4}$$

$$n_t = tanh(W_{in}x_t + b_{in} + rt(W_{hn}h_{(t-1)} + b_{hn})) \tag{5}$$

$$h_t = (1 - i_t)n_t + i_t * h_{t-1} \tag{6}$$

where σ is a Sigmoid function, and r_t, i_t and n_t are the reset gate, update gate and cell state, respectively. The role of reset gate r_t is to determine how much of the information in the previous cell hidden state n_t needs to be forgotten. The role of update gate i_t is to determine how much information from the previous hidden layer state is passed to the current hidden state h_t. These gates control the learning process of the neural unit and are formed from a large amount of training data. However, the stay data of private cars are obviously sparse, which makes it difficult to learn a gate with good function. The adjacent stay behaviors will affect one another. To avoid this influence, we randomly input the spatiotemporal features into the model. We believe that the temporal influence should be explicitly entered into the learning of the gate mechanism, so we use the arrival time as one of the inputs, as previously described.

B. Neural Arithmetic Logic Units

The generalization of neural networks has been the focus of many researchers. In short, neural networks are more similar to memories than to learning. The reason [13] why the network's behavior does not generally appear as systematic may be the large number of nonlinear activation functions used in neural networks. Inspired by the idea of enhancing the linear relationship in neural networks, A. Trask *et al.* [13] proposed the neural arithmetic logic unit (NALU), as shown in the Fig. 4.

A neural arithmetic logic unit uses two neural accumulator units (purple circles) with bound weights to support basic arithmetic functions, controlled by a gate (orange circles) as follows:

$$\mathbf{W} = tanh(\hat{\mathbf{W}}) \odot \sigma(\hat{\mathbf{M}}) \quad a = \mathbf{W}x \tag{7}$$

$$m = exp\mathbf{W}(log(\mid x \mid + \varepsilon)) \quad g = \sigma(\mathbf{G}x) \tag{8}$$

$$y = g \odot a + (1 - g) \odot m \tag{9}$$

where $\mathbf{W}$ is guaranteed to be in the range [-1:1] and biased to be close to -1, 0, and 1. m saves the results of running in log space. g is a learned Sigmoidal gate to control basic arithmetic function types. As mentioned earlier, there are a large number of linear relationships between stay events. In the face of new small changes, neural networks need to respond sensitively. NALUs greatly improve the learning ability of the linear relationship between the neural network and the extrapolation ability of the prediction model.

C. Stay Time Prediction Model

As discussed in Section III, the travel information of private cars is sparse, and stay time is unstable. An MLP and RNNs can be used with travel information to capture spatiotemporal characteristics. We leverage this insight in designing the encoder, which addresses the problem of spatiotemporal sparsity, to extract spatiotemporal features. Our prediction model consists of three key components—an encoder module, an exception module, and MLP dropout—as shown in Fig. 3. The exception module is based on the NALU layer, where we reconstruct the basic arithmetic logic relationship of hidden state. MLP dropout takes a hidden vector and outputs predicted stay time $\hat{y}_i$. This module reduces the possibility of model overfitting by dropout.

1) Encoder: We already know that spatiotemporal information is related to the stay time. To extract spatiotemporal features from travel information, we first embed S^i_{TI} using an MLP to obtain a fixed-length hidden vector h_i. S^i_{TI} contains the user's arrival time (t_i) and arrival location (lon_i, lat_i). As mentioned earlier, we believe that the arrival time of the stay event and different time intervals of adjacent stay events are highly related to the change in stay time. We mitigate a certain time sparsity effect by using a single input. The hidden vector h_i contains complete spatiotemporal information for trips. Recurrent neural networks are often used in encoders and encode the input feature into a fixed-length hidden vector, which contains most of the information in the feature. Recurrent neural networks can generate similar probability distributions. The stay time is a space- and time-varying process. We use some GRUs to capture spatiotemporal changes in hidden space, which can effectively accumulate

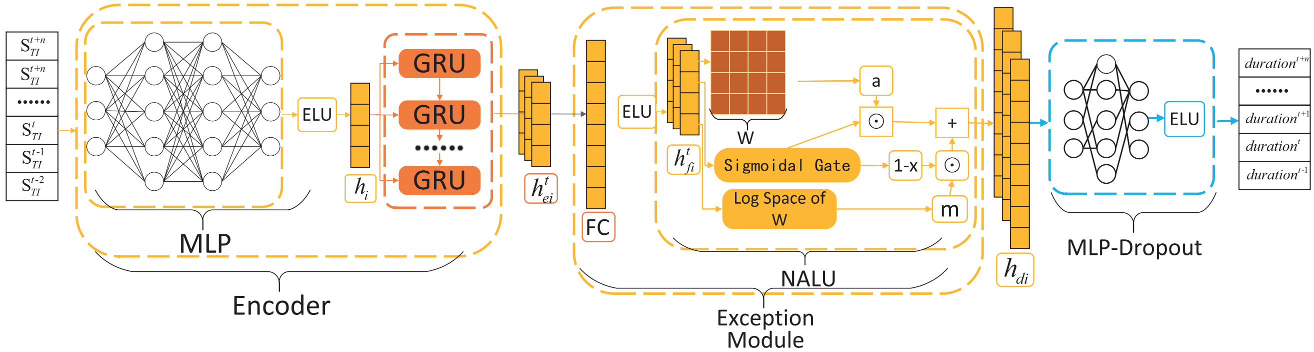

Fig. 3. The architecture of the stay time prediction model. MLP: Multilayer perceptron, ELU: Exponential linear unit, GRU: Gated recurrent unit, FC: Fully connected, and "⊙" denotes the elementwise product.

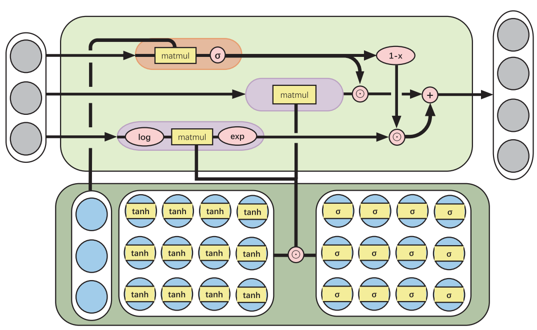

Fig. 4. An overview of neural arithmetic logic units. "⊙" denotes the elementwise product of tanh($\hat{\mathbf{W}}$) and $\sigma(\hat{\mathbf{M}})$. The NALU uses two neural accumulators (NACs, two purple cells) with tied weights to enable arithmetic functions, controlled by a gate (one orange cell).

spatiotemporal feature information. The hidden vectors are used as input for the encoder module as follows:

$$h_i = \phi(lon_i, lat_i, t_i : W_{ee}) \tag{10}$$

$$h_{ei}^n = GRU(h_{ei}^{n-1}, h_i : W_{Encoder}) \tag{11}$$

where $\phi(\cdot)$ is an embedding function with ELU nonlinearity, W_{ee} is the embedding weight, and $W_{Encoder}$ is the GRU cell weight.

2) Exception Module: Private car trips have some similarities in space and time. To handle the linear relationship between similar trips, we need to share information for all trips in the predicted model. However, deep learning models face some difficulties in dealing with linear relationships and are not very sensitive to noise. Therefore, to reason on all trips, we design an exception module. We pass the hidden vector h_{fi}^n through a fully connected layer for feature extraction. The NALU layer reconstructs the hidden vector h_{fi}^n. As one of the basic structures of the exception module, the NALU layer can handle linear relationships between features, such as addition and multiplication, by reconstructing basic arithmetic relationships.

The current approaches to modeling numeracy in neural networks fall short because numerical representations fail to generalize outside of the range observed during training. The NALU structure can be applied to rectify these shortcomings across a wide variety of domains, facilitating both numerical representations and functions on numerical representations that generalize outside of the range observed during training [13]. It is noteworthy that the NALU layer is a separate unit. Without changing the original model structure, the NALU layer is capable of achieving good performance in regard to the sparseness and instability of travel information and can effectively handle exception events:

$$h_{fi}^n = \omega(h_{ei}^n) \tag{12}$$

$$m = exp\mathbf{W}(log(\mid h_{fi}^n \mid + \varepsilon)) \tag{13}$$

$$g = \sigma(\mathbf{G}h_{fi}^n) \tag{14}$$

$$h_{di} = g \odot a + (1 - g) \odot m \tag{15}$$

where ε prevents log0. **w** is one of -1, 0, 1, and g is a learned Sigmoidal gate. The NALU cell learns arithmetic functions consisting of multiplication, addition, subtraction, division, and power functions, which extrapolate to numbers outside of the range observed during training.

3) MLP dropout: This part consists of an MLP dropout. We decode the hidden state h_{di} by the MLP dropout. The predicted value of the final output stay time $\hat{y}_i$ is as follows:

$$\hat{y}_i = \psi(h_{di} : W_{MLP-dropout}) \tag{16}$$

where $\psi(\cdot)$ is an MLP with ELU nonlinearity, and $W_{MLP-dropout}$ is the MLP dropout weight. The data imbalance is obvious in this research. The main problem of data imbalance is that the model will focus on the main situation, leading to serious overfitting. We have adopted two methods in our model to prevent overfitting. On the one hand, we set the weight decay in the hyperparameters to adjust for the influence of model complexity on the loss function. On the other hand, we setup a dropout layer to abandon some of the parameters in the model so that the model is not prone to overfitting. To solve the overfitting problem, we use dropout in the final MLP. We randomly delete a part of the neurons with a certain probability during training. With each training, different networks produce

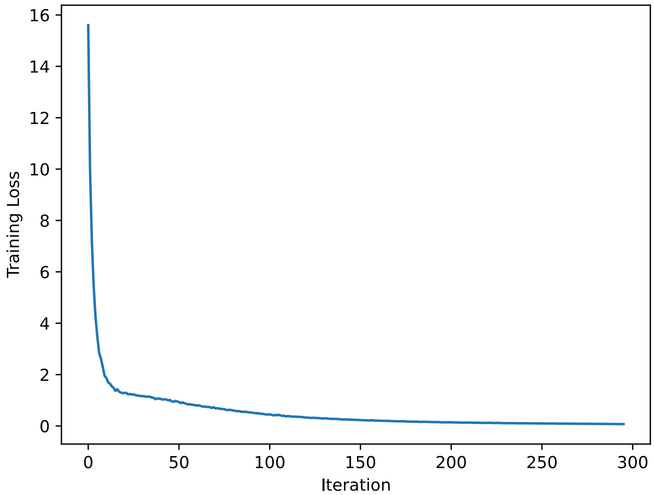

Fig. 5. During the training process, the loss value changes with the number of iterations.

different overfits. Some of the inverse fits cancel each other out and reduce overfitting as a whole.

D. Implementation Details

We implement our prediction model with Pytorch. The MLP of the encoder module is implemented by three fully connected layers with 64, 256 and 512 neurons, respectively. The input dim and output dim of the NALU is set to 512. The dropout value of the MLP dropout is set to 0.2. For the whole model, the parameters are initialized by default setting. The size of a mini-batch is set to 64. The learning rate is set to 10^{-8}. We optimize our model in an end-to-end manner via Adam optimization [51] by minimizing the MSE loss between the ground truth and the predicted stay time.

V. Experiments

Our previous works [6], [48], [52] have obtained a large trajectory dataset of private cars. The data set we use contains the trajectory information of more than 4000 users. However, in the experimental part, we only used the data of one user for training and testing. Most users' stay events are about a thousand times. In order to thoroughly study various features of stay events, we selected users with more stay events to conduct research and analysis. In this section, we use the trajectory data from January 2016 to August 2018. During this time, users had 5,141 effective stay events. We randomly use 80% of effective stay events as a training set and 20% of effective stay events as a test set. To reduce the impact of data imbalance, we resample the training dataset. The final training dataset contains 5,517 stay events, and the test dataset contains 1,000 stay events.

Algorithm 2 presents the process of the prediction model based on an RNN. First, we process the S_{TI} data from the private car trajectory dataset, extract the effective stay events, divide them into training data and test data, and then use the RNN-based prediction model to predict stay time. The Fig. 5 shows that as the number of iterations increases, the Loss value can always converge to close to zero.

Algorithm 2: Stay Time Prediction Model.

Input: $\mathbf{S}_{TI}$ retrieved from the private car trajectory dataset
Output: The predicted value $\hat{y}_i$ of the stay time
function Clean the dataset
  Extract S_{TI} from the dataset
  stay event detection
  Separate data by User ID
  Normalize the data of each user
end function
function Neural Networks based on an RNN
  Split Data into Training Data and Test Data, and change the shape of datasets
  Resample Training Data
  Initialize the prediction model
  for i = 1 to EPOCH**do**
    $h_i = MLP(lon_i, lat_i, t_i)$
    for n = 1 to N**do**
      $h_{ei}^{n} = GRUcell(h_{ei}^{n-1}, h_i)$
    end for
    $h_{di} = Exception\,Module(h_{ei}^{n})$
    $\hat{y}_i = MLPdropout(h_{di})$
end for
end function

A. Alternative Techniques

Linear regression (LR) is a statistical analysis method that uses regression analysis in mathematical statistics to determine the quantitative relationships between two or more variables.

Multilayer perceptron (MLP) is a feedforward artificial neural network model that maps multiple datasets of an input to a single output.

Support vector regression (SVR) [53] is a supervised learning technique, maintaining all the main features that characterize the algorithm. SVR perform well in time series and nonlinear prediction.

Decision tree (DT) [10] is a predictive model that represents a mapping relationship between object attributes and object values.

Long short-term memory (LSTM) [54] is a type of recurrent neural network. This network is generally used to predict time series. To make a fair comparison with our proposed method, we use a gridsearch method to find the best parameters of the benchmark model. The maximum depth is set to 19, and the minimum sample leaf is set to 3 in the DT. The C is set to 1, and the gamma is set to 100 in SVR.

Graph Convolution Networks (GCN) [55] is a neural network that operates on graph data and is currently one of the most popular predictive models.

Hierarchical GCN (HGCN) [56]is a new type of graph neural network used for prediction. It uses a micro-layer to capture the relationship between nodes and a macro-layer to capture the relationship between regions.

B. Analysis of Results

To evaluate the performance of our predictive models, we conduct quantitative studies based on four metrics: root mean square error (RMSE), mean absolute error (MAE), symmetric mean absolute percentage error (SMAPE) and mean absolute percentage error (MAPE). The RMSE is used to measure the deviation between the observed value and true value. The MAE can better reflect the actual situation of the predicted value error. The closer the RMSE and MSE are to 0, the closer the predicted value is to the true value. Let y_i and $\hat{y}_i$ denote the ground truth and prediction, respectively. The RMSE and MAE are calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{17}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{18}$$

For the MAPE and SMAPE, to evaluate the relative error of the predicted value $\hat{y}_i$ and the ground truth value y_i, the calculations are shown below:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{19}$$

$$SMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{(|\hat{y}_i| + |y_i|)/2} \tag{20}$$

In addition, to measure the distribution difference between the predicted value and the true value, we select the KL divergence, which reflects the degree of difference between the two probability distributions, for comparison. The KL divergence represents the information loss caused by fitting the theoretical distribution to the true distribution, which is given as follows:

$$D_{KL(p\|q)} = \sum_{i=1}^{N} p(x_i)\log\frac{p(x_i)}{q(x_i)} \tag{21}$$

where $p(x_i)$ is the ground truth distribution, and $q(x_i)$ is the predicted distribution to match. If the two distributions exactly match, then $D_{KL(p\|q)}$ is close to 0.

Table III presents the results in terms of the RMSE, MAE, MAPE and SMAPE. It is obvious that the proposed method has better performance than the comparison method. For example, by using the proposed method, the percentage improvement of the RMSE, MAE, MAPE and SMAPE are 0.1429, 0.1103, 55.8533%, and 47.3190%, respectively.

To verify the divergence between the predicted stay time distribution and the true stay time distribution, we calculated the KL divergence for different time periods. Our proposed model performs better than other methods in terms of KL divergence, except for the LR and LSTM methods in Table V. This shows that the divergence between the predicted stay time of our proposed method and the real stay time is smaller than that of most comparison methods in probability distributions. Recurrent

TABLE III
PREDICTION MODEL PERFORMANCE

	RMSE	MAE	MAPE	SMAPE
LR	0.2370	0.2025	112.3124%	61.6588%
DT	0.2210	0.1476	62.6358%	43.7662%
MLP	0.2165	0.1758	91.7414%	54.1442%
SVR	0.1921	0.1483	75.0817%	47.1654%
LSTM	0.2259	0.1986	113.5623%	61.1772%
GCN	0.2153	0.1768	93.1810%	53.1487%
HGCN	0.1964	0.1589	82.3310%	49.4188%
our model	**0.1429**	**0.1103**	**55.8533%**	**47.3190%**

neural networks are often used in encoder modules. Recurrent neural networks encode the input feature into a fixed-length hidden vector, which contains most of the information of the feature. By decoding this hidden vector, we can predict stay time. A recurrent neural network is part of our model, as it can show good results for KL divergence when it makes predictions alone. This shows that recurrent neural networks can generate similar probability distributions. We have also seen that the linear regression model also performs well on KL divergence. This verifies that the previously mentioned stay event contains some linear relationships.

The results in Table IV and Table VI show that the proposed method is better than other methods in predicting different stay times. We observed that the LSTM model, LR model and SVR model performed close to our method with stay time of 15-30 minutes. The reason is that the extrapolation ability of these methods is worse than that of our model, resulting in the predicted data being concentrated in a small range. We apply the NALU layer in the proposed method, which effectively addresses some abnormal stay situations. For example, when a commonly used stay place is unavailable, the user may change the stay time due to a nearby stay point.

In the short-term stay time prediction, The comparison method is close to the effect of our model, but our model is still better than theirs. Firstly, short-term stay events are the most frequent. In the comparison methods, such as comparison method, some long stay times are rare abnormalities and are not sensitive to these abnormal samples, accordingly, these methods ignore the long stay time and focus on the short stay time. Comparison methods tend to get a locally optimal prediction value, on this basis, to meet the pre-set prediction performance requirements. Under normal circumstances, this locally optimal prediction value will fall within the short stay time with the enormous sample data volume, so in the case of short dwell time prediction, our method is not much different from the comparison method. Secondly, the Spatio-temporal features mentioned in the article cannot fully represent the stay time in a shorter stay time. Many other factors also affect the stay time, so the ability to extrapolate only using temporal and spatial features is limited. On the other hand, for long-term stay time prediction, multi-layer perceptron in the decoder structure encodes the input features. At this time, the encoded feature space forms a hyperplane between different

TABLE IV
PERFORMANCE COMPARISON IN DIFFERENT SCENARIOS

Model	0-15 min			15-30 min			30-45 min			45-60 min		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
LR	0.2527	187.6026%	0.2706	0.1050	32.3554%	0.1325	0.1884	30.7990%	0.2159	0.4610	53.1360%	0.4688
DT	0.1199	84.2855%	0.2033	0.1409	41.1135%	0.1908	0.2080	34.5316%	0.2541	0.4217	48.6417%	0.4865
MLP	0.1965	144.2625%	0.2362	0.1337	40.4893%	0.1602	0.1484	24.5725%	0.1879	0.3838	43.9203%	0.4102
SVR	0.1575	115.7436%	0.2044	0.1163	35.1704%	0.1430	0.1388	22.8367%	0.1717	0.3636	41.3133%	0.3947
LSTM	0.2623	193.3853%	0.2672	0.0841	27.2407%	0.0999	0.1755	28.4404%	0.1901	0.4620	53.1975%	0.4677
GCN	0.1979	145.2909%	0.2353	0.1527	45.5126%	0.1866	0.1177	19.1941%	0.1450	0.3050	35.2575%	0.3375
HGCN	0.1730	128.0580%	0.2096	0.1284	38.3359%	0.1582	0.1319	21.2654%	0.1616	0.3444	39.8820%	0.3704
Our model	**0.1163**	**85.0820%**	**0.1505**	**0.0857**	**25.6915%**	**0.1086**	**0.0985**	**16.1607%**	**0.1211**	**0.2483**	**28.3601%**	**0.2729**

TABLE V
KL DIVERGENCE

	0-15 min	15-30 min	30-45 min	45-60 min
LR	0.0687	0.0567	0.0289	0.0119
DT	0.2009	0.1213	0.1021	0.1716
MLP	0.1074	0.0719	0.0449	0.0405
SVR	0.1273	0.0634	0.0327	0.0381
LSTM	0.0478	0.0196	0.0081	0.0048
GCN	0.1614	0.0554	0.0265	0.0363
HGCN	0.1448	0.0462	0.0375	0.0323
Our model	**0.0945**	**0.0368**	**0.0143**	**0.0093**

TABLE VI
SMPAE

	0-15 min	15-30 min	30-45 min	45-60 min
LR	87.9668%	28.5890%	38.5608%	73.0217%
DT	44.7737%	38.9513%	45.0253%	72.5329%
MLP	71.4482%	35.8982%	30.3688%	58.8860%
SVR	61.0437%	32.0748%	27.2601%	54.6192%
LSTM	61.1772%	22.7109%	33.7890%	72.7412%
GCN	73.0638%	34.6911%	25.5322%	45.1925%
HGCN	67.0884%	30.8414%	24.9554%	52.0511%
Our model	**47.3190%**	**49.9227%**	**22.9748%**	**18.0036%**

classes. Since we reconstructed the basic arithmetic and logical relationship in the abnormal module, the neural network began to pay attention to the linear relationship between the input data. Through the calculation of the linear relationship, the long-term stay event was connected with other stay events. At this time, the long-term stay event was also affected. It is considered a meaningful event and will not become an abnormal situation ignored by the neural network.

In the case of more than 30 min, our method is significantly better than the comparison method. Because of the sparseness of the stay behavior, there are some differences between the stay behavior in the test dataset and that in the training dataset. The NALU structure greatly improves the extrapolation ability of our model, so our model has better predictive ability for staying behavior that has not appeared in the training dataset.

VI. CONCLUSION

In this paper, we introduce an interesting task to explore human travel behavior, which aims at predicting the stay time of private cars. We propose a prediction model for short-term private car stay time based on an RNN and use a large-scale real-world private car trajectory dataset for verification. As described in Section III, we believe that behaviors such as private car stay have strong randomness and sparseness. Therefore, we introduce an exception module to deal with such problems and comprehensively consider all the travel information of users. The experimental results show that our prediction model achieves an RMSE of 0.1429, an MAE of 0.1103, an SMAPE of 47.3190% and an MAPE of 55.8533%. Moreover, it is superior to other methods in four different stay time prediction tasks. In addition, we explore the utility of the predicted values. The KL divergence is the closest to the true values. We proved a linear relationship problem that cannot be ignored in the stay time prediction problem. Although the basic network structure provides a solid nonlinear fitting ability, it is challenging to learn the linear relationship in the stay time feature. In particular, we have added basic arithmetic logic units to the network framework to find linear relationships. By reconstructing the basic arithmetic and logical relations of the network, we have improved the ability of the neural network to handle linear relations and the extrapolation ability of the neural network. Our method can remember the number patterns seen in the training set very well and infer this representation reasonably. Our neural network can also respond quickly to tilting the numerical representation beyond the range of numbers seen in the training data.

How private car stay events reflect human mobility remains an open question. The behavioral characteristics of hundreds of millions of vehicles in the world can provide a new perspective for social networks. In future work, we will devote our efforts to exploring the relationship between private car stay events and contextual information and further studying the impact of private car stay events on the transportation network. Additionally, we will collect easy-to-obtain external information (such as that on weather and regional traffic flow) and add to our prediction model to further enhance the prediction accuracy of private car stay time. It is expected that such predictive models will promote the development of intelligent transportation systems.

REFERENCES

[1] J. Manweiler, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Predicting length of stay at WiFi hotspots," in *Proc. IEEE INFOCOM*, 2013, pp. 3102–3110.

[2] D. Wang *et al.*, "Stop-and-wait: Discover aggregation effect based on private car trajectory data," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3623–3633, Oct. 2018.

[3] Z. Li, K. Ishibashi, K. Takai, and K. Yada, "Shop area visit ratio, stay time, and sales outcomes: In-depth analysis based on RFID data," in *Proc. 2nd Asia-Pacific World Congr. Comput. Sci. Eng.*, 2015, pp. 1–7.

[4] Y. Huang, Z. Xiao, D. Wang, H. Jiang, and D. Wu, "Exploring individual travel patterns across private car trajectory data," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 12, pp. 5036–5050, Dec. 2020, doi: 10.1109/TITS.2019.2948188.

[5] T. Li, A. Alhilal, A. Zhang, M. A. Hoque, D. Chatzopoulos, Z. Xiao, Y. Li, and P. Hui, "Driving Big Data: A first look at driving behavior via a large-scale private car dataset," in *Proc. IEEE 35th Int. Proc. Data Eng. Workshops*, 2019, pp. 61–68.

[6] Y. Huang, Z. Xiao, X. Yu, D. Wang, V. Havyarimana, and J. Bai, "Road network construction with complex intersections based on sparsely sampled private car trajectory data," *ACM Trans. Knowl. Discov. From Data*, vol. 13, no. 3, pp. 1–28, Jun. 2019.

[7] J. Chen *et al.*, "Toward opportunistic compression and transmission for private car trajectory data collection," *IEEE Sensors J.*, vol. 19, no. 5, pp. 1925–1935, Mar. 2019, doi: 10.1109/JSEN.2018.2885121.

[8] G. Yang, Y. Cai, and C. K. Reddy, "Spatio-temporal check-in time prediction with recurrent neural network based survival analysis," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2976–2983.

[9] L. Zheng, D. Xia, L. Chen, and D. Sun, "Understanding citywide resident mobility using big data of electronic registration identification of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4363–4377, Oct. 2020, doi: 10.1109/TITS.2019.2940724.

[10] J. Chen, Z. Xiao, D. Wang, W. Long, J. Bai, and V. Havyarimana, "Stay time prediction for individual stay behavior," *IEEE Access*, vol. 7, pp. 130 085–130 100, 2019.

[11] Y. Li, W. Ren, D. Jin, P. Hui, L. Zeng, and D. Wu, "Potential predictability of vehicular staying time for large-scale urban environment," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 322–333, Jan. 2013.

[12] D. Kondor, H. Zhang, R. Tachet, P. Santi, and C. Ratti, "Estimating savings in parking demand using shared vehicles for home-work commuting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2903–2912, Aug. 2018.

[13] A. Trask, F. Hill, S. E. Reed, J. Rae, C. Dyer, and P. Blunsom, "Neural arithmetic logic units," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8035–8044.

[14] B. M. Booth, T. Feng, A. Jangalwa, and S. S. Narayanan, "Toward robust interpretable human movement pattern analysis in a workplace setting," in *Proc. IEEE Int. Conf. Acoust. Speech Signal, Process*, 2019, pp. 7630–7634.

[15] E. V. Prawirodidjojo, R. J. Quek, B.-S. Lee, V. Gauthier, and M. Schläpfer, "Using mobile phone data to determine human mobility patterns in Paris," in *Proc. Int. Conf. Cyberworlds*, 2018, pp. 399–402.

[16] C. Zhang, M. Dong, T. H. Luan, and K. Ota, "Battery maintenance of pedelec sharing system: Big Data based usage prediction and replenishment scheduling," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 127–138, Jan.–Mar. 2020.

[17] J. Xing, Z. Liu, C. Wu, and S. Chen, "Traffic volume estimation in multimodal urban networks using cell phone location data," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 3, pp. 93–104, Jun. 2019.

[18] S. He, F. Ding, Y. Zhou, Y. Cheng, and B. Ran, "Investigating and modelling the relationship between traffic volume and extracts from cellphone activity data," *IET Intell. Transp. Syst.*, vol. 13, no. 8, pp. 1299–1308, 2019.

[19] S. Jiang, G. A. Fiore, Y. Yang, J. Ferreira Jr, E. Frazzoli, and M. C. González, "A review of urban computing for mobile phone traces: Current methods, challenges and opportunities," in *Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput.*, 2013, pp. 1–9.

[20] S. Büyükçorak, G. K. Kurt, and O. Cengaver, "A probabilistic framework for estimating call holding time distributions," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 811–821, Feb. 2013.

[21] S. Hasan, C. M. Schneider, S. V. Ukkusuri, and M. C. González, "Spatiotemporal patterns of urban human mobility," *J. Stat. Phys.*, vol. 151, no. 1/2, pp. 304–318, 2013.

[22] C. Song, T. Koren, P. Wang, and A.-L. Barabási, "Modelling the scaling properties of human mobility," *Nat. Phys.*, vol. 6, no. 10, pp. 818–823, 2010.

[23] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[24] N. Caceres, J. Wideberg, and F. G. Benitez, "Review of traffic data estimations extracted from cellular networks," *IET Intell. Transp. Syst.*, vol. 2, no. 3, pp. 179–192, 2008.

[25] M. G. Demissie, S. Phithakkitnukoon, and L. Kattan, "Trip distribution modeling using mobile phone data: Emphasis on intra-zonal trips," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2605–2617, Jul. 2018.

[26] J. L. Toole, S. Colak, B. Sturt, L. P. Alexander, A. Evsukoff, and M. C. González, "The path most traveled: Travel demand estimation using big data resources," *Transp. Res. Part C, Emerg. Technol.*, vol. 58, pp. 162–177, 2015.

[27] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 791–800.

[28] E. F. d. S. Soares, C. A. d. M. S. Quintella, and C. A. V. Campos, "Smartphone-based real-time travel mode detection for intelligent transportation systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1179–1189, Feb. 2021.

[29] H. Huang, Y. Cheng, and R. Weibel, "Transport mode detection based on mobile phone network data: A systematic review," *Transp. Res. Part C, Emerg. Technol.*, vol. 101, pp. 297–312, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X1831369X

[30] H. R. Eftekhari and M. Ghatee, "An inference engine for smartphones to preprocess data and detect stationary and transportation modes," *Transp. Res. Part C, Emerg. Technol.*, vol. 69, pp. 313–327, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X16300730

[31] H. R. Eftekhari and M. Ghatee, "Hybrid of discrete wavelet transform and adaptive neuro fuzzy inference system for overall driving behavior recognition," *Transp. Res. Part F, Traffic Psychol. Behav.*, vol. 58, pp. 782–796, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369847817306216

[32] D. I. Tselentis, E. I. Vlahogianni, and G. Yannis, "Driving safety efficiency benchmarking using smartphone data," *Transp. Res. Part C, Emerg. Technol.*, vol. 109, pp. 343–357, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18308271

[33] S. Kanarachos, S.-R. G. Christopoulos, and A. Chroneos, "Smartphones as an integrated platform for monitoring driver behaviour: The role of sensor fusion and connectivity," *Transp. Res. Part C, Emerg. Technol.*, vol. 95, pp. 867–882, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18303954

[34] Y. Cui, Q. He, and L. Bian, "Generating a synthetic probabilistic daily activity-location schedule using large-scale, long-term and low-frequency smartphone GPS data with limited activity information," *Transp. Res. Part C, Emerg. Technol.*, vol. 132, 2021, Art. no. 103408. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X21004034

[35] O. Burkhard, H. Becker, R. Weibel, and K. Axhausen, "On the requirements on spatial accuracy and sampling rate for transport mode detection in view of a shift to passive signalling data," *Transp. Res. Part C, Emerg. Technol.*, vol. 114, pp. 99–117, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X1831814X

[36] W.-H. Lee, C.-H. Hu, and S.-R. Hu, "A travel time prediction model for freeway with considering interchange disturbances," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst.*, 2014, pp. 2174–2179.

[37] T. Derrmann, R. Frank, S. Faye, G. Castignani, and T. Engel, "Towards privacy-neutral travel time estimation from mobile phone signalling data," in *Proc. IEEE Int. Smart Cities Conf.*, 2016, pp. 1–6.

[38] W. Guo, L. Li, Z. Li, J. Wang, Z. Ming, and P. Jiang, "Exploring human stay time patterns from mobile phone data," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 1378–1383.

[39] S. Liu, H. Cao, L. Li, and M. Zhou, "Predicting stay time of mobile users with contextual information," *IEEE Trans. Automat. Sci. Eng.*, vol. 10, no. 4, pp. 1026–1036, 2013.

[40] R. Low, L. Cheah, and L. You, "Commercial vehicle activity prediction with imbalanced class distribution using a hybrid sampling and gradient boosting approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1–10, Mar. 2020.

[41] R. Low, Z. D. Tekler, and L. Cheah, "Predicting commercial vehicle parking duration using generative adversarial multiple imputation networks," *Transp. Res. Rec.*, vol. 2674, no. 9, pp. 820–831, 2020.

[42] K. S. Kung, K. Greco, S. Sobolevsky, and C. Ratti, "Exploring universal patterns in human home-work commuting from mobile phone data," *PLoS One*, vol. 9, no. 6, 2014, Art. no. e96180.

[43] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 630–645.

[45] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[46] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014, *arXiv:1410.5401*.

[47] E. Grefenstette, K. M. Hermann, M. Suleyman, and P. Blunsom, "Learning to transduce with unbounded memory," *in Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1828–1836.

[48] Z. Xiao *et al.*, "Toward accurate vehicle state estimation under non-Gaussian noises," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 652–10 664, Dec. 2019.

[49] Z. Xiao, P. Li, V. Havyarimana, G. M. Hassana, D. Wang, and K. Li, "GOI: A novel design for vehicle positioning and trajectory prediction under urban environments," *IEEE Sensors J.*, vol. 18, no. 13, pp. 5586–5594, Jul. 2018.

[50] C. C. Robusto, "The cosine-haversine formula," *Amer. Math. Monthly*, vol. 64, no. 1, pp. 38–40, 1957.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[52] Z. Xiao, F. Li, R. Wu, H. Jiang, Y. Hu, J. Ren, C. Cai, and A. Iyengar, "TrajData: On vehicle trajectory collection with commodity plug-and- play OBU devices," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9066–9079, Sep. 2020.

[53] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[55] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv*:1609.02907.

[56] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, and B. Yin, "Hierarchical graph convolution networks for traffic forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 1, pp. 151–159, 2021.

Qibo Zhang is currently working toward the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include intelligent transportation system and neural network.

Fanzi Zeng (Member, IEEE) received the Ph.D. degree in signal and information processing from Beijing Jiao tong University, Beijing, China, in 2005. Since 2005, he has been with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, where he is currently a Professor. In 2009, he was with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA, as a Visiting Scholar. His research interests include cognitive radio technology and artificial intelligence.

Zhu Xiao (Senior Member, IEEE) received the M.S. and Ph.D. degrees in communication and information system from Xidian University, Xi'an, China, in 2007 and 2009, respectively. From 2010 to 2012, he was a Research Fellow with the Department of Computer Science and Technology, University of Bedfordshire, Luton, U.K. He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include Internet of Vehicles and trajectory data mining. He is serving as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

Hongbo Jiang (Senior Member, IEEE) received the Ph.D. degree from Case Western Reserve University, Cleveland, OH, USA, in 2008. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. He was a Professor with the Huazhong University of Science and Technology, Wuhan, China. His research interests include computer networking, especially algorithms and protocols for wireless and mobile networks. He is the Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING, an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING, and an Associate Technical Editor for *IEEE Communications Magazine*. He is an Elected Fellow of IET, Fellow of BCS.

Amelia C. Regan (Member, IEEE) received the B.A.S. degree in systems engineering from the University of Pennsylvania, Philadelphia, PA, USA, the M.S. degree in applied mathematics from Johns Hopkins University, Baltimore, MD, USA, and the M.S.E. and Ph.D. degrees in civil transportation systems engineering from the University of Texas at Austin, Austin, TX, USA. She was a Software Engineer and Operations Research Analyst for the Association of American Railroads and UPS earlier in her career. She has been a Professor of computer science and transportation systems engineering with the University of California Irvine, Irvine, CA, USA, since 1997.

Kehua Yang received the Ph.D. degree from Southeast University, Nanjing, China, in 2005. Since 2005, he has been with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, where he is currently an Associate Professor. From 2017 to 2018, he was with Virginia Polytechnic Institute, Blacksburg, VA, USA, and with State University, Virginia, Petersburg, VA, USA, as a Visiting Scholar. His research interests include database and embedded software development.

Yongdong Zhu received the bachelor's degree from Xi'an Jiaotong University, Xi'an, China, in 1997, and the Ph.D. degree from the University of Essex, Colchester, U.K., in 2007. He is currently a Professor with the Institution of Intelligent Network, Zhejiang laboratory, China. His research interests include next-generation mobile communication, mobile edge network architecture, vehicular communication networks, and Internet of Things.