Improving Commute Experience for Private Car Users via Blockchain-Enabled Multitask Learning

Jiali Yang, Kehua Yang, Zhu Xiao, *Senior Member, IEEE*, Hongbo Jiang, *Senior Member, IEEE*, Shenyuan Xu, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*—With deepening urbanization and Internet of Vehicles (IoV) applications, the number of private cars has been increasing in recent years. However, because the surging number of private cars is not compatible with limited road resources, private car users have had unsatisfactory commute experiences during their daily travel. In this work, we focus on improving private car users' commute experience based on an analysis of IoV trajectory data in a privacy-preserving way. Our idea is based on the following observations: 1) the commute experience of private car users is closely related to the departure time and the travel cost and 2) most travel costs are spent on urban hot zones. Motivated by these findings, we propose a novel blockchain-enabled model named Deep Improving Commute Experience (DeepICE) to improve private car users' commute experience by predicting when to depart and when to arrive. In this model, a blockchain with a consensus mechanism is developed to address private car user privacy concerns. In addition, we propose a multitask learning-enabled graph convolution network (GCN) method to capture the highly complex features and relations between two tasks, i.e., the departure time and travel cost, and then develop the model to predict these two tasks. The experimental results demonstrate the superior performance of our proposed model compared to existing approaches. Our model can be applied to efficiently enhance private car users' commute experience.**

***Index Terms*—Blockchain, commute experience, multitask learning, privacy-preserving, private car.**

Manuscript received 18 November 2022; revised 30 June 2023 and 7 September 2023; accepted 15 September 2023. Date of publication 20 September 2023; date of current version 7 December 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1713400 and Grant 2022YFE0137700; in part by the National Natural Science Foundation of China under Grant 62272152, Grant 62372161, and Grant U20A20181; in part by the Key Research and Development Project of Hunan Province of China under Grant 2022GK2020 and Grant 2021WK2001; in part by the Hunan Natural Science Foundation of China under Grant 2022JJ30171; in part by the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) under Grant GML-KF-22-22 and Grant GML-KF-22-23; in part by the CAAI-Huawei MindSpore Open Fund; in part by the Shenzhen Science and Technology Program under Grant CYJ20220530160408019; and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011915. *(Corresponding authors: Kehua Yang; Zhu Xiao; Hongbo Jiang.)*

Jiali Yang, Kehua Yang, Zhu Xiao, Hongbo Jiang, and Shenyuan Xu are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the AI Lab, Shenzhen Research Institute of Hunan University, Shenzhen 518055, China (e-mail: jialiyang@hnu.edu.cn; khyang@hnu.edu.cn; zhxiao@hnu.edu.cn; hongbojiang2004@gmail.com; syxu@hnu.edu.cn).

Schahram Dustdar is with the Research Division of Distributed Systems, TU Wien, 1040 Vienna, Austria (e-mail: dustdar@infosys.tuwien.ac.at).

Digital Object Identifier 10.1109/JIOT.2023.3317639

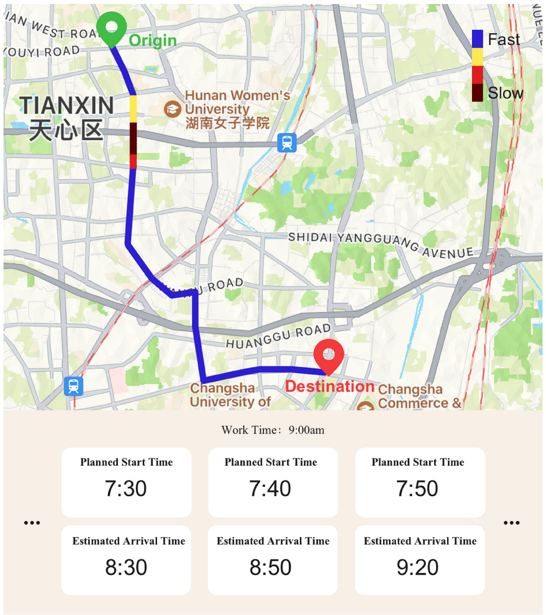
I. Introduction

DRIVEN by deepening urbanization and the advance of the Internet of Vehicles (IoV) [1], private cars have been experiencing continuously increasing growth for decades and constitute the vast majority of urban automobiles. For example, the number of private cars reached 207 million by 2020 in China, accounting for 79.61% of automobiles [2]. In particular, in the past five years, the average annual growth of automobiles was 19.66 million according to [2], and over 94% of them were private cars. In the U.S. and Canada, the number of private cars has exceeded 200 million, and 80% of Americans possess private cars [3]. With this background, an increasing number of people commute by using private cars to conduct daily activities and fulfill travel needs [4], [5], [6], [7].

During people's daily travel, they have unsatisfactory commute experiences because the surging number of private cars exceeds limited road resources. This problem motivates us to seek solutions to improve private car users' commuting experience. Doing so will not only help to decrease the probability of an unpleasant commute but also enhance private car users' driving safety, thereby alleviating traffic congestion and pollutant emissions and helping the government smartly manage urban traffic.

Specifically, we make the following observations on the commute experience of private cars according to [8] and [9].

1) The commute experience is closely related to the private cars' departure time and the travel cost [10], particularly the interaction of departure time and travel cost.
2) Existing works such as [11] find that a large number of urban residents choose similar travel routes for their commutes during working days. More specifically, private car users tend to choose an unchanged route for commutes regardless of the traffic condition and external factors, such as weather and the distribution of points of interest (PoIs) [11], [12]. For this reason, route optimization or path recommendation [13], [14], [15], [16], [17] methods would not help achieve a good commute experience.
3) When people use their private cars for commuting, they will spend more time commuting and passing through urban hot zones. As such, travel cost depends mostly on the hot zones that they drive through. Furthermore, popularity of hot zones and the number of hot zones in private car users' routes directly determine whether their commute route incurs high travel costs [18].



Work Time: 9:00am

...	Planned Start Time 7:30	Planned Start Time 7:40	Planned Start Time 7:50	...
	Estimated Arrival Time 8:30	Estimated Arrival Time 8:50	Estimated Arrival Time 9:20	

Fig. 1. Application of improving users commuting experience.

As illustrated in Fig. 1, the private car user sticks to the same route, and various departure times and travel costs provide varying commute experiences. The user could choose to leave at 7:30 A.M. if the driver needs to arrive at his office in advance, or the driver could leave at 7:40 A.M. if the driver is not in a hurry. However, if the driver departs at 7:50 A.M., the driver will be late. By selecting a suitable departure time and evaluating the expected travel cost, the private car user can obtain a satisfactory commute experience. Here, we emphasize that travel cost is tightly connected with travel time, namely, the time the private car users need to spend arriving at their commute destinations. A longer travel time makes them not only endure long commutes and congested traffic conditions but also face the risk of being late. Intuitively, if these *two tasks*, i.e., departure time and travel cost (e.g., travel time), can be accurately predicted, it will help enhance private car users commute experience.

In this article, we strive to improve private car users commute experience based on an analysis of their trajectory data. In recent years, deep learning (DL) methods have been developed to focus on trajectory data research. For example, Yao et al. [19], [20], and Zhou et al. [21] utilized cars' historical trajectory data to model and predict traffic. Li et al. [22], Chen et al. [23], and Zhou et al. [24] proposed evaluating the risk factors during traffic based on trajectory data. Motivated by these studies, we aim to improve private car users' commute experience by designing a DL-based approach with trajectory data.

Although existing solutions look promising, they introduce the following issue: *How can we jointly predict the departure time and travel cost based on private cars' trajectory data in a privacy-preserving manner?* To resolve this problem, we must address the following issues.

1) A more abnormal traffic environment occurs in the commute period. For example, there are more vehicles on the road and more congested travel conditions in this period. In addition, it is becoming increasingly difficult to extract the task dependencies between the two tasks, i.e., selecting a proper departure time and foreseeing the expected travel cost.
2) It is challenging to effectively represent temporal and spatial features during the learning process of task modeling. On the one hand, the spatial structures of hot zones are irregular, and the spatial correlations between these hot zones are complex. Hence, modeling the travel cost spent on the different hot zones is becoming more difficult. On the other hand, the temporal relations of the departure time and the travel cost are complicated due to their periodicity property [12].

To address the abovementioned challenges, many works have been proposed to incorporate different tasks to fully exploit the spatial and temporal relations. For example, Zhang et al. [25], Jiang et al. [26], Li et al. [27], and Zhou et al. [28], [29] introduced multitask learning to exploit the spatial and temporal relations between different tasks. They utilize traditional DL-based functions, such as fully connected networks and CNNs, as the kernel functions to extract the task dependencies from the training data set. One shortcoming of these works is that they cannot fully exploit irregular spatial structures to properly model task dependencies because the spatial dependency between hot zones is a non-Euclidean construction. Accordingly, several works [30], [31], [32], [33], [34] utilize graph structures to model irregular spatial constructions. Motivated by this approach, the graph structure is applied to simulate the irregular spatial structure. We purposely utilize GCN to construct the graph structure, which is one of the most popular algorithms in the field of graph DL. Nevertheless, using only graph structures cannot realize task dependency extraction.

In this article, we propose a blockchain-enabled multitask learning model [Deep Improving Commute Experience (DeepICE)], to efficiently improve private car users' commute experience in a privacy-preserving way. Within the proposed method, we first extract several hot zones based on spatial and temporal trajectory data through the spatial-temporal density-based spatial clustering of application with noise (ST-DBSCAN) algorithm, considering that most private car users' travel costs are spent on the hot zones of the city [18]. Note that the blockchain technology is applied in information sharing so that each private car user can acquire knowledge from others' trajectory data in a trustworthy way. Focusing on these hot zones will alleviate computational complexity and achieve lightweight computing. Then, a multitask graph convolution network (GCN) is proposed to determine the inherent trend between the departure time and travel cost. Specifically, we design a multitask learning approach to extract potential relations of the aforementioned tasks and improve prediction accuracy [34], [35], [36], [37], [38]. Overall, the blockchain-enabled DeepICE model can be utilized to improve private car users' commute experience and provide privacy, thereby facilitating related services, for instance, alleviating traffic congestion and decreasing pollutant emissions.

This work makes the following contributions.

1) To the best of the authors' knowledge, this work is the first to predict the departure time and travel cost to efficiently improve private car users' commute experience.
2) We propose an end-to-end model DeepICE to improve private car users' commute experience, in which deep learning methods are developed to predict private car users' departure time and travel cost, because the commute experience is closely related to the departure time and the travel cost.
3) In the prediction part of DeepICE, we propose a multitask GCN to simultaneously learn two tasks, i.e., the proper departure time and the expected travel cost, which is able to fully exploit the potential spatial and temporal features to improve prediction accuracy and enhance private car users' commute experience.
4) Extensive experiments are conducted based on a real-world private car trajectory data set. The results demonstrate the superiority of our approach; most of the experimental results are better than the baselines in both single-task learning and multitask learning.

The remainder of this article is organized as follows: Section II presents the related works. Section III introduces the definitions and the problem formulation. Section IV presents an overview of the proposed method and illustrates its details. Next, Section V describes the experimental results based on private car trajectory data. Finally, Section VI concludes this article.

II. Related Works

A. Commute Experience

As commutes are becoming routine in most people's lives, several recent works have focused on studying the experience of personal commutes. For example, Nunes et al. [39] investigated how to leverage passenger knowledge to enhance the travel experience. The authors aim to enhance the travel experience of people who usually take urban public transportation. Thus, the authors propose a framework that unifies their collective intelligence through mobile computing devices and dedicated Web services. They strive to intensify win-win relationships between public transport passengers and operators.

Bonera et al. [40] developed travel experience indices for evaluating travel experience quantitatively. In addition, they aim to identify relationships between objective factors and users' perceptions and pinpoint similarities and differences between the two contexts. This work conducts a comparative investigation of two different contexts, Bristol (U.K.) and Brescia (IT), by analyzing the quality of time spent on board urban buses.

Zhu et al. [41] proposed offering a path planning method based on urban traffic big data. Because a real-time path planning system between the server, public vehicles, and passengers is indispensable for improving transportation efficiency, the authors increase real-time performance by restricting search places for each public vehicle through the managing process. Doing so can alleviate the computational pressure and thus increase the real-time performance of path planning systems. However, the system cannot be utilized to truly improve passengers' commuting experience. In the busy commute period, passengers must bear tremendous commute stress. Furthermore, private car users tend to choose the same commute route, regardless of traffic conditions or external factors [5], [12]. For these reasons, path planning or route optimization methods are not suitable for improving private car users commute experience.

B. Travel Cost Prediction

As we discussed in Section I, travel time is the major concern related to travel cost. There are three main approaches to travel cost prediction: 1) OD-based prediction; 2) segment-based prediction; and 3) path-based prediction.

1) OD-Based Prediction Approach: One approach to travel time prediction is origin-destination (OD) prediction, which is suitable for trajectory data with only origins and destinations. In [42], a nearest neighbor-based method for OD travel time estimation is proposed. This approach estimates the travel time by averaging the scaled travel times of all historical trips with a similar origin, destination, and time of day. However, this approach neglects the underlying spatial structures and the spatial relations of the place that the user passes through, which could lead to inaccurate prediction.

2) Segment-Based Prediction Approach: Segment-based prediction requires full trajectory data with all segments. Wang et al. [43] forecast vehicle speed from the loop sensor and then forecast travel time on individual road segments. In [44], trajectory data are collected via GPS receivers equipped on cars. The travel speed of individual cars on road segments at time t is collected in the GPS so that the travel time through these segments can be predicted. However, due to the low sampling rate, the trajectory data of these cars are sparse, and it is difficult to cover all road segments that vehicles pass through under the low sampling rate and sparse trajectory data, thereby leading to unsatisfactory prediction performance.

3) Path-Based Prediction Approach: Another approach to travel time prediction is path-based prediction. Because the time spent on intersections is not negligible in prediction, existing research considers this aspect [45]. However, these works do not directly use subpaths to estimate travel costs. Subpaths were first proposed in [46] to improve travel cost prediction. Fu et al. [47] further improved the path-based method by mining frequent patterns between trajectories. Nevertheless, with scarce trajectory data that have only origins and destinations, this approach is not applicable. Furthermore, it demands great computational complexity to support the prediction tasks.

Due to the complex spatial and temporal relations between trajectory data, binding such relations to improve the prediction is essential. However, these approaches cannot capture these features well to obtain a better result. Furthermore, because trajectory data are always sparse, they need to balance the coverage of queries and the accuracy of the travel time, which are quite time consuming and memory consuming. Hence, an efficient way to predict travel costs is needed.

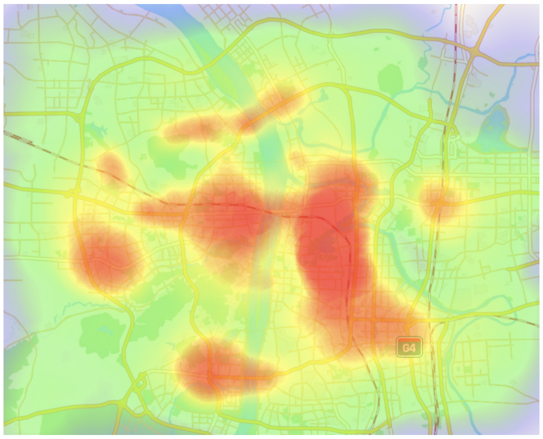


Fig. 2. "Hot zones" on 20 June 2023, in Changsha, China. 9:00 A.M.

III. Preliminaries

In this section, we first present some definitions in this work followed by the problem formulation.

A. Definitions

Trajectory Data: The trajectory data, tra $= \{p_1, p_2, \ldots, p_m\}$, are composed of consecutive historical GPS points. The ith trajectory point p_i is represented by the latitude (lat_i), longitude (lng_i), and timestamp (t_i). m denotes the number of trajectory points.

The trajectory trip contains essential information, such as the start time (*StartTime*), the stop time (*StopTime*), and the corresponding car's ID (*ObjectID*). Table I gives an example of a trajectory trip. Table II presents an example of trajectory data.

Hot Zones Network: From Fig. 2, we observe that when people use their private cars for commuting, these private car users (i.e., noncommercial and nonoperational vehicles, which are only used for private commuting) tend to choose a fixed route for their commutes [12], and they spend more time passing through urban hot zones. Accordingly, their travel cost is highly dependent on the hot zones that they drive through. Focusing on the computing of hot zones leads to lightweight computing. In addition, the trajectory data of private cars are sparse due to their stop-and-wait nature [5].

Considering the sparsity of trajectory data and the specific relations between private car users and hot zones, we construct a hot zone network Z to alleviate the sparsity of private car trajectory data and avoid burdensome computing. In particular, the trajectory data are collected by private car users sharing their trajectory trips. In such data sharing, the blockchain with the consensus mechanism [48] is used to establish trust among untrusted private car users.

In doing so, the sparsity of the overall trajectory data is scattered to these hot zones. Accordingly, the computing complexity can be reduced. Specifically, we first extract hot zones in the city based on the private car trajectory data through the ST-DBSCAN algorithm [49]. Then, each hot zone serves as a node z. If two hot zones, e.g., z_i and z_j, are geographically adjacent, they are interrelated. The hot zone network is represented as $Z = (z, E)$, where z is a set of hot zones (z_1, z_2, ..., z_n) and n denotes the number of extracted hot zones. E denotes a set of edges, indicating whether there are relations between hot zones. The element of E is described as follows:

$$e = \begin{cases} 0, \text{ if there are no relations between hot zones} \\ 1, \text{ if there are relations between hot zones} \end{cases} \qquad (1)$$

where the existence of relations between hot zones means that the minimum spatial distance of trajectory points in hot zone z_i and in hot zone z_j is less than the threshold θ. If the minimum spatial distance is more than the threshold value θ, there is no relation between the two hot zones.

In addition, the spatial distance is calculated as follows:

$$\text{distance} = \sqrt{(\text{lat}_1 - \text{lat}_2)^2 + (\text{lon}_1 - \text{lon}_2)^2} \qquad (2)$$

where lat_1 and lat_2 denote the latitudes of the two trajectory points, and lon_1 and lon_2 are the longitudes of the two points.

Travel Cost: Travel cost [10], for example, travel speed or travel time, could imply the degree of traffic congestion. According to [18], we observe that private cars spend more time passing through urban hot zones. To alleviate the sparsity of the trajectory data and decrease computational pressure, we choose the travel time spent passing each hot zone as the research object. Note that private car users' overall travel cost relies on the travel cost in hot zones; it is not required to calculate total travel time between the origin and destination. Therefore, by predicting the travel cost for passing through each hot zone, we can obtain the overall travel cost for a private car user.

As shown in Table I, with the input trajectory vector *trip={ObjectID, StartTime, StopTime, StartLon, StopLat}*, we calculate the travel cost for each hot zone

$$\text{travel cost}_i = \text{StopTime}_i - \text{StartTime}_i, i = 1, \ldots, p, \qquad (3)$$

$$\text{travel cost}_j = \left(\sum_{i=1}^{p} \text{travel cost}_i\right) / p \qquad (4)$$

where travel cost_i denotes the travel cost of the ith trajectory record in the hot zone, p denotes the number of trajectory records in the hot zone, and travel cost_j is the travel cost of the jth hot zone.

Feature Matrix: The feature matrix X is composed of different features, such as travel cost and departure time. The calculation process is as follows:

$$X = (X_c, X_d), \qquad (5)$$

$$X_{t+1}, \ldots, X_{t+T} = f_\theta\left(\mathcal{G}; X_{t-T'+1}, \ldots, X_t\right) \qquad (6)$$

where X_c denotes the travel cost, X_d is the departure time, and θ is a tunable parameter. Furthermore, because the original formulations of X_c and X_d are heterogeneous, for example, the original X_d is represented in date format (8:30 A.M. However, X_c is the travel time (e.g., 30 min); hence, we transform the original X_d into the nondata format, i.e., 8:30 A.M. is transformed into 8.5, thereby decreasing the heterogeneity of the input data X_c and X_d.

TABLE I
TRAJECTORY TRIP IN THE DATA SET

ObjectID	StartTime	StopTime	StartLon	StartLat	StopLon	StopLat
130399	2020-03-07 08:06:54	2020-04-07 08:16:00	112.559135	28.254652	112.549917	28.274228
130399	2020-04-12 08:32:06	2020-04-12 09:57:55	112.923892	28.212222	112.980295	28.177803
130399	2020-05-11 07:47:43	2020-05-11 08:22:15	112.91322	28.227385	113.007045	28.232185

TABLE II
TRAJECTORY DATA IN THE DATA SET

ObjectID	GPSTime	Lon	Lat
180915	2020-03-05 08:10:34	113.029168	28.204063
180915	2020-04-10 07:22:13	113.014115	28.223977
180915	2020-05-20 08:01:00	113.056863	28.203608

B. Problem Formulation

On collecting large-scale private cars' trajectory data, historical traffic data, for example, travel cost and departure time, can be denoted as a feature matrix $X \in R^{N\times T'}$, where T' is the number of historical time periods. Hence, this problem can be regarded as learning a function f with the feature matrix $X \in R^{N\times T'}$ and hot zone network $H = (h, E)$ to forecast the future travel cost and departure time in the next T time periods. The input feature matrix of the problem is as follows:

$$\begin{aligned}&\Big(\Big(x_1^{t-T'+1}, x_1^{t-T'+2}, \ldots, x_1^{t}\Big)\Big)\\ &\Big(x_2^{t-T'+1}, x_2^{t-T'+2}, \ldots, x_2^{t}\Big)\\ &\Big(x_3^{t-T'+1}, x_3^{t-T'+2}, \ldots, x_3^{t}\Big)\\ &\ldots\\ &\Big(x_n^{t-T'+1}, x_n^{t-T'+2}, \ldots, x_n^{t}\Big)\Big]^T.\end{aligned}$$

IV. METHODOLOGY

To improve private car users' commute experience, we propose a blockchain-enabled multitask learning model named DeepICE, which consists of three components as follows.

1) Preprocessing.
2) Spatial and temporal feature modeling.
3) Blockchain-enabled multitask learning.

Fig. 3 illustrates the framework of the proposed DeepICE model. First, in the preprocessing component, we construct the hot zone network (see *Definitions* in Section III-A) by extracting the hot zones from weekday trajectory data. Next, we model the spatial features and temporal features through DL-based methods. Finally, we utilize multitask learning to jointly predict travel cost and departure time. Algorithm 2 presents a detailed processing of the proposed model. See Table III for formulas and definitions.

A. Preprocessing

In the preprocessing of DeepICE, we filter trajectory records whose travel time is less than 5 min and whose departure time is not 6:30 A.M. Note that we strive to improve the commute experience of private car users; hence, we select private cars' trajectory data on workdays (excluding weekends and holidays). Notably, private car users share their trajectory data in a distributed way, and hence, privacy issues emerge. To solve this problem, the blockchain with a consensus mechanism is employed during trajectory acquisition to improve user privacy.

TABLE III
NOTATIONS AND DEFINITIONS

Notation	Definition
X	Feature matrix
X_c	The feature matrix of the travel cost
X_d	The feature matrix of the departure time
$\boldsymbol{U}$	The graph Fourier basis
D	The diagonal degree matrix.
W	Weight matrix
δ	The activation function.
h_{t-1}	The hidden state at time t-1
x_t	The traffic data at t
h_t	The output state at time t
$\tilde{\boldsymbol{h}}_t$	The momentary state at time t.
b_r, b_u, b_h	The deviations of each variable.
$L_c(\theta)$,$L_d(\theta)$	The prediction loss functions.
$\widetilde{Y}_c$	The prediction result of the travel cost.
$\widetilde{Y}_d$	The prediction result of the departure time.
δ_c, δ_d	The weighted parameters.

After classifying the filtered trajectory data into five types according to different workdays, we then extract the hot zones for the workdays. To that end, we utilize the ST-DBSCAN algorithm to extract the hot zones [49]. Based on the trajectory data, Algorithm 1 presents a detailed processing of ST-DBSCAN for extracting urban hot zones. ST-DBSCAN is able to discover clusters according to nonspatial, spatial, and temporal values of the objects [50]. The algorithm uses two distance parameters, Eps1 and Eps2, to measure the similarity of the spatiotemporal trajectory data. Eps1 is used to measure the spatial distance, and Eps2 is applied to calculate the temporal distance. For example, for the points $P(\text{Lon}1, \text{Lat}1, t1, t2)$ and $Q(\text{Lon}2, \text{Lat}2, t3, t4)$, we compute Eps1 and Eps2 as follows:

$$\text{Eps } 1 = \text{Euclidean}(\text{Lon}1, \text{Lat}1, \text{Lon}2, \text{Lat}2) \quad (7)$$

$$\text{Eps } 2 = \sqrt{(t_3 - t_1)^2 + (t_4 - t_2)^2}. \quad (8)$$

B. Spatial and Temporal Feature Modeling

In this section, we implement the feature modeling component of DeepICE. Specifically, we model the spatial features of the extracted hot zone network and capture the temporal features of different workdays through a one-layer graph convolution network (GCN) and gated recurrent unit (GRU).

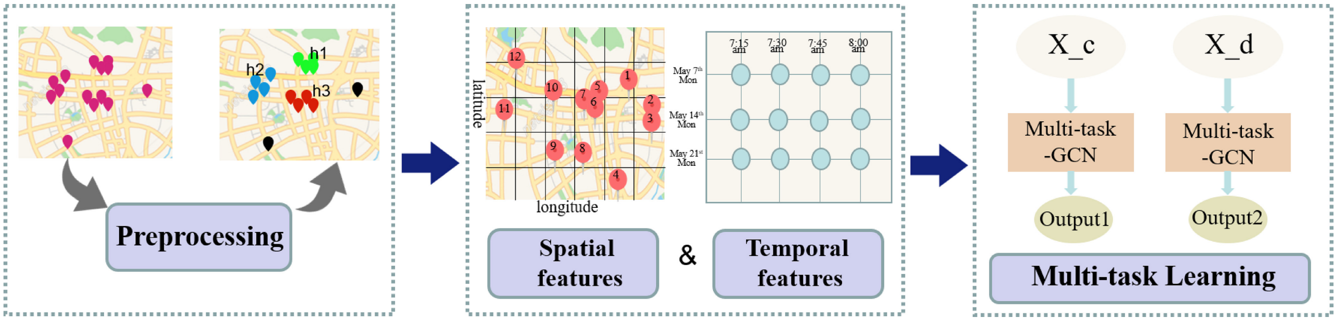


Fig. 3. Overview of deepICE.

1) Spatial Feature Modeling: Traditional DL methods, such as convolutional neural networks (CNNs) and fully connected networks can obtain the spatial features of regular structures. For example, they can model the features based on the regular grid structure [51]. However, they cannot exploit irregular structures to properly represent task dependencies because the spatial dependencies between hot zone networks are non-Euclidean. To tackle this problem, we propose utilizing graph structures to model the spatial features of the spatial hot zone network. To that end, we first introduce the general concept of the graph convolution neural network-graph operator

$$o = (f * Gg)_\theta = U g_\theta U^T f \tag{9}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is the graph Fourier basis that represents the matrix of eigenvectors of the normalized graph Laplacian matrix $L = I_n - D^{-(1/2)} A D^{-(1/2)} = U \Lambda U^T \in \mathbb{R}^{n\times n}$. I_n denotes an identity matrix, and $D \in \mathbb{R}^{n\times n}$ is the diagonal degree matrix. $\Lambda \in \mathbb{R}^{n\times n}$ is the diagonal matrix of eigenvalues of L.

Note that this computation is expensive due to the (n^2) complexity with the graph Fourier basis U. To resolve this problem, we introduce a more efficient spectral graph convolution as follows:

$$o = \zeta\left(I_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)x = \zeta\left(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}\right)x \tag{10}$$

where $\zeta \in \mathbb{R}^{\ltimes}$ is a vector of parameters in the Fourier domain, $\widetilde{A} = A + I_n$, $\widetilde{D} = \sum_{i=1} \widetilde{a}_{jj}$.

Upon extracting the graph operators o, we then obtain the output of the GCN

$$\tilde{\boldsymbol{Y}}_t = \delta\left(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}\right) XW \tag{11}$$

where X denotes the feature matrix, W is the weight matrix, and δ is the activation function. In this work, we choose ReLU as the activation function.

2) Temporal Feature Modeling: Note that there are long-term features such as periodicity features in private car trajectories. For example, Mondays' commute trajectories exhibit similar traveling trends [11]. Extracting the features and utilizing them will improve prediction accuracy. To capture these hidden features, we leverage a GRU to capture the task dependencies and long-term features in the temporary dimension [52].

The GRU method, an improved version of the RNN, is able to resolve the problem of gradient disappearance or gradient explosion in the RNN by adding gate structures. Compared to another improvement in RNNs, namely, long short-term memory (LSTM), GRU can realize results equal to or better than LSTM [53], [54], but the GRU, with fewer neurons, can alleviate computational pressure and accelerate computational speed [55]. Fig. 4 illustrates the inner structure of the GRU. In Fig. 4, h_{t-1} represents the hidden state at time t-1; x_t represents the traffic data at t; r_t is the reset gate to control the degree of ignoring the status information in the previous moment; and u_t denotes the update gate to control the degree to which status information at the previous time is brought into the current status. h_t is the output state at time t, and $\tilde{\boldsymbol{h}}_t$ is the momentary state at time t.

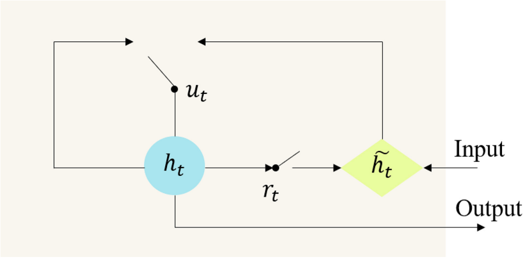


Fig. 4. Inner structure of a GRU.

After obtaining the output of the last layer $\widetilde{Y}_t$, the GRU can transform $\widetilde{Y}_t$ to h_t as follows:

$$\boldsymbol{h}_t = \mathrm{GRU}\left(\boldsymbol{h}_{t-1}, \tilde{\boldsymbol{Y}}_t\right). \tag{12}$$

The details of the GRU process are as follows:

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_r\left[\tilde{\boldsymbol{Y}}_t, \boldsymbol{h}_{t-1}\right] + \boldsymbol{b}_r\right) \tag{13}$$

$$u_t = \sigma\left(W_u\left[\tilde{\boldsymbol{Y}}_t, h_{t-1}\right] + b_u\right) \tag{14}$$

$$\tilde{\boldsymbol{h}}_t = \tanh\left(\boldsymbol{W}_h\left[\tilde{\boldsymbol{Y}}_t, (\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})\right] + \boldsymbol{b}_h\right) \tag{15}$$

$$h_t = u_t \odot h_{t-1} + (1 - u_t) \odot \tilde{\boldsymbol{h}}_t \tag{16}$$

where W_r, W_u, and W_h denote the weight matrices, b_r, b_u, and b_h are the deviations of each variable, and σ and $\odot$ are the activation functions and a multiplication operator, respectively.

C. Blockchain-Enabled Multitask Learning

During people's commute, travel cost and departure time are closely connected and influence each other. For instance, an earlier departure time will lead to a lower travel cost,

Algorithm 1 ST-DBSCAN for Extracting Hot Zones

```
Input: D={$p_1, p_2, \ldots, p_m$}, denotes the set of trajectory points.
        Eps1: Maximum spatial distance value.
        Eps2: Maximum temporal distance value.
        MinPts: Minimum number of trajectory points within the Eps1 and Eps2 distances.
        $\Delta\varepsilon$: Threshold value to be included in a cluster.
Output: $C = \{C_1, C_2, \ldots, C_k\}$ set of $k$ clusters
Cluster_Label=0
for all $i$=0 to m do
    if $p_i$ is not a cluster then
        X=Retrieve_Neighbors($p_i$, Eps1, Eps2);
        if |X| < MinPts then
            Mark $p_i$ as noise;
        else
            Cluster_Label=Cluster_Label+1;
            for all $j$=1 to |X| do
                Mark all trajectory points in X with current Cluster_label
            end for
            Push {all trajectory points in X}
            while not Empty do
                CurrentPoints=Pop()
                Y=Retrieve_Neighbors(CurrentPoints, Eps1, Eps2)
                if |Y| > MinPts then
                    for all  trajectory points p in Y do
                        if ($p$ is not marked as noise or it is not in a cluster) and
                            $|Cluster\_Avg() - p.Value| <= \Delta\varepsilon$ then
                            Mark $p$ with current Cluster_label;
                            Push($p$)
                        end if
                    end for
                end if
            end while
        end if
    end if
end for
```

Algorithm 2 DeepICE Model

```
Input: Historical track data $S$ and $r_h$.
    $S = \{ObjID, StartTime, StopTime, StartLon, StartLat, StopLon, StopLat\}$.
Output: Predicted travel cost $Y_d$ and departure time $Y_c$ of each hot zone at the next moment.
Initialize weight matrix parameters and bias terms: $W_r, W_u, W_h, b_r, b_u, b_h, b_C$.
Calculate the travel cost and average departure time through each hot zone based on $S$ and $r_h$.
Construct the hot zone matrix $X_Z$ according to Algorithm 1.
Calculate the adjacency matrix $E$ of the hot zone matrix according to Equation (1).
Construct a 2-D matrix $X_d$ based on average departure time and a 2-D matrix $X_c$ based on average speed as follows:
    $$X_c = \left[x_{ci}^{t-T'+1}, x_{ci}^{t-T'+2}, \ldots, x_{ci}^{t}\right]^T, i = 1, \ldots n$$
    $$X_d = \left[x_{di}^{t-T'+1}, x_{di}^{t-T'+2}, \ldots, x_{di}^{t}\right]^T, i = 1, \ldots n$$
while $i \leq$ total number of hot zones do
    for each hot zone do
        Calculate $Y_d$ and $Y_c$ according to Equations (11)–(19)
    end for
end while
```

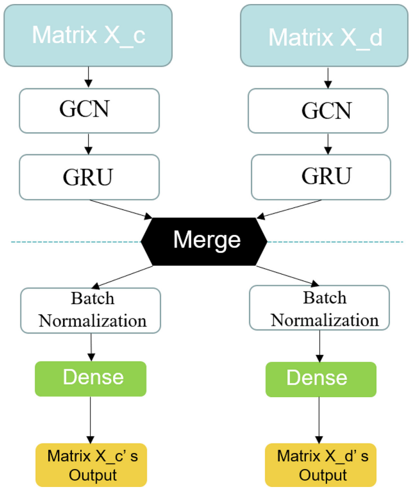


Fig. 5. Multitask GCN.

and the nearer the departure time is to the peak hour, the higher the travel cost will be. There are hidden task dependencies between departure time and travel cost. Extracting these dependencies and achieving representation sharing will help to improve prediction and private car users' commute experience. Accordingly, we utilize a multitask learning framework to jointly predict travel cost and departure time. Multitask learning can reveal the latent features in tasks and benefit each individual task [56]. The underlying shared representation can be captured by multitask learning to improve the overall prediction [27], [57], [58]. Different tasks are treated as interactive activities, making the obtained information more robust than that gained from one single task. Recalling spatial and temporal feature modeling (see Section IV-B), we propose a multitask GCN to better model and predict the travel cost as well as the departure time. Fig. 5 details the construction of the multitask GCN.

Furthermore, we design a hard parameter sharing method to jointly predict the travel cost and the departure time because hard parameter sharing is effective for addressing tasks with strong correlations. In particular, we utilize blockchain technology to address the privacy concerns of private car users during the multitask learning process. In so doing, each private car user can learn the spatial and temporal connection from the trusted trajectory data shared by other users. As such, it is suitable for our prediction tasks, i.e., predicting the proper departure time and the expected travel cost, and the privacy issues of private cars are properly handled. Specifically, this method can be applied to all hidden layers of all tasks, which is suitable for tasks with strong correlations, such as departure time prediction, and retaining the output layer related to the task.

The goal of the two prediction tasks is to minimize the prediction loss functions $L_c(\theta)$ and $L_d(\theta)$, which can be represented as follows:

$$L_c(\theta) = \left(\widetilde{Y}_c - X_c\right)^2 \tag{17}$$

$$L_d(\theta) = \left(\widetilde{Y}_d - X_d\right)^2. \tag{18}$$

Here, $\widetilde{Y}_c$ denotes the travel cost prediction result, X_c denotes the ground truth of the travel cost, $\widetilde{Y}_d$ is the departure time prediction result, and X_d is the real value of the departure time. The proposed DeepICE utilizes multitask learning in the prediction, and the target is transformed to jointly predict two tasks and minimize the comprehensive loss function as follows:

$$L(\theta) = \delta_c\left(\widetilde{Y}_c - X_c\right)^2 + \delta_d\left(\widetilde{Y}_d - X_d\right)^2 \tag{19}$$

where δ_c and δ_d are the weighted parameters. We set δ_c and δ_d to 0.5 in the experiment.

V. Experiments

A. Data Set

Private Car Trajectory Data: In previous work, we utilized vehicle positioning methods [59], [60] in real-world IoV scenarios and implemented trajectory collection in real-world urban environments. The blockchain technology is used to secure vehicle-to-everything (V2X) communication so that trajectory data collection is a trusted task. In other words, private car users can trust others' trajectory data. At present, we have obtained large-scale private car trajectory data (see examples in Tables I and II). For more details of the trajectory data set, please refer to [61].

In the experiments, we select the private car trajectory data set from Yuhua District of Changsha City, China. The experimental period is sampled from 1 May 2020 to 31 August 2020. The overview of our data set is provided in Table IV. From the original data, we first extract the travel cost and the departure time of each trajectory record. Considering that our experiments are based on different workdays' hot zone distributions, we cluster five workdays' trajectories into several hot zones by the ST-DBSCAN algorithm (see Section IV-A). As presented in Table V, the spatial threshold Eps1 is set as 500 m, and θ is set to 2Eps1. The temporal threshold Eps2 is set as 15 min, and the minimum neighbors are set as 3. For instance, Fig. 6 depicts the hot zone network of Monday, and there are 14 hot zones in the selected region in Yuhua District on Mondays between May 1st and August 31st. The experiments are conducted based on the MindSpore framework platform.

TABLE IV
Overview of Private Car Trajectory

DATASET	Private Car Trajectory
City	Changsha, China
District	Yuhua District
* Longitude Span	112.5730-113.063
* Latitude Span	28.023-28.113
* Training Span	5/1/2020-7/31/2020
* Testing Span	8/1/2020-8/31/2020

TABLE V
Number of Hot Zones of Workdays

WORKDAY	Mon.	Tue.	Wed.	Thu.	Fri.
The number	14	15	15	15	15

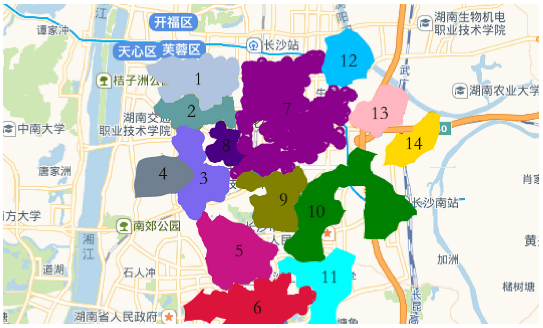


Fig. 6. Hot zone network for Monday.

B. Experimental Environment

1) Evaluation Metrics: We choose the following metrics to evaluate the performance of the proposed DeepICE and the baselines: root mean squared error (RMSE) and mean absolute error (MAE). The following formulations are defined:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widetilde{y}_i - y_i)^2} \tag{20}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\widetilde{y}_i - y_i| \tag{21}$$

where $\widetilde{y} = \{\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_n\}$ denotes the predicted value of the proposed model, $y = \{y_1, y_2, \ldots, y_n\}$ denotes the ground truth, and n is the number of testing samples.

2) Baselines: Departure time and travel cost are jointly predicted by the proposed DeepICE. We evaluate the proposed method with two types of baselines: 1) single-task prediction and 2) multitask prediction.

Single-Task Prediction: We introduce the single-task prediction as follow:

1) *History Average (HA):* This model is a basic statistical model based on historical values, predicting future values by averaging the values of corresponding historical periods [62].
2) *Autoregressive Integrated Moving Average (ARIMA):* It is a prediction model based on data analysis, forecasting the data in a given time series based on past values [63].
3) *LSTM:* It is a time loop neural network. Due to its unique design structure, LSTM is suitable for processing and predicting important events with very long intervals and delays in the time series [64].

4) *BO-Support Vector Recurrent (BO-SVR):* It is an improvement in SVR and can be utilized to predict the time series problem [65].
5) *The Diffusion Convolutional Recurrent Neural Network (DCRNN):* It is a neural network that can maximize the probability of generating target future time series by using temporal backpropagation, thereby capturing the spatial and temporal correlations between time series [66].
6) *The Temporal Graph Convolutional Network (T-GCN):* It is combined with the GCN and GRU, and the GRU is used to learn dynamic changes in traffic data to capture temporal dependence [67].
7) *The Graph Convolutional Network (GCN):* It is a neural network that operates on graphs. It can handle non-Euclidean data and is widely used for traffic prediction [68].

Multitask Prediction: We introduce the multi-task prediction as follow:

1) *3S-FCN:* This method is based on a fully connected network and multitask learning, modeling the temporal correlations and capturing the spatial correlations via multiple convolutions simultaneously. Hence, it can predict two tasks in the meantime [25].
2) *Multi-CNN:* This model has two CNN layers sharing a similar multitask architecture and can be used to predict several tasks [69].
3) *DeepICE:* Our proposed model can capture the temporal and spatial features of different tasks and then utilize these features to predict and improve accuracy.

C. Experiments and Results

1) Performance Comparison: Table VI lists the experimental results on travel cost prediction, and Table VII presents the comparison results of predicting the departure time only. Table VIII presents the performance of baselines on predicting two tasks simultaneously. All the comparisons are evaluated by the metric RMSE. From these results, we observe the following.

1) Compared with the baselines, our proposed DeepICE achieves the best performance in terms of one-task prediction and two-task prediction.
2) Basic statistical models, such as HA [62] and ARIMA [63] perform worse than other baselines because these two methods lack the ability to capture the underlying spatial and temporal features.
3) DL-based models, such as GCN [68], LSTM [64], BO-SVR [65], DCRNN [66], and T-GCN [67] perform better than basic statistical models because DL networks can capture temporal features. However, they cannot connect two tasks to achieve feature sharing, so their performance deteriorates significantly compared to that of DeepICE.
4) Compared to multitask models, such as 3s-FCN [25] and Multi-CNN [69], our proposed DeepICE obtains better results. The reasons behind this result stem from importing GCN as the spatial feature extractor to model historical data at a high level, while the other two models fail to represent the spatial factors in the prediction task. Therefore, even though the concept of multitask learning is utilized in these two models, they show poorer performance.

TABLE VI
PERFORMANCE ON DEPARTURE TIME PREDICTION (RMSE)

Models	Mon.	Tue.	Wed.	Thu.	Fri.
HA	5.7145	6.112	6.836	5.532	6.301
ARIMA	6.2988	5.216	5.486	6.0012	5.784
LSTM	4.91	4.508	4.613	3.985	4.037
BO-SVR	4.716	4.119	5.226	4.203	4.735
DCRNN	4.522	3.914	4.449	3.654	3.692
GCN	3.723	3.152	3.001	2.429	2.523
T-GCN	2.953	2.583	2.658	2.231	2.439
DeepICE	**0.4631**	**0.1326**	**0.4148**	**0.6656**	**0.0895**

TABLE VII
PERFORMANCE ON TRAVEL COST PREDICTION (RMSE)

Models	Mon.	Tue.	Wed.	Thu.	Fri.
HA	8.4427	9.024	9.023	8.875	9.754
ARIMA	9.98	7.456	8.496	8.657	8.237
LSTM	6.593	6.777	6.023	6.872	6.254
BO-SVR	5.663	5.78	7.842	5.138	7.99
DCRNN	4.935	5.01	5.999	4.762	6.211
GCN	4.003	4.265	5.12	4.365	5.86
T-GCN	3.795	3.886	4.663	4.177	5.134
DeepICE	**2.4968**	**2.4798**	**3.239**	**2.1716**	**2.5592**

TABLE VIII
PERFORMANCE ON TWO-TASK PREDICTION (RMSE)

Models	Mon.	Tue.	Wed.	Thu.	Fri.
3S-FCN	6.975	7.36	6.281	7.14	6.839
Multi-CNN	4.0328	4.965	4.2576	5.4007	3.485
DeepICE	**2.9599**	**2.6154**	**3.6538**	**2.8372**	**2.6487**

Furthermore, we compare the proposed DeepICE with baselines from the perspective of MAE (%). The experimental results are presented in Figs. 7–9. We observe the following.

1) The proposed DeepICE obtains the best results in not only the RMSE as the evaluation metric but also the MAE as the evaluation metric. These results demonstrate the superiority of DeepICE over the baselines.
2) The prediction of travel cost achieves a lower accuracy than the departure time prediction in all the methods because the departure time fluctuates less than the travel cost. Furthermore, because the departure time is fixed in a time frame (6:30 A.M. to 9:00 A.M.), it is transformed into a value range of 6.5 to 9.0, but the value of travel cost has no range; it changes through the different time and spatial dimensions. Accordingly, it is more difficult to extract the spatial and temporal features in the travel cost prediction task than in the departure time prediction task.
3) Due to the relations between travel cost and departure time, our proposed DeepICE realizes hard parameter sharing and network sharing between these two tasks. In addition, by introducing a GRU and GCN into DeepICE, the spatial and temporal features can be captured efficiently.

2) Effect of Each Component in DeepICE: To evaluate the effect of the DeepICE components, we test performance

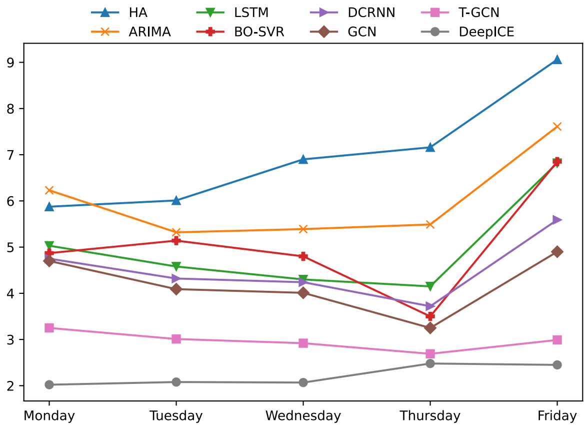


Fig. 7. Travel cost prediction performance (MAE).

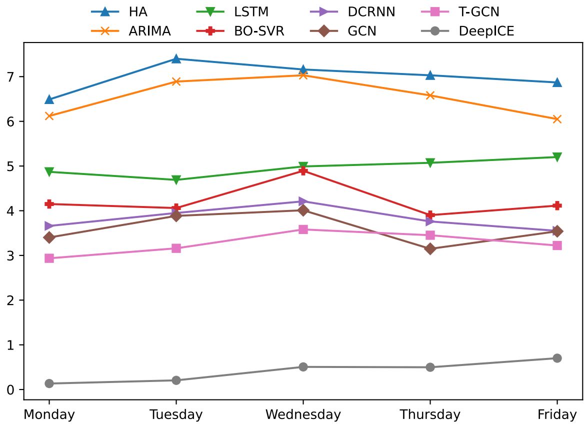


Fig. 8. Departure time prediction performance (MAE).

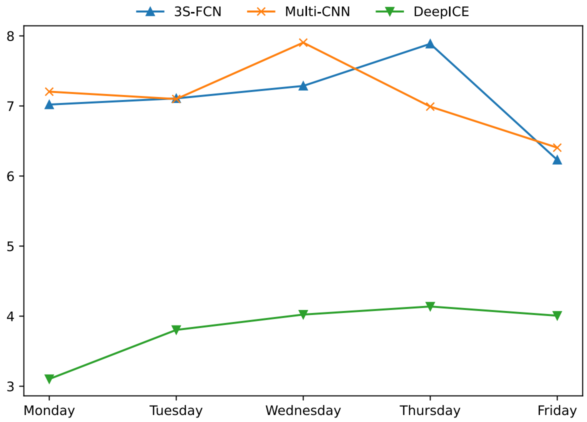


Fig. 9. Performance comparison on two-task prediction (MAE).

TABLE IX
PERFORMANCE OF EACH COMPONENT ON TWO-TASK PREDICTION (RMSE)

Variants	Mon.	Tue.	Wed.	Thu.	Fri.
Deep-nohot	5.7984	5.921	6.347	6.272	6.21
Deep-noGCN	4.3872	4.0279	4.7326	3.9981	4.5292
Deep-noGRU	5.34	4.8194	4.6902	5.4751	5.114
Deep-nomulti	4.446	4.325	5.648	5.789	4.258
DeepICE	**2.9599**	**2.6154**	**3.6538**	**2.8372**	**2.6487**

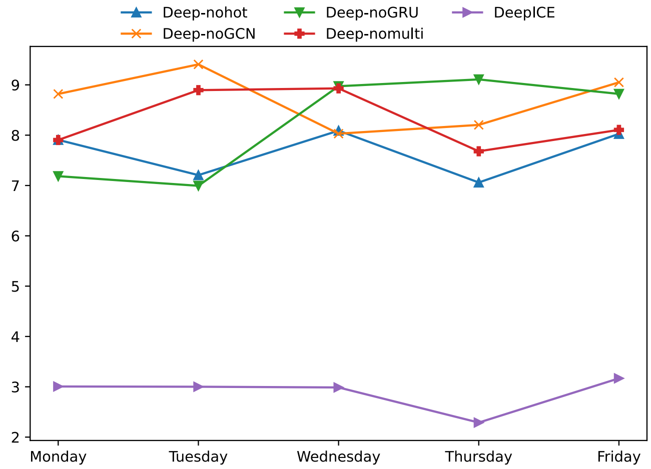


Fig. 10. Performance of each component on two-task prediction (evaluation metric: MAE).

individually by removing each component. As presented in Section IV, the proposed DeepICE consists of four components.

1) *Preprocessing: a*) the extraction of hot zones.
2) *Spatial and Temporal Feature Modeling: b*) the importing of GCN; *c*) the importing of GRU.
3) *Multitask Learning: d*) the importing of multitask learning.

Thus, there are four variants of DeepICE as follows.

1) *Deep-nohot* indicates that there is no “hot zone” extraction (i.e., without *a*).
2) *Deep-noGCN* indicates that the GCN is removed from the proposed method (i.e., without *b*).
3) *Deep-noGRU* indicates that the GRU is removed from the proposed method (i.e., without *c*).
4) *Deep-nomulti* means not using multitask learning (i.e., without *d*).

We implement experiments on two-task prediction. Table IX compares the performance of these variants with that of DeepICE. The evaluation metric is RMSE. Fig. 10 depicts the comparison results for the joint prediction of travel cost and departure time, with MAE serving as the evaluation metric. From Table IX and Fig. 10, we find the following.

1) The proposed DeepICE outperforms all the variants, illustrating the indispensability of each component in the overall model.
2) Each component contributes to the prediction task, especially the extractions of hot zone functions, because hot zone extraction significantly influences spatial feature extraction.
3) With multitask learning being removed, the RMSE of Deep-nomulti increases slightly compared to the other variants of DeepICE. This result implies that the influence of multitask learning is less than that of the other components. In other words, multitask learning can play an essential role in improving the prediction results, given that the temporal and spatial features are properly modeled.

3) Parameter Sensitivity Analysis: To investigate the efficiency of the parameter setting of the proposed DeepICE, we conduct two-task prediction to evaluate the sensitivity of the hyperparameters based on the number of training epochs

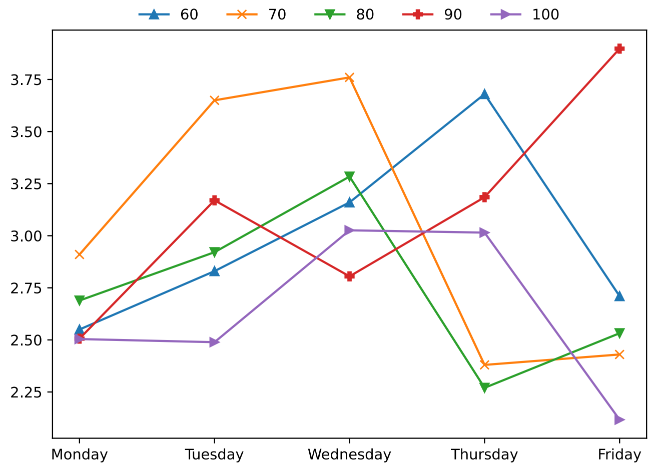


Fig. 11. Parameter Sensitivity: number of epochs (evaluation metric: MAE).

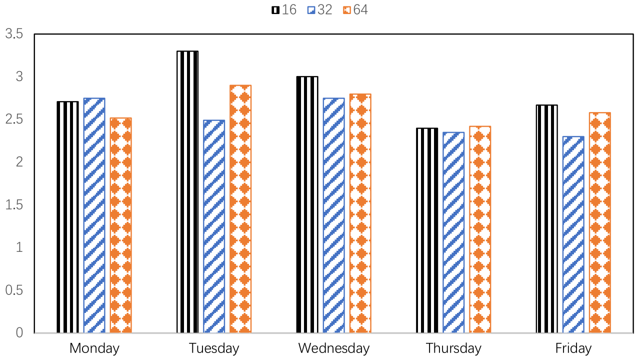


Fig. 12. Parameter Sensitivity: number of GRUs (evaluation metric: RMSE).

TABLE X
COMPUTATION PLATFORMS USED BY DIFFERENT CAR BRANDS

Brand	Model	SOC	GFLOPS
Tesla	Model 3	AMD Ryzen	10000
Cadillac	LYRIQ	Qualcomm	1843
Benz	EQS	Nvidia	845
BWM	7 Series	Intel	216

and the number of GRUs. The training epochs vary in the range {60, 70, 80, 90, 100}, and the number of GRUs varies in the range {16, 32, 64}.

Figs. 11 and 12 show comparison results for the number of epochs and the number of GRUs, respectively. When other parameters are set to their defaults, RMSE decreases significantly as the number of training epochs increases on most workdays. In addition, various trends in prediction tasks emerge on different workdays. In particular, the proposed model achieves the best results, namely, it obtains the lowest MAE on most workdays when the number of training epochs is 100. Hence, we set the number of training epochs to 100 in the experiments. Similarly, prediction performance is tightly dependent on the number of GRUs. As shown in Fig. 12, on most workdays, when the number of GRUs is 32, the best performance is achieved. Accordingly, we set the number of GRUs to 32 in our experiments.

Blockchain protocols, by design, involve complex cryptographic operations and consensus algorithms that can impose a significant computational burden. Additionally, the communication overhead of maintaining a distributed ledger can increase network traffic and latency. The runtime performance of GCNs is a crucial aspect to consider when deploying a predictive algorithm in real-world scenarios, particularly in the context of vehicular platforms with limited computational resources. To address concerns about overhead acceptability, further discussion is necessary.

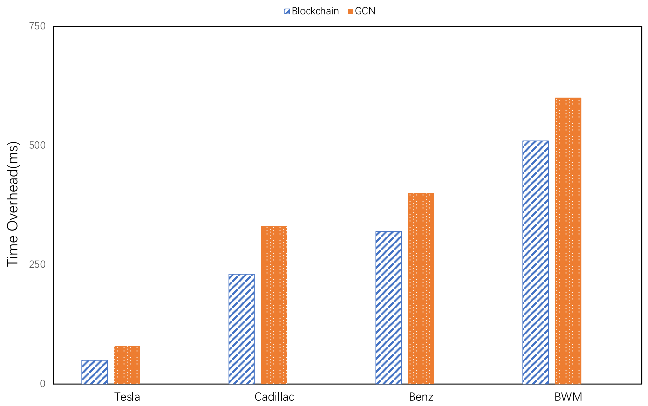


Fig. 13. Time overhead in the blockchain and GCN.

As shown in Table X, Tesla's SOC computing power is up to 10 000 GFLOPS, while BMW's SOC computing power is the lowest at 216. However, Fig. 13 shows that the time overhead of different models is not completely determined by their SOC computing power. For example, the computing power of Cadillac is 118% higher than that of Mercedes-Benz, and the time overhead on the blockchain and GCN is reduced by 38% and 14%, respectively. This result is due to Nvidia's unique computing module, CUDA, which has an advantage in computing. This observation shows that under limited computing resources, an acceptable time cost can be obtained by advanced algorithms. On the other hand, even on the BMW platform, which has the worst time performance, the actual performance of the blockchain and GCN is 512 and 630 ms, respectively, which shows that our algorithm can effectively cope with limited vehicle computing resources.

VI. CONCLUSION

In this article, we propose a blockchain-enabled DL-based model, i.e., DeepICE, to improve private car users commute experience in a privacy-preserving manner. In this model, we jointly predict the travel cost and departure time of private cars. Due to the relations between these two tasks, we designed a multitask GCN architecture to represent the complex features in these two tasks and predict them. Thanks to the introduction of the blockchain technology, the privacy issues of private cars can be properly addressed. The experimental results based on real-world private car trajectories show that our model outperforms all the baseline methods in terms of the evaluation metrics, illustrating the superiority of our model.

In the future, we will devote our efforts to extending our work from the following aspects. We will consider environmental and external factors such as the weather into the multitask learning model. Besides, we will incorporate traffic information into our work to boost the prediction performance, thereby further improving private car users' commute experience.

References

[1] X. Dai et al., “A learning-based approach for vehicle-to-vehicle computation offloading,” *IEEE Internet Things J.*, vol. 10, no. 8, pp. 7244–7258, Apr. 2023.

[2] T. C. P. G. of the People’s Republic of China. “In 2020, private car ownership in China exceeded 200 million, 66 cities have more than one million cars.” 2020. [Online]. Available: http://www.gov.cn/xinwen/2020-01/07/content_5467341.htm

[3] Wikipedia. “List of car ownership per capita by country.” 2020. [Online]. Available: https://en.wikipedia.org/wiki/List_of_countries_by_vehicles_per_capita

[4] Z. Xiao et al., “Understanding private car aggregation effect via Spatio-temporal analysis of trajectory data,” *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2346–2357, Apr. 2023.

[5] 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, Nov. 2018.

[6] Z. Xiao, H. Fang, H. Jiang, J. Bai, V. Havyarimana, and H. Chen, “Understanding urban area attractiveness based on private car trajectory data using a deep learning approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12343–12352, Aug. 2023.

[7] Z. Xiao, H. Xiao, W. Chen, H. Chen, A. C. Regan, and H. Jiang, “Exploring human mobility pattern and travel Behavior: A focus on private cars,” *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 5, pp. 129–146, Sep. 2022.

[8] B. Map. “Commuter monitoring report of national major cities in 2020.” 2020. [Online]. Available: http://huiyan.baidu.com/reports/landing?id=58

[9] X. N. A. Beijing. “Gaode map releases “top 10 most popular ways to work in the country”: Beijing Xierqi and Houchangcun road have the highest commuting enthusiasm.” 2021. [Online]. Available: http://www.xinhuanet.com/2021-01/19/c_1127001747.htm

[10] G. R. Parsons, “The travel cost model,” in *A Primer on Nonmarket Valuation*. Dordrecht, The Netherlands: Springer, 2003, pp. 269–329.

[11] 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, Oct. 2019.

[12] Z. Xiao et al., “On extracting regular travel behavior of private cars based on trajectory data analysis,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14537–14549, Dec. 2020.

[13] F. Yuan, L. D. Han, S.-M. Chin, and H. Hwang, “Proposed framework for simultaneous optimization of evacuation traffic destination and route assignment,” *Transp. Res. Rec.*, vol. 1964, no. 1, pp. 50–58, Jan. 2006.

[14] M. U. S. Khan et al., “MacroServ: A route recommendation service for large-scale evacuations,” *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 589–602, Jul./Aug. 2017.

[15] M. Lujak, S. Giordani, and S. Ossowski, “Route guidance: Bridging system and user optimization in traffic assignment,” *Neurocomputing*, vol. 151, pp. 449–460, Sep. 2015.

[16] P. Li, P. Mirchandani, and X. Zhou, “Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork,” *Transp. Res. Part B: Methodol.*, vol. 81, pp. 103–130, Sep. 2015.

[17] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, Y. Xia, and S. Wang, “Solving trajectory optimization problems in the presence of probabilistic constraints,” *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4332–4345, Oct. 2020.

[18] D. Xue, A. Cook, and C. Tisdell, “Biodiversity and the tourism value of Changbai mountain biosphere reserve, China: A travel cost approach,” *Tourism Econ.*, vol. 6, no. 4, pp. 335–357, Dec. 2000.

[19] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, “Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction,” *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 5668–5675, 2019.

[20] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, and Z. Li, “Modeling spatial-temporal dynamics for traffic prediction,” Jun. 2018, *arXiv:1803.01254*.

[21] X. Zhou et al., “Hierarchical federated learning with social context clustering-based participant selection for Internet of Medical Things applications,” *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 4, pp. 1742–1751, Aug. 2023.

[22] P. Li, M. Abdel-Aty, and J. Yuan, “Using bus critical driving events as surrogate safety measures for pedestrian and bicycle crashes based on GPS trajectory data,” *Accid. Anal. Prev.*, vol. 150, Dec. 2021, Art. no. 105924.

[23] Q. Chen, R. Gu, H. Huang, J. Lee, X. Zhai, and Y. Li, “Using vehicular trajectory data to explore risky factors and unobserved heterogeneity during lane-changing,” *Accid. Anal. Prev.*, vol. 151, Dec. 2021, Art. no. 105871.

[24] X. Zhou, W. Liang, K. I.-K. Wang, and L. T. Yang, “Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations,” *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 1, pp. 171–178, Feb. 2021.

[25] J. Zhang, Y. Zheng, J. Sun, and D. Qi, “Flow prediction in spatio-temporal networks based on multitask deep learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Jan. 2019.

[26] R. Jiang et al., “DeepUrbanEvent: A system for predicting citywide crowd dynamics at big events,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Anchorage, AK, USA, Jul. 2019, pp. 2114–2122.

[27] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, “Multi-task representation learning for travel time estimation,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, London, U.K., Jul. 2018, pp. 1695–1704.

[28] X. Zhou, W. Liang, J. She, Z. Yan, and K. I.-K. Wang, “Two-layer federated learning with heterogeneous model aggregation for 6G supported Internet of Vehicles,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5308–5317, Jun. 2021.

[29] X. Zhou, Y. Hu, J. Wu, W. Liang, J. Ma, and Q. Jin, “Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in Industrial IoT,” *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 570–580, Jan. 2023.

[30] B. Yu, Y. Lee, and K. Sohn, “Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN),” *Transp. Res. Part C: Emerg. Technol.*, vol. 114, pp. 189–204, Feb. 2020.

[31] H. Shi et al., “Predicting origin-destination flow via multi-perspective graph convolutional network,” in *Proc. 36th IEEE Int. Conf. Data Eng.*, Dallas, TX, USA, Apr. 2020, pp. 1818–1821.

[32] H. Wu, Z. Zhang, J. Luo, K. Yue, and C.-H. Hsu, “Multiple attributes QoS prediction via deep neural model with contexts,” *IEEE Trans. Services Comput.*, vol. 14, no. 4, pp. 1084–1096, Jul./Aug. 2021.

[33] M. Shi, Y. Zhuang, Y. Tang, M. Lin, X. Zhu, and J. Liu, “Web service network embedding based on link prediction and convolutional learning,” *IEEE Trans. Services Comput.*, vol. 15, no. 6, pp. 3620–3633, Nov./Dec. 2022.

[34] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, “Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT,” *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12588–12596, Aug. 2021.

[35] X. Zhou et al., “Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems,” *IEEE Wireless Commun.*, vol. 30, no. 2, pp. 82–89, Apr. 2023.

[36] Y. Zhang and Q. Yang, “A survey on multi-task learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, Dec. 2022.

[37] S. Ruder, “An overview of multi-task learning in deep neural networks,” Jun. 2017, *arXiv:1706.05098*.

[38] Y. Zhang, L. Chu, Y. Ou, C. Guo, Y. Liu, and X. Tang, “A cyber-physical system-based velocity-profile prediction method and case study of application in plug-in hybrid electric vehicle,” *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 40–51, Jan. 2021.

[39] A. A. Nunes, T. Galvão, and J. F. e Cunha, “Urban public transport service co-creation: Leveraging passenger’s knowledge to enhance travel experience,” *Procedia-Soc. Behav. Sci.*, vol. 111, pp. 577–585, Feb. 2014.

[40] M. Bonera, G. Maternini, G. Parkhurst, D. Paddeu, W. Clayton, and D. Vetturi, “Travel experience on board urban buses: A comparison between Bristol and Brescia,” *Eur. Transp. Trasp. Eur.*, vol. 76, pp. 1–12, 2020. [Online]. Available: http://www.istiee.unict.it/sites/default/files/files/1_4_ET_39.pdf

[41] M. Zhu, X.-Y. Liu, M. Qiu, R. Shen, W. Shu, and M.-Y. Wu, “Traffic big data based path planning strategy in public vehicle systems,” in *Proc. IEEE/ACM 24th Int. Symp. Qual. Service (IWQoS)*, Jun. 2016, pp. 1–2.

[42] X. Zhan, S. Hasan, S. V. Ukkusuri, and C. Kamga, “Urban link travel time estimation using large-scale taxi data with partial information,” *Transp. Res. Part C: Emerg. Technol.*, vol. 33, pp. 37–49, May 2013.

[43] H. Wang, X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li, “A simple baseline for travel time estimation using large-scale trip data,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–22, Feb. 2019.

[44] E. Jenelius and H. N. Koutsopoulos, "Travel time estimation for urban road networks using low frequency probe vehicle data," *Transp. Res. Part B: Methodol.*, vol. 53, pp. 64–81, Apr. 2013.

[45] K. M. Dresner and P. Stone, "Traffic intersections of the future," in *Proc. 21st Nat. Conf. Artif. Intell. 18th Innov. Appl. Artif. Intell. Conf., Boston, MA, USA, AAAI Press*, Jul. 2006, pp. 1593–1596.

[46] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, Aug. 2014, pp. 25–34.

[47] Z. Fu, Z. Tian, Y. Xu, and K. Zhou, "Mining frequent route patterns based on personal trajectory abstraction," *IEEE Access*, vol. 5, pp. 11352–11363, 2017.

[48] M. B. Mollah et al., "Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021.

[49] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, Mar. 2007.

[50] R. Trismingsih and S. S. Shaztika, "ST-DBSCAN clustering module in SpagoBI for hotspots distribution in indonesia," in *Proc. 3rd Int. Conf. Inf. Technol., Comput., Elect. Eng. (ICITACEE)*, Semarang, Indonesia, Apr. 2016, pp. 327–330.

[51] P. Gundaliya, T. V. Mathew, and S. L. Dhingra, "Heterogeneous traffic flow modelling for an arterial using grid based approach," *J. Adv. Transp.*, vol. 42, no. 4, pp. 467–491, 2008.

[52] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS 2014 Workshop Deep Learn.*, Dec. 2014.

[53] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Jul. 2017.

[54] G. Zhang, S. Chu, X. Jin, and W. Zhang, "Composite neural learning fault-tolerant control for underactuated vehicles with event-triggered input," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2327–2338, May 2021.

[55] A. F. Agarap, "A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data," in *Proc. Int. Conf. Mach. Learn. Comput. (ICMLC), Macau, China*, Feb. 2018, pp. 26–30.

[56] D. Deng, C. Shahabi, U. Demiryurek, and L. Zhu, "Situation aware multi-task learning for traffic prediction," in *Proc. IEEE Int. Conf. Data Mining*, New Orleans, LA, USA, Nov. 2017, pp. 81–90.

[57] F. Sattler, K. R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed Multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.

[58] F. Zhou, C. Shui, M. Abbasi, L.-É. Robitaille, B. Wang, and C. Gagné, "Task similarity estimation through adversarial multitask neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 466–480, Oct. 2021.

[59] V. Havyarimana, Z. Xiao, A. Sibomana, D. Wu, and J. Bai, "A fusion framework based on sparse Gaussian-Wigner prediction for vehicle localization using GDOP of GPS satellites," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 680–689, Feb. 2020.

[60] Z. Xiao et al., "Toward accurate vehicle state estimation under non-Gaussian noises," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10652–10664, Dec. 2019.

[61] Z. Xiao et al., "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.

[62] J. Y. Campbell and S. B. Thompson, "Predicting excess stock returns out of sample: Can anything beat the historical average?" *Rev. Financ. Stud.*, vol. 21, no. 4, pp. 1509–1531, Apr. 2008.

[63] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003.

[64] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw. ICANN 99.*, vol. 2, Jan. 1999, pp. 850–855.

[65] D. Wang, C. Wang, J. Xiao, Z. Xiao, W. Chen, and V. Havyarimana, "Bayesian optimization of support vector machine for regression prediction of short-term traffic flow," *Intell. Data Anal.*, vol. 23, no. 2, pp. 481–497, Apr. 2019.

[66] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[67] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[68] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017.

[69] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-task CNN model for attribute prediction," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1949–1959, Sep. 2015.

Jiali Yang received the M.S. degree from the School of Design, Hunan University, Changsha, China, in 2021, where she is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering.

Her research interests include Internet of Vehicles and mobile edge computing.

Kehua Yang received the bachelor's degree from Xiangtan University, Xiangtan, China, in 1997, and the master's and Ph.D. degrees from Southeast University, Nanjing, China, in 2000 and 2005, respectively.

He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His current research interests include Internet of Things, mobile computing, and machine learning.

Zhu Xiao (Senior Member, IEEE) received the M.S. and Ph.D. degrees in communication and information systems 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 a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include wireless localization, Internet of Vehicles, and intelligent transportation systems.

Prof. Xiao 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 Huazhong University of Science and Technology, Wuhan, China. His current research focuses on computer networking, especially, wireless networks, data science in Internet of Things, and mobile computing.

Prof. Jiang has been serving on the editorial board of IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON MOBILE COMPUTING, *ACM Transactions on Sensor Networks*, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and IEEE INTERNET OF THINGS JOURNAL. He was also invited to serve on the TPC of IEEE INFOCOM, ACM WWW, ACM/IEEE MobiHoc, IEEE ICDCS, and IEEE ICNP. He is an Elected Fellow of The Institution of Engineering and Technology, a Fellow of The British Computer Society, a Senior Member of ACM, and a Full Member of IFIP TC6 WG6.2.

Shenyuan Xu received the B.S. degree in communication engineering from Hunan Normal University of Information Technology, Changsha, China, in 2019. She is currently pursuing the M.S. degree in communication engineering with Hunan University, Changsha.

Schahram Dustdar (Fellow, IEEE) received the Ph.D. degree in business informatics from the University of Linz, Linz, Austria, in 1992.

He is currently a Full Professor of Computer Science (informatics) with a focus on Internet technologies heading the Distributed Systems Group, TU Wien, Wein, Austria.

Prof. Dustdar was a recipient of the ACM Distinguished Scientist Award in 2009 and the IBM Faculty Award in 2012. He is an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology*. He is on the Editorial Board of IEEE. He has been the Chairman of the Informatics Section of the Academia Europaea since December 2016. He has been a member of the IEEE Conference Activities Committee since 2016, the Section Committee of Informatics of the Academia Europaea since 2015, and the Academia Europaea: The Academy of Europe, Informatics Section since 2013.