The VLDB Journal
https://doi.org/10.1007/s00778-021-00722-0

SPECIAL ISSUE PAPER

Payment behavior prediction on shared parking lots with TR-GCN

Qingyu Xu[1] · Feng Zhang[1] · Mingde Zhang[1] · Jidong Zhai[2] · Bingsheng He[3,4] · Cheng Yang[5] · Shuhao Zhang[6] · Jiazao Lin[7,8] · Haidi Liu[9] · Xiaoyong Du[1]

Received: 21 February 2021 / Revised: 26 September 2021 / Accepted: 3 December 2021

Abstract

Shared parking lots are new types of sharing economy and generate a large social impact in our daily lives. Post-use payment is a hallmark method in the shared parking lots: it reflects trust in users and brings convenience to everyone. Accordingly, payment behavior prediction via data science technology becomes extremely important. We cooperate with a real intelligent parking platform, ThsParking, which is one of the top smart parking platforms in China, to study payment prediction, and encounter three challenges. First, we need to process a large volume of data generated every day. Second, a variety of parking related data shall be utilized to build the prediction model. Third, we need to consider the temporal characteristics of input data. In response, we propose TR-GCN, a temporal relational graph convolutional network for payment behavior prediction on shared parking lots, and we build a reminder to remind unpaid users. TR-GCN addresses the aforementioned challenges with three modules. 1) We develop an efficient data preprocessing module to extract key information from big data. 2) We build a GCN-based module with user association graphs from three different perspectives to describe the diverse hidden relations among data, including relations between user profile, temporal relations between parking patterns, and spatial relations between different parking lots. 3) We build an LSTM-based module to capture the temporal information from historical events. Experiments based on 50 real parking lots show that our TR-GCN achieves 91.2% accuracy, which is about 7% higher than the state-of-the-art and the reminder service makes more than half of the late-payment users pay, saving 1.9% loss for shared parking lots.

✉ Feng Zhang
fengzhang@ruc.edu.cn

Qingyu Xu
qingyuxu@ruc.edu.cn

Mingde Zhang
2019202191@ruc.edu.cn

Jidong Zhai
zhaijidong@tsinghua.edu.cn

Bingsheng He
hebs@comp.nus.edu.sg

Cheng Yang
albertyang33@gmail.com

Shuhao Zhang
shuhaozhang@sutd.edu.sg

Jiazao Lin
linzj@pku.edu.cn

Haidi Liu
liuhd@huaching.com

Xiaoyong Du
duyong@ruc.edu.cn

1 Key Laboratory of Data Engineering and Knowledge Engineering (MOE), and School of Information, Renmin University of China, Beijing, China

2 Department of Computer Science and Technology, Tsinghua University, Beijing, China

3 School of Computing, National University of Singapore, Singapore, Singapore

4 NUS Center for Trust Internet and Community, Singapore, Singapore

5 School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China

6 Information Systems Technology and Design pillar, Singapore University of Technology and Design, Singapore, Singapore

7 Department of Information Management, Peking University, Beijing, China

8 Institute of Automation, Chinese Academy of Sciences, Beijing, China

9 Zhongzhi Huaching (Beijing) Technology Co., Beijing, China

Published online: 09 January 2022

1 Introduction

Sharing economy is booming in most countries, and many manifestations of sharing economy have emerged, such as e-scooters [60], ZipCar [64], and shared bicycles [70]. Intelligent shared parking lots, as a new representative of sharing economy, are becoming popular and generate more and more social impact in recent days. Post-use payment, which means that users pay after use, is a hallmark method in the shared parking lots: it reflects trust in users and brings convenience to everyone. However, users may forget to pay on time. In a shared parking system, correctly predicting the payment behavior of each user, i.e., payment duration after parking, is important for three reasons. First, we need to understand the difference in the payment behavior of people in different regions and remind users who forget to pay, which relates to the income of the parking lot [1]. Note that we should not remind all users, as it could degrade users' experience. Second, we need to identify the potential high-risk users and require them to pay a deposit before the next parking. Third, based on payment behavior prediction, we can provide reminder service and incentive mechanism to help users build good payment habits [55].

In collaboration with an intelligent parking platform, *ThsParking*[1] (developed by *Huaching Tech*[2]), we explore the opportunities of applying artificial intelligence to payment prediction for shared parking lots. Building an accurate payment behavior prediction model for large-scale shared parking lots is non-trivial, and we encounter the following three technical challenges. First, *ThsParking* covers more than 20 cities with more than 1,700 parking lots, providing more than 60 thousand parking spaces. We need to process a large volume of data generated every day. Second, a large variety of parking related data (e.g., user profiles and spatial, temporal parking patterns) need to be utilized to build an accurate prediction model. Third, we need to capture the timing characteristics of input data as parking orders are chronologically generated. Characteristics such as the sudden spikes during peak hours must be considered in the prediction model.

In this paper, we propose **TR-GCN**, a **t**emporal **r**elational **g**raph **c**onvolutional **n**etwork. TR-GCN contains three modules. 1) To solve the volume challenge, we develop an efficient data preprocessing module to extract key information from big data, detailed in Sect. 5.2. 2) To solve the variety challenge, we build a GCN-based module with user association graphs from three different dimensions to describe the diverse hidden relations among data, including relations between user profile, temporal relations between parking patterns, and spatial relations between different parking lots, detailed in Sect. 5.3. 3) To solve the timing challenge, we build an LSTM-based module to capture the temporal information from users' historical transactions, detailed in Sect. 5.4. Considering the efficiency of LSTM, we save the data locally to increase the processing speed.

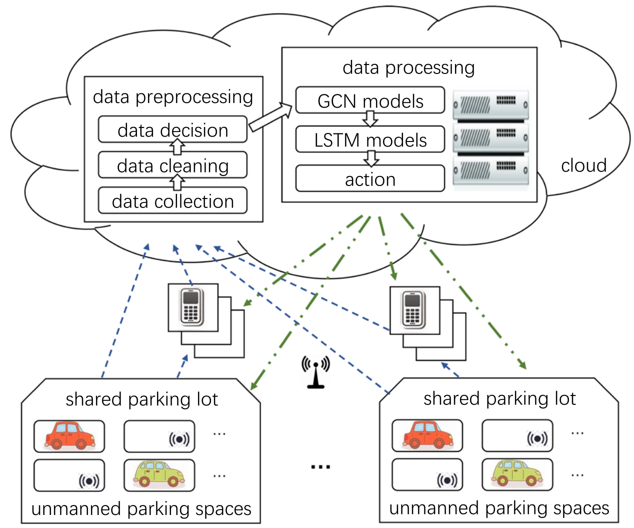


Fig. 1 Data processing pipeline in shared parking lots

We have integrated TR-GCN into the ThsParking platform, as shown in Fig. 1. The overall workflow contains three steps. First, user information and parking space information are sent to the cloud. Second, the system executes the data preprocessing module to perform data collection and data cleaning, followed by a data decision to determine whether this is a valid order for further data processing. Third, the system executes the GCN-based module and the LSTM-based module, to analyze user payment behaviors. Finally, the output of the payment order is sent to the users.

We evaluate TR-GCN on fifty parking lots from nine cities with various surroundings. We collect 131 thousand payment records for evaluation, and each record includes the payment amount, the time interval from the previous order (0 if there is only one order), parking lot ID, coupon usage, and date. We have classified the prediction results into two categories, which are on-time payment and delayed payment. Experiments show that TR-GCN achieves 91.2% accuracy, which is about 7% higher than the state-of-the-art payment prediction method [74]. Moreover, our reminder service could save 1.9% loss for shared parking lots and help users build good payment habits, which brings obvious social impact.

We summarize our contributions as follows.

– We discover the hidden relations between users, and develop novel methods to capture the association graphs from various aspects.

[1] http://www.thsparking.com

[2] http://www.huaching.com

- We develop TR-GCN, which integrates temporal and spatial pattern features to predict payment behavior for shared parking lots.
- We evaluate TR-GCN with real payment records and achieve 91.2% accuracy, which is about 7% higher than the state-of-the-art method.

The remainder of the paper is organized as follows. Section 2 introduces the background of payment behavior in shared parking lots, graph convolutional network, and LSTM in prediction. Section 3 presents the observation and finding of shared parking, and motivation of this work, followed by the definition of the problem and challenges in Sect. 4. We present the overview of TR-GCN and the detailed design in Sect. 5. We show our evaluation in Sect. 6 and the related work in Sect. 7. Finally, we show the conclusion in Sect. 8.

2 Background and preliminaries

In this section, we introduce the background and preliminaries of payment behavior prediction.

2.1 Payment behavior in shared parking lots

Following previous work [74], we model the payment behavior as the process of whether users will pay for parking on time. Different from payment in traditional parking lots, intelligent parking lots adopt an unmanned payment method and users pay after use. This brings higher requirements to users: in order to improve the user experience, we try not to use too many reminders to remind users to pay. Instead, we let users pay by themselves at their available time with a minimal number of reminders. Therefore, payment behavior prediction on shared parking lot becomes extremely important.

Our intelligent parking platform can operate automatically without being guarded, as shown in Fig. 2. The process of parking in a shared parking space is as follows. First, users use their electronic devices to scan the QR code to obtain the parking space information via our APP. Second, our APP uploads the parking space and user information to our cloud servers. Third, after the cloud server receives the order, the server sends the unlock signal to the electric lock and the lock shall go down. Then, users can drive their car into the space directly with no additional operations required. After parking, the users can just drive away. When the car lock sensor detects that the car has been driven away, it will raise the lock and send the end time to the server for billing.

Why we do not use instant payment. We do not use instant payment methods, such as ETC [29], in our intelligent shared parking lots because the pay-after-use mechanisms provide

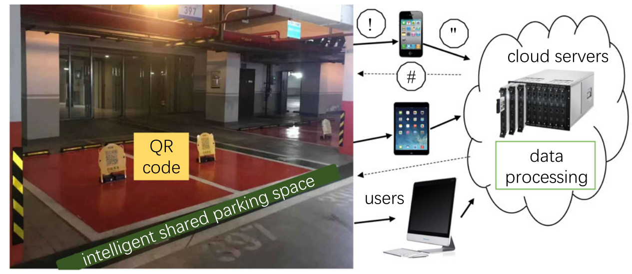


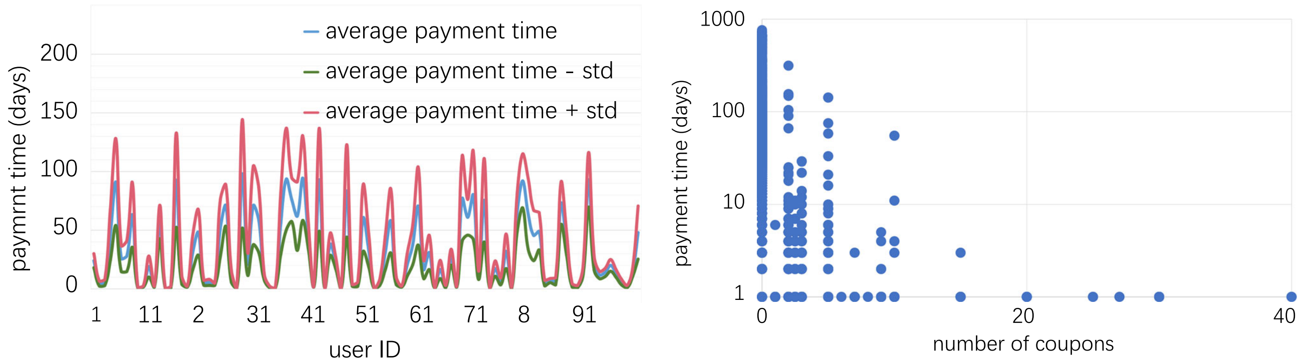
Fig. 2 Parking process in shared parking lots

users better experience: the shared parking lots do not contain toll stations or roadblocks to stop users from leaving. Accordingly, users can pay at any time and anywhere after they leave. This becomes a hallmark in sharing economy. In addition, the shared parking lots are nationwide and can have a large scale, so it is not cost-effective to install toll stations or roadblocks.

In the literature, many studies have been conducted on parking payment prediction [6,19,30,74,80]. Xu *et al*. [74] applied decision tree for payment prediction, which is the closest work to ours. However, the work [74] has three limitations. 1) Volume. Only a small dataset from one parking lot is used in their evaluation, which is not representative. 2) Variety. They analyze only parking records, but ignore hidden relationships from multiple dimensions. 3) Temporal relationship. Only a simple decision tree model has been built, which fails to consider timing characteristics. In response, our work addresses these three limitations by developing three modules in TR-GCN: 1)data preprocessing, 2)GCN modeling, and 3)LSTM modeling. We elaborate the detailed design in Sect. 5.

2.2 Graph convolutional network

GCN [32] is an effective variant of convolutional neural networks [35] that runs directly on graph structured data. Graph neural network can fuse the features of current users with those of their neighbors so the representation of the features can be improved. GCN has been applied to various prediction scenarios [26,40,53]. Yang *et al*. [76] used GCN to extract spatial and temporal features to predict parking spaces based on traffic information. Chu *et al*. [10] proposed a semi-supervised classification model based on GCN to predict the waiting level of public transportation platforms. Liu *et al*. [44] proposed the first heterogeneous graph neural network to detect malicious accounts. Kim *et al*. [31] proposed a deep dense convolutional network that can predict the borrower's repayment. Tam *et al*. [65] proposed an algorithm based on end-to-end GCN to learn node and edge embedding, which can be used to detect abnormal transactions in electronic payment networks. There are also other variants



(a) Payment trends. **(b)** Coupons.

Fig. 3 Potential relation between user profile and payment behavior

of convolutional neural networks such as graph autoencoder (GAE) [58]. Autoencoder (AE) and its variants are widely used in unsupervised learning. It is suitable for learning node representation of graphs without supervised information. As for our work, GCN is more suitable and efficient. The reason is that GAE are more powerful in unsupervised tasks, while GCN has been proved to be efficient in graph embedding by many works [26,40,53]. Moreover, there are many works using GCN to model dynamic graph-structured data [47,92].

2.3 Long short-term memory

LSTM [21] is a recurrent neural network (RNN) and has been applied to capture the impact of historical events in many fields [42,61,72,75,85,87]. LSTM is good at processing and predicting events with long intervals and delays in time series. An LSTM network is composed of LSTM blocks, which include input gates, forget gates, and output gates. Such special architectures enable LSTM to capture the complex nonlinear relationships between different features in time series. Zhang *et al.* [80] proposed a novel LSTM method with weather factors and periodic parking patterns to predict parking behaviors. Zhao *et al.* [91] used LSTM for short-term traffic forecasts. Han *et al.* [24] developed a fast and energy-efficient LSTM in speech recognition.

3 Observation and motivation

We select fifty parking lots in nine cities from a real intelligent parking system. In addition to the payment records, the platform also provides parking coupons, which could be used to analyze the correlations in data and mine the hidden relations between users. In our study, we find three hidden relations that may affect the payment behavior: 1) relations between user profile, 2) temporal relations between parking patterns, and 3) spatial relations between different parking lots.

3.1 Hidden relation 1: user profile

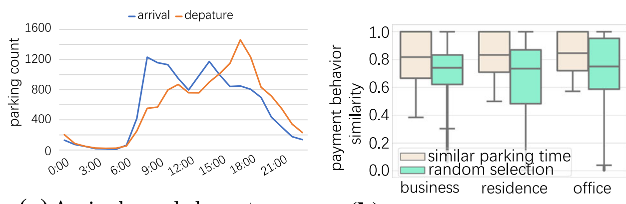
Observation (1): The user payment behavior has a strong correlation with the users' payment profiles.
Insight (1): Profile contact needs to be involved in payment behavior prediction.

We explore the relation between users' historical records and their new orders. We show the payment correlation from the user profile dimension in Fig. 3.

We first find that correlations exist between users' historical payment behavior and users' average payment time. We randomly select 100 users and demonstrate the correlations in Fig. 3a. In Fig. 3a, the x axis shows different user IDs, and the y axis shows the payment time in terms of days. We plot the average payment time for each user and show the standard variance, which is the two lines around the average payment time line. We can see that the standard variance is correlated to the average payment time, which means users with high average payment time tend to fluctuate and are less likely to pay in time.

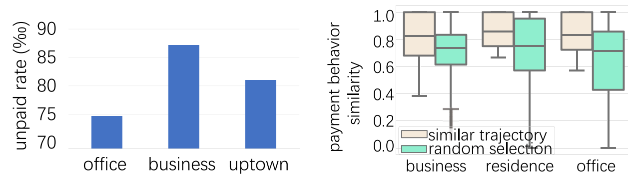
We observe that coupon history has correlations on users' willingness to pay in time, as shown in Fig. 3b. Figure 3b exhibits the relationship between the number of coupons a user bought (x axis) and the payment time of the order (y axis). Coupons can reduce the payment cost and have use period, and we find that people who bought coupons are more willing to pay in time.

Based on these findings, we have the insight that the majority of users' payment behaviors relate to their past payment behaviors. It reveals that users' profile features have connections with their payment behaviors. Thus, we need to make use of the users' profile data for future payment behavior prediction.



(a) Arrivals and departures. **(b)** Payment similarities.

Fig. 4 Relation between temporal parking patterns and payment time



(a) Different parking lots. **(b)** Parking lot group.

Fig. 5 Influence of spatial relations on payments

3.2 Hidden relation 2: users' temporal parking patterns

> **Observation (2):** Users with similar parking behaviors tend to have similar payment behaviors.
> **Insight (2):** Parking period patterns should be considered in payment behavior modeling.

We find that users have potential correlations with each other, and users with similar parking patterns can have similar payment patterns. For example, the users may be colleagues who have the same commuting time. Since we lack users' private friendship relations, we propose another way to explore what range of users are more likely to be colleagues or have the same working identities. We show the correlation between payment time and temporal parking patterns in Fig. 4.

Figure 4a shows the statistics of user arrival and departure times of parking lot P10 (detailed in Sect. 6.1).

In Fig. 4 (b), we analyze 53,168 parking records from October 20, 2017, to October 20, 2018, and compare the payment similarities between user pairs with similar temporal parking patterns and random user pairs in different types of parking lots. The payment similarity of a user pair is defined as $1/(|AvgPaytime_i - AvgPaytime_j| + 1)$, where $AvgPaytime_i$ denotes the average paytime of user i. For user pairs with similar temporal parking patterns, we group the users whose arrival time gap is within one hour and departure time gap is also within one hour (detailed in Sect. 5.3). In contrast, for random user pairs, we randomly select users to form user pairs. Figure 4b shows that the payment similarity with similar temporal parking patterns reaches 0.81, which is much higher than the randomly picked user pairs whose average similarity is only 0.75. Furthermore, the minimum point of payment similarity with similar temporal parking patterns is closer to the average compared to the random one.

Accordingly, these data can be used in payment prediction: if we identify that two users belong to the same group, we can perform cross-user payment prediction.

Based on the observation, we have the insight that people who share the same temporal parking patterns, such as similar arrival and departure patterns, have the same payment habits. Therefore, parking period patterns need to be involved to refine the payment behavior model in shared parking lots.

3.3 Hidden relation 3: spatial relations between parking lots

> **Observation (3):** Users who have similar historical trajectories are likely to behave similarly when they pay their orders.
> **Insight (3):** Capturing users' hidden relations by utilizing their parking spatial patterns can be helpful to payment behavior modeling.

Inspired by practical experiences, we consider that users who have similar trajectories can share similar payment patterns, and they are likely to belong to the same identity. Moreover, we can verify that users who often go to the same locations have similar payment behaviors from our parking platform. The payment rates in different parking lots are different. We show the unpaid rates of different parking lots in Fig. 5a. We can see that parking lots around residential areas have the lowest unpaid rate while parking lots around office buildings have the highest rate. Moreover, we find that users who go to similar parking lots have similar payment times. Figure 5b shows the payment similarity comparison between user pairs with similar spatial relations and random user pairs in different types of parking lots, from October 20, 2017, to October 20, 2018. We define the similarity between two users' average paytime as $1/(|AvgPaytime_i - AvgPaytime_j| + 1)$, where $AvgPaytime_i$ denotes the average paytime of user i. We regard users' trajectories as their movements among different parking locations. When the number of two users' co-occurrence parking lots exceeds our threshold (default 2), an edge shall be built between them, which is to say, they have similar trajectories. Figure 5b shows that the average payment similarity of user pairs with similar spatial relations reaches 0.85, which is higher than the randomly picked user pairs whose average similarity is 0.74. Moreover, the payment behaviors of similar spatial relations are more concentrated, while those of the random user pairs are more scattered. The above findings prove the effectiveness of our method.

Therefore, we have the insight that users who have the same historical trajectories can belong to the same user group. For example, users go to the same parking lots that are near the business center, and they can be visitors who may forget to pay after they leave. Accordingly, we should utilize this relation to build a spatial connection graph based on historical trajectories. We can also speculate that unpaid rates are different in various types of parking lot, so we should consider users from different types of parking lots, respectively.

3.4 Hidden relation 4: types of parking lots and weekdays

Observation (4): It can be observed that parking lot types and weekdays have correlation with the unpaid rates.
Insight (4): We should involve date information and parking lot types in our features to better model users' payment behaviors.

Previous study [80] has demonstrated the influence of parking lot type and weekdays on parking behavior. It would be interesting to find out whether certain relations between types of parking lots and payment behaviors exist in shared parking scenarios, or whether the date feature has an influence on our shared parking payment in a non-trivial way.

We observe that the unpaid rates of different types of parking lots differ from each other and vary on different weekdays. In detail, in Figure 6a, we show the unpaid rates of three types of parking lots. Parking lot P7 is located near a business center, parking lot P10 is located beside an office building, and parking lot P18 is near a residential area. We can see that these three types of parking lots show totally different behaviors. To explain this, people can go shopping in the business street, so on weekends, P7 has the highest unpaid rate. People often visit office buildings on working days so P10 has a high unpaid rate on weekdays. In contrast, the residential area has a relatively stable structure, and residents who live for a long time are more likely to develop a fixed payment habit. Hence, P18 has the lowest unpaid rate. We also demonstrate the average unpaid rates for all fifty parking lots used in our evaluation in Fig. 6 b. On average, we can see that unpaid rates are high on weekends. In detail, there are 29 parking lots located near a business center. Besides, people are willing to go out on weekends and may forget to pay when they go home.

Based on these observations and analysis, we conclude that the unpaid rates have a strong correlation with parking lot types and weekdays. This also reveals that the specific influence of weekday information on various types of parking lots is different. This finding directly leads to the conclusion that we can make more precise predictions by involving the parking lot type and weekday information.

3.5 Motivation

Previous studies [6,15,25,68,69,74] ignore these important hidden relations. We should abstract and involve these hidden relations in our payment prediction. Moreover, the periodic payment history should also be utilized in payment behavior prediction. A graph convolutional network can be used to capture the complicated relations between users while an LSTM-based layer can be applied to capture the temporal relations during payment prediction.

4 Problem definition

The problem we are trying to solve is to predict whether a user will pay on time when an order is generated.

4.1 Parking payment behavior

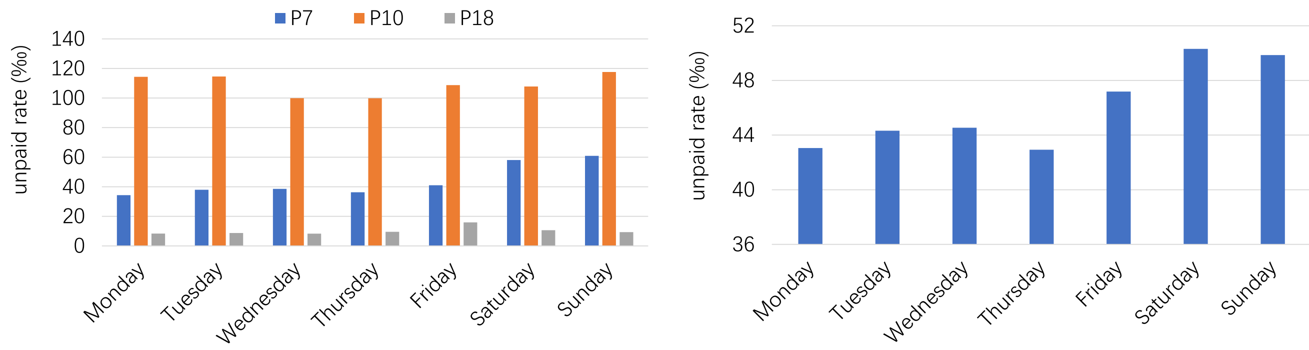
The payment behavior refers to the payment process for a user after parking. Considering a set of parking records associated with the user a, $P^a = \left\{P_i^a\right\}, i \in [0, N_a - 1]$, where N_a is the number of parking records that belong to the user a. The payment behavior of 1) timely payment, 2) late payment, and 3) unpaid, relates to two considerations, user characteristics U_i^a and order characteristics O_i^a:

- $U_i^a \in R^{1\times M}$ denotes the extracted M-dimensional feature vector (*e.g.*, recency, frequency, and monetary), which belongs to the user a in the time of her order P_i^a.
- $O_i^a \in R^{1\times V}$ denotes the property of the order P_i^a (*e.g.*, parking duration and date).

In detail, the M-dimensional feature vector contains traditional features used in RFM model [8], which can reveal the patterns in users' purchase history.

Criteria We use the average payment time of all paid orders as the threshold to decide whether an order is paid on time. In detail, we first calculate the average payment time of all historical orders as t_{avg}. We find that t_{avg} varies in different types of parking lots, so we count t_{avg} separately for each parking lot. We leave the time interval for re-evaluating t_{avg} to the owner of the parking lot. By default, t_{avg} is re-evaluated every three months. Then, given an order P_i^a, if its payment time is longer than t_{avg}, it is regarded as a late payment; otherwise, it is a timely payment.

Importance of payment behavior prediction. Our payment behavior prediction is important for shared parking lots. First, since the intelligent shared parking lots use unmanned methods to operate, users can pay at any time after leaving the parking lots. Thus, delayed payments occur and can lead to a



(a) Different types of parking lots.

(b) Average unpaid rate for all parking lots.

Fig. 6 Unpaid rates in different types of parking lots and weekdays

high income loss of the shared parking lots. If we can detect the delayed payment in advance, we can take corresponding actions. For example, we can build a reminder afterward to remind these users to pay. Second, besides shared parking, many intelligent products of the sharing economy adopt the business mode of unmanned payment methods. However, we have not found any methods that can be used in the payment prediction situations with hidden relations. Thus, it is necessary for us to put forward an effective and practical payment prediction model for shared parking lots. It is notable that our model can be adapted to other similar problems, detailed in Sect. 6.4. Third, inspired by previous works such as fraud detection [13], detecting the users who are potentially unwilling to pay can generate positive social impacts.

Necessity of payment behavior prediction Parking is becoming a part of our lives and in the era of sharing economy, shared parking lots have become an inevitable trend. The basis of shared parking spaces and even the sharing economy is trust, so our platform does not require a deposit in advance. Accordingly, payment behavior prediction becomes extremely important, based on three reasons. First, we can understand the payment behaviors of different users and remind the users who forget to pay. Note that users have different payment habits which should be respected, so we cannot simply remind them by rules. Second, we can identify the potential high-risk users, and then require them to pay a deposit before the next parking. Third, by user payment behavior prediction, we can provide reminder service and incentive mechanism to help users build good payment habits.

Privacy versus accuracy If we can directly obtain the user's detailed information and the relationship between users, we can build an accurate payment prediction model. However, due to privacy issues, we are unable to obtain the personal information of users. Fortunately, we can roughly supplement their relationship by mining their parking behaviors.

Abstraction for payment prediction We further abstract the payment prediction problem. Given the features of the orders $\Theta = \left\{O_i^a\right\}, i \in [0, N_a - 1]$, and features of the users in different time periods $\upsilon = \left\{U_i^a\right\}, i \in [0, N_a - 1]$, our problem is to predict whether the order P_i^a will be paid on time, as shown in Equation 1. If the order is paid in time, y is equal to 1; otherwise, 0. $f(\cdot)$ is the mapping function we want to learn by our models.

$$f(\Theta, \upsilon) \rightarrow y, y \in \{0, 1\} \tag{1}$$

Significance AI techniques have been widely adopted in data science domain [78,86]. The social impact of intelligent parking payment prediction is significant, especially in the background of the sharing economy era. In addition, the parking payment prediction has not been adequately addressed by the AI community. In this paper, we propose an effective neuron-based approach to solve it in an AI mechanism.

4.2 Challenges

To solve the parking payment behavior problem, we need to handle the following three challenges.

Challenge 1: Volume Our intelligent shared parking lots are distributed in multiple cities over the country, and the daily user flow is large, especially in peak hours. For example, our system contain 11 million parking records in total. Currently, the daily user flow is 11,352. In peak hours, we need to process 2,058 MB per minute. Our service is placed on Alibaba Cloud and is responsible for nationwide business. We need to process the uploaded data from shared parking lots in time. Moreover, with the increase of the amount of data, noise data gradually appear and we should also handle the problem of dirty data. In the large amount of data input, there are issues of missing data (such as attribute missing in a record), wrong data (data format is not correct, outliers, etc.), and data unavailability [93].

Challenge 2: variety We need to involve various data types to describe the hidden relations in our input. For example, we can use user payment records and member purchase records. We need to join these data from different perspectives to generate our final features. In addition, we are unable to extract users' connections directly due to privacy issues. Hence, we need to dive deep into different types of data and mine the hidden relationships between users. However, there are few works about how to extract association graphs for users in the parking scenario. Even worse, we do not have users' friendship information or any obvious personal information in our platform. Thus, this is the most difficult part of our work and worth lots of investigation.

Challenge 3: timing Our parking records, payment records, and the purchase records all come as time series data, which means that we need to either summarize the information in historical records or find another way to utilize users' historical behaviors. Previous work [74] extracted features from historical data and used the decision tree model for parking behavior prediction. However, it relied heavily on feature engineering, which is not universal. Thus, we need to develop other methods to handle the time series data. There are several variants of temporal neural networks that can be utilized to handle time series data, so special designs need to be made to choose and adapt temporal networks in parking situations.

5 TR-GCN

In this section, we start with our TR-GCN framework, which consists of a data preprocessing module, a GCN module, and an LSTM module. Then, we show our detailed data preprocessing, design of GCN-based modeling for hidden relations between users, and LSTM-based modeling for temporal relations.

5.1 Overview

We show our TR-GCN framework in Fig. 7. We find that graph neural network combined with time series model is very suitable for parking payment prediction. The input is the adjacency matrix and characteristic matrix of graphs and the output is the classification result. The input data consist of three parts: the user purchase history, the user payment history for buying memberships and coupons, and the parking lot location.

Framework In our TR-GCN shown in Fig. 7, we treat each user as a node in a graph, and use a clustering step to build their connections by different data types. Matrix A represents the relations between different users. A graph node represents a user and the edges in the graph represent the relations between users. In matrix A, each element represents whether there is an edge exiting between the two nodes. Matrix D represents the features of each user. In matrix D, each row is a feature vector and each column represents a different attribute. GCN takes these two matrices as inputs. For users who have similar parking patterns, we build edges between them. With such methods, we build three association graphs, including contact graph G_p^T, pattern graph G_t^T, and spatial connection graph G_S^T. We next design a fusion model to integrate the three association graphs into the same one. At a time step, each user generates an adjacency matrix A^T and a feature representation matrix D^T. The LSTM-based modeling includes an LSTM layer to capture the relationships in a temporal mechanism, followed by a fully-connected layer, which outputs the predicted results.

Necessity for data preprocessing To tackle the volume challenge mentioned in Sect. 4, we perform data preprocessing to ensure the high quality of our input parking data. The general preprocessing method is not optimized for the parking payment scenario. The preprocessing for parking data can be divided into three parts. The first part is data collection. At this stage we select the parking attributes required for training. We face special difficulties in data merging. The data resources are various, and we need to merge different user data together, such as users' parking records and membership records. The second part is data cleaning, which includes data format transformation, missing attribute supplement, and outlier analysis. We need to develop data cleaning methods targeting the parking scenario. The third part is data decision, which is used to input the preprocessed data into the model. After data cleaning, we apply sampling methods to solve the imbalance problem. More details can be found in Sect. 5.2.

GCN To handle the variety challenge mentioned in Sect. 4, we need to use multi-dimensional information and mine hidden relationships. As discussed in Sect. 3, we find that hidden relations exist between users, which should be utilized in payment behavior prediction. GCN is a powerful variant of convolutional neural network and it can capture the complicated hidden relations between users.

LSTM LSTM can capture the temporal information from historical events. First, the relation graphs are changing temporally. Second, user payment may relate to different time points. For example, 9:00 on Thursday could be a good time but 9:00 on Saturday could not (detailed in Sect. 6.3). Third, user payment records with timestamps belong to time series data and LSTM is capable of learning such long-term depen-

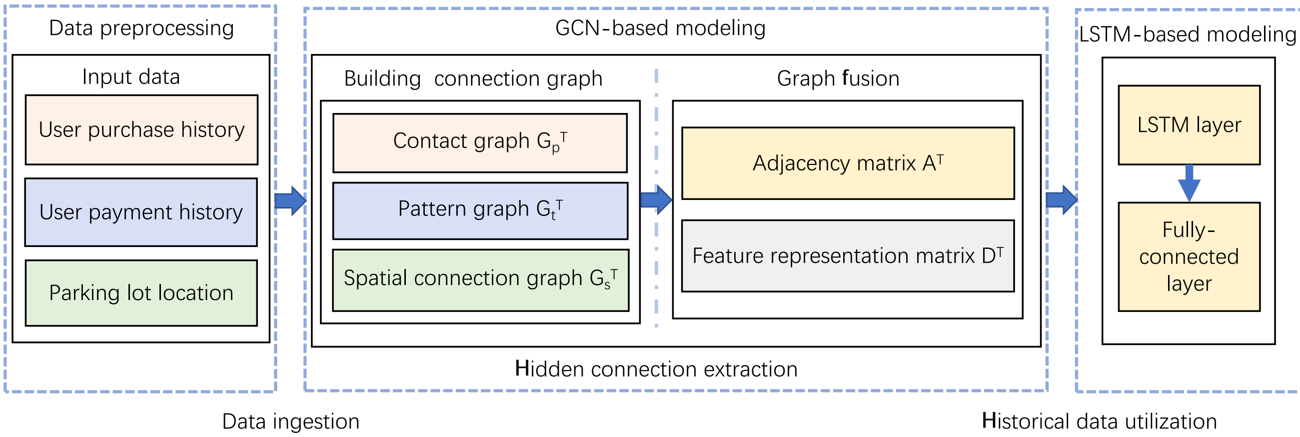


Fig. 7 TR-GCN overview

dencies. Considering the efficiency of LSTM, we save the data locally to increase the LSTM processing speed. We have followed the design of the previous work [48] to support fast data ingestion in our system. Due to space limitations, we leave a detailed discussion to separate work.

Difference from previous work We focus on the special scene of parking payment on shared parking lots and pay special attention to the mining of hidden information from users. As discussed in Sect. 3, we observe that the relations of user profile, users' temporal parking patterns, and spatial relations between parking lots all relate to parking payment behavior. Previous parking related research works [3,6,15,17, 25,43,68,69,74,89,96] do not consider these hidden factors. We show in detail how to build GCN-based models to abstract these hidden relations in Sect. 5.3.

Workflow The workflow of our TR-GCN is shown in Fig. 8. The matrix X represents the output of GCN and it also works as the input of LSTM. X_i^t is the feature matrix of user i in time t, and it has been processed by GCN to serve as the input matrix for LSTM. R represents the real number field. N and de in Fig. 8 represent the number of input orders and the dimensionality of input features. The LSTM model in parking payment prediction is further elaborated in Sect. 5.4. TR-GCN includes four major steps. The first step is to construct the user association graphs according to the relevant observations. We consider three types of association graphs in our system. In each association graph, we specifically create edges that can aggregate users' information. Accordingly, we can map users' features to hidden vectors. The second step is to construct a graph neural network that can learn the user's hidden vector. The user network diagram constructed in the first step is used as input and put into our GCN for training. After obtaining the output of GCN, we have a behavior sequence of users. In the third step, we input the sequence features into our LSTM module for training and then use the next feature vector of users to predict the final payment behavior. Fourth, we use the latent vector of LSTM output as global embeddings and local features of the current order to input into a fully connected layer for final classification. We find that such a design can obtain the hidden relationships between users, using only the parking information.

Solution to challenges Our work, TR-GCN, is motivated by three challenges, as discussed in Sect. 1, which are successfully solved in this paper. First, the data collections of payment relations, especially the users' hidden relations, are difficult to handle. To solve this challenge, we build three user association graphs to capture the hidden relations from different perspectives. Second, the payment modeling problem is a time-intensive activity, which involves understanding the payment behavior from both temporal and spatial perspectives. To handle this challenge, we develop a GCN-based modeling approach to capture the hidden relations between users and develop an LSTM-based modeling approach to utilize the temporal approach. Third, the social impact from intelligent parking payment prediction is hard to measure. To solve this challenge, we develop novel experimental designs to analyze the social impact of our neuron-based payment prediction method.

Novelty Our novelty is mainly reflected in the following three aspects. First, for the parking scenario, we integrate the temporal relational model with graph convolutional model, which can capture both the spatial relations between users and the temporal relations between parking records. Second, we develop a novel data gathering technology to obtain hidden relationships between users without knowing the user's private data. Third, we prove that a novel reminder can be built based on TR-GCN to help users establish good payment habits and save losses.

Relevance to responsible data science TR-GCN not only explores the application of graph neural networks to data

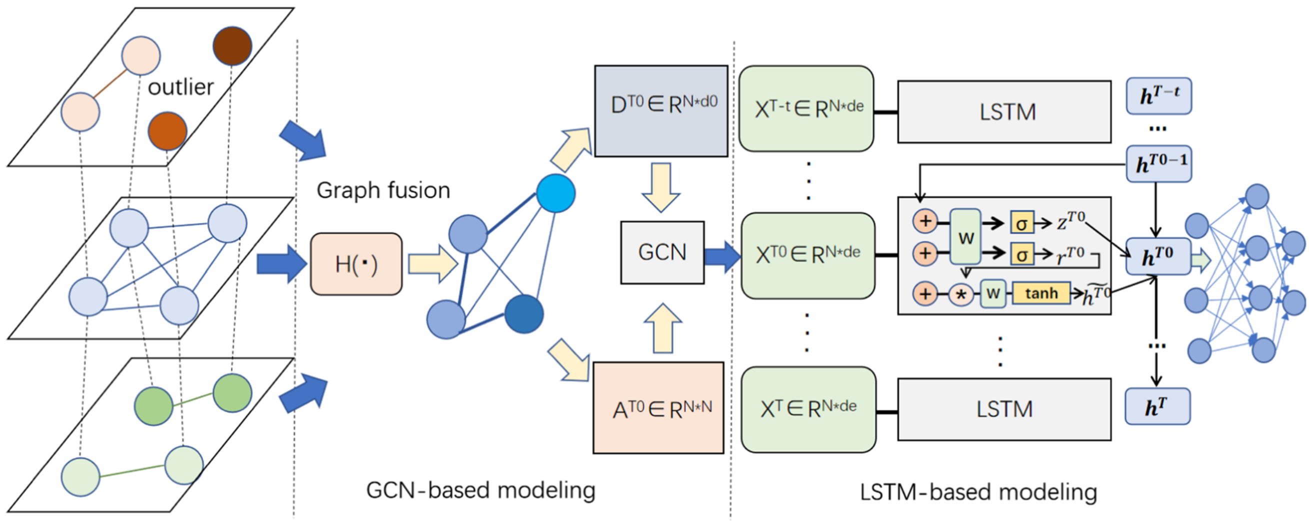


Fig. 8 TR-GCN workflow. The nodes on the left represent users in different association graphs, and the edges between nodes represent their relations

management, which can increase our knowledge from the data we have, but also can validate the reliability of this method on real datasets and set up the system to verify the reliability of the relationships we found. Specifically, we provide a real-world AI application scenario in data science domain. We also conduct data driven analysis and observations, and prove the effectiveness of AI-powered data science technologies. Moreover, we involve data cleaning and selection for effective learning, detailed in Sect. 5.2. All these results can shed lights on the future research of AI-powered data science applications.

In the rest part of this section, we introduce the data preprocessing, GCN-based modeling, and LSTM-based modeling of TR-GCN, which are the solutions to the challenges mentioned in Sect. 4.2.

5.2 Data preprocessing

The original orders are both dirty and messy, with large data volumes, so the data preprocessing is important in our system.

Solving the problem of a large volume of data The large volume challenge brings us great difficulties. As discussed in Sect. 4.2, the large data volume adds massive burden to both data analysis and model training. Consequently, we need to develop an efficient mechanism to solve the volume challenge. First, we use the data preprocessing module to correct the dirty data and gather useful data with proper format, so as to reduce our analysis and training time. Second, in parameter setting, we use sampling methods to set parameters through a small amount of data. Third, we use the preprocessed data for training, instead of the original data, which can help us store the data in a more compact form and save memory space. In addition, we store the input data into memory in blocks, and then merge the blocks together, instead of putting all the data into memory at once.

Data collection Our platform covers major cities in China and has accumulated a large number of users. The number of our registered users reaches two million. However, the parking lots could be out of service due to construction, disasters, and activities. Parking lots can be recently put into use with only a small amount of data. We collected a large number of parking records from different parking lots in various cities. More details can be found in Sect. 6.1. These parking lots are representative and relatively stable in operation. Due to various data resources, we need to merge the data together. For example, we need to connect users' parking records with their membership information through a telephone number-ID table. We collect their user IDs, coupon usage, contact information, parking lot ID, parking lot location, environment, parking start time, end time, payment time after parking, and payment amount.

Data cleaning After data collection, we need to perform data cleaning and selection for effective GCN-based modeling. Although the AI-powered technology has been proved to be successful in parking behavior prediction [74,80,81], the accuracy of payment behavior prediction of TR-GCN depends on the quality of input data. For example, the various data format, missing attributes, and outliers in parking records generate adverse effects on the decision model of parking payment prediction. The general preprocessing method is not optimized for the parking payment scenario. We utilize responsible data processing to maximize the quality of data, including data format transformation, missing attribute supplement, and outlier analysis, as shown in Fig. 9.

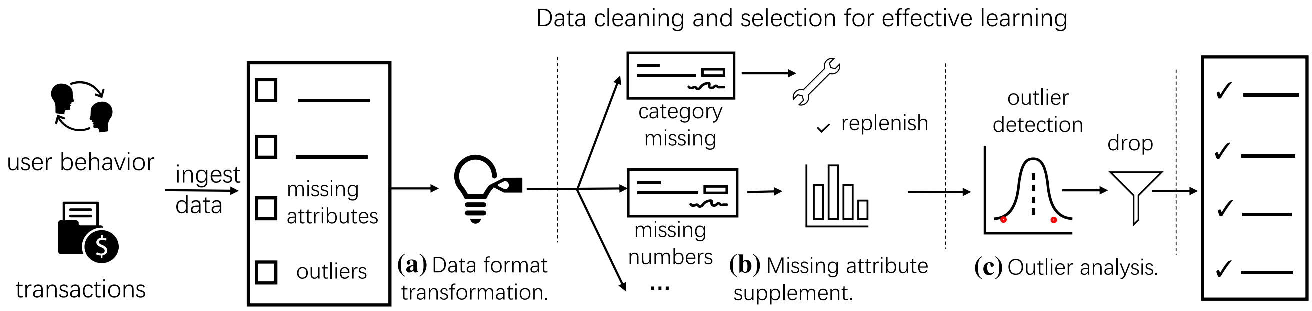


Fig. 9 Data cleaning and selection for effective learning

1) Data format. The first difficulty is data format, because the collected data attributes are of various data types, such as date, unique integers, float, and text. We need to design an efficient structured format to store the collected data. We use descriptive variables for parking lot surroundings via text analysis and feature selection method. In addition, we allow a parking lot to belong to multiple categories. We also turn the date format from string to a specially designed date format.

2) Missing attributes. The collected records involve missing attributes, which could be caused by power failure or other device problems. We do not simply drop the record with missing features because that means we will drop other useful information. For example, for the missing categorical features, such as the parking lot, we add a new category and label the attribute as "missing." For the missing numerical features, we fill the blank with the mean value of this attribute. However, for missing important features such as parking time, we have to remove the record.

3) Outliers. There are also noise data that affect accuracy. For example, if the interval between start and end times is extremely short or long, then this could be an outlier. There are many studies on detecting and processing noise data [4,14,34], and an outlier analysis needs to be conducted to remove noise records. Basically, we follow the rule of Pauta criterion [66], which assumes that the possibility of values that lies in the band between $(\mu - 3\sigma, \mu + 3\sigma)$ is about 99.73%. μ is the mean of the value, and σ is the standard deviation of the value. Any value lies outside the bound will be labeled as an outlier.

Data decision After data cleaning, we input the preprocessed data into TR-GCN. The GCN-based model builds the relation graphs from user connection, parking pattern, and spatial connection dimensions, and the dynamic graphs are then input into the LSTM-based model. It is noticeable that the imbalanced data require extra sampling methods for processing, which shall be discussed in Sect. 6.4. After an order is generated, TR-GCN provides the information about whether to remind the user to pay on time.

5.3 GCN-based modeling

In GCN-based modeling, we first generate user association graphs from different dimensions. Second, we merge these user association graphs. Third, we input them into GCN for training. A similar process has also been used in [59,88].

Clustering for user connection graphs We develop an adaptive clustering method for user connection graphs based on the traditional K-means [33]. In terms of user clustering, what we mainly use are users' payment features.

Regarding the traditional RFM profile user features [8], we define a user in five dimensions: 1) the interval between the user's last payment, 2) the current payment (recency, 0 if it is a new customer), 3) the payment amount of this time, 4) users' membership, and 5) coupon usage. We use these features to cluster users. Note that a user is still a point after clustering. We can measure the distance between two points and regard it as the similarity of two users' profile features. If the distance between two points is less than the threshold or the distance between k nearest neighbors, an edge shall be established between the two users. We standardize the numeric features, and we count the distance between two nodes as $dist = \sum_{i=1}^{5} \left| feature_i^{node1} - feature_i^{node2} \right|$, where i denotes the i-th feature of the node. Because the clustering results from K-means are affected by the centroid of random initialization at the beginning, which may fall into local optimum, we utilize the grid search method [5] to generate ten random numbers in each cluster, and select the result with the smallest clustering SSE [50].

T represents a timestamp. We set T at day granularity based on the following reasons. First, we observe that most users generate one or two parking orders per day. Second, the day granularity is accurate enough from the perspective of the parking platform. Third, smaller granularity can increase the graph construction time.

The three types of association graphs are generated as follows.

1) Profile contact graph For a given time T, we construct a profile contact graph $\mathcal{G}_p^T = \left(U^T, E_p^T\right)$. The sequence of the profile contact graphs for the past T time steps is G_p = $\{G_p^{t-T}, G_p^{t-T+1}, ..., G_p^{t-2}, G_p^{t-1}\}$, where t represents the current time and G_p^i represents the profile contact graph of the users arrived before day i. $U^T = \{u_a^T\}$ is the list of users appeared before time T, and $E_p^T = \left\{\left(u_i^T, u_j^T\right)\right\}$ is the set of edges that connect users according to similar profile properties.

Basic idea The idea of constructing this graph is that if the attributes between two users are similar, they are likely to belong to the same type of consumers, so their payment behaviors may also be similar. We try to use this connection to improve the representation of the original features.

Building edges The graph construction process is as follows. First, we set up a user profile attribute association graph. We are mainly concerned about the payment habits and payment amount of users. We use the interval between the user's last payment and the current payment (recency, 0 if it is a new customer), the payment amount of this time, and coupon usage, as the user's profile features. Second, we follow traditional user profile division methods (RMF model) and use the clustering method to divide users into three categories with reference to [8,9].

Third, after clustering, we specify that if the distance between user features is less than the threshold value (set by grid-search [38]) or the distance between k nearest neighbors, an edge shall be established between the two users, as shown in Eq. 2.

$$e_{i,j} = \begin{cases} 1, & \text{dist}\left(u_i, u_j\right) \le \alpha || max\left(\epsilon, dist_{knn}\left(u_i\right)\right), i \ne j \\ 0, & otherwise \end{cases} \tag{2}$$

Parameter setting Here, we specify the threshold value α by using grid-search [38]. Because there are few studies of constructing association graphs in our parking scenario, we utilize automatic methods to help us find the suitable parameter in this situation. There are several unsupervised clustering methods. We choose K-means [33] for its simplicity and strong interpretability. Since we choose K-means as our clustering method, we set k according to the clustering result the user belongs to.

2) Parking period pattern graph For a given time T, we construct a time pattern graph $\mathcal{G}_{tp}^T = \left(U^T, E_{tp}^T\right)$. The sequence of the time pattern graphs for the past L time steps is G_{tp} = $\{G_{tp}^{t-T}, G_{tp}^{t-T+1}, ..., G_{tp}^{t-2}, G_{tp}^{t-1}\}$, where t represents the current time and G_{tp}^i represents the time pattern graph of the users arrived before day i. $U^T = \{u_a^T\}$ is the same list of users in $\mathcal{G}_p^T$, and $E_t^T = \left\{\left(u_i^T, u_j^T\right)\right\}$ is the set of constructed edges based on user co-occurrence matrix. The co-occurrence matrix captures the similarities of user parking time habits.

Basic idea Our basic idea is that if two users have similar parking times and parking duration, they are likely to belong to the same identity group. For example, in an office region, the parking time and duration of the employees in the same company could be similar.

Building edges In detail, we create the parking time pattern graph in the following three steps. First, to identify the nodes of our graph, we collect the user list u_a^t that contains the users appeared before the given time T. Second, we create a co-occurrence matrix A of user time periods, with $A = \{0\}$ initially. We stipulate that if user i and user j park at the same time on the same day (with a time difference of no more than one hour) and the duration of the parking is also similar, then we set $a_{i,j} = a_{i,j} + 1, a_{i,j} \in A$. We define that the duration of two parking records are similar only when the following two restrictions are met: their lockdown time should not exceed an hour, and their parking duration should not exceed two hours. These two thresholds are set according to experiments. More details are shown in Sect. 6.4. Third, given the threshold γ, if $a_{i,j} \ge \gamma$, an edge shall be built between user i and user j, as shown in Eq. 3. In our experiments, we have 95,849 user pairs with similar temporal parking patterns, and each pair of similar users has 8 parking records on average. We randomly select 100,000 user pairs, and on average, each random user pair has 14 parking records.

$$e_{i,j} = \begin{cases} 1, & a_{i,j} \ge \gamma, i \ne j \\ 0, & otherwise \end{cases} \tag{3}$$

Parameter setting We also set the parameter γ by grid-search. We have strict constraints for users to have temporal connections. More details are introduced in Sect. 6.1.

3) Spatial connection graph For a given time T, we construct a spatial pattern graph $\mathcal{G}_s^T = \left(U^T, E_s^T\right)$. The sequence of the spatial pattern graphs for the past T time steps is G_p = $\{G_s^{t-T}, G_s^{t-T+1}, ..., G_s^{t-2}, G_s^{t-1}\}$, where t represents the current time and G_s^i represents the spatial pattern graph of the users arrived before day i. $U^T = \{u_a^T\}$ is the same list of users in $\mathcal{G}_p^T$ and $\mathcal{G}_t^T$, and $E_s^T = \left\{\left(u_i^T, u_j^T\right)\right\}$ is the set of edges built based on the users' spatial trajectories.

Basic idea Our idea in constructing spatial connection graph is to use users' spatial co-occurrence information to explore the connections of the trajectories of different users. For example, if a group of users often go to the same group of parking lots, there could be connections between these users due to the same trajectory.

Building edges In detail, we try to capture the users' spatial relevance through their trajectories. We establish a co-occurrence matrix $S = \left\{s_{i,j} | \forall s_{i,j} \in S, s_{i,j} = cs_{i,j}\right\}$, where

$cs_{i,j}$ is equal to the number of co-occurrence parking lots two users have co-appeared. We define the number of co-occurrence parking lots as the number of different common parking lots used by two users. When two users use the same parking lot multiple times, we still consider this as one co-occurrence parking lot. Here, we limit the time period within three months, which means that if user i parks in the same parking lot before or after three months compared to the parking time of user j, we do not count this record as a co-occurrence. We do not consider the factor of parking time because we already have a temporal connection graph for it. For the threshold, we set it to be two to keep the connection graphs dense and credible. Given the threshold γ, if $s_{i,j} \geq \gamma$, an edge shall be built between user i and user j, as shown in Equation 4. By building this graph, we can capture the spatial relations between users that the first two association graphs ignore. In our experiments, we find that there are 483,092 user pairs with similar spatial parking patterns, and on average, each user in these pairs has six parking records. We randomly select 600,000 random user pairs, and on average, each random user pair has 12 parking records.

$$e_{i,j} = \begin{cases} 1, & s_{i,j} \geq \gamma, i \neq j \\ 0, & otherwise \end{cases} \tag{4}$$

Parameter setting Here, we set the parameter γ the same as the parameter γ in the parking period pattern graph. The reasons that we do not change the way we set γ are as follows. First, as we mentioned before, grid-search is an effective automatic method that can help us to find the suitable parameter. Second, in our observation, the parking spaces of most users are stable, so the co-occurrence values of two users are small in most cases. Third, the process of setting parameters can be consistent.

In summary, we improve the significance of the users' characteristics by using their similarities in different correlations, so that the users' characteristics are more representative. Through creating the co-occurrence matrix and clustering, we capture users' connections and build different association graphs accordingly. Based on the three association graphs, we next construct the multi-dimensional user association graph. Similar ideas have also been applied in other domains, such as [73,77,90,97].

Graph fusion After obtaining the user association graphs from the three different dimensions, we next fuse the three association graphs to build a complete user association graph. We extend GCN, a widely used variant of convolutional neural network, to process the graph data. First, we obtain the node list and edges from three association graphs. Second, we traverse the edges of these graphs and fuse the three graphs, as shown in Fig. 10. We assume that if two users have an edge in graph G_t, G_p, or G_s, then, there shall be an edge in this fusion graph with edge weight adjusted appropriately. Third, we obtain the adjacency matrix A^T and the feature representation matrix D^T of the complete user association graph G^T at a given time T. Given time T, our original features include the amount of the user's last payment, the interval between the user's last two payments, the parking lot ID, coupon usage, and the parking time and date of the user's latest parking record. The feature x_a^T becomes 10 dimensions, represented as x_{ga}^T, when it passes through the first layer of GCN and becomes five dimensions x_{gga}^T after passing through the second layer of GCN. Different types of user parking relationships can be merged together.

Details. When constructing the three connection graphs, we define the weights as "$1/distance$" between two users for the user profile connection graph. For the temporal connection graph and spatial connection graph, we define the weight as their co-occurrence times. Accordingly, we add the weights from temporal connection graph, spatial connection graph, and profile contact graph together, and then normalize the weight. In our method, we consider these three graphs are of the same importance. Thus, we do not change the weight of each graph. However, we find that adjustment on the weights affects the performance of the final model. We conduct weight adjustment to demonstrate the influence of choosing different weight for each graph on the overall performance. We show the weight adjustment in Fig. 11. By default, the weight of each graph is 1/3. The x axis represents the weight of the corresponding graph w_g. The weights of the other two graphs are equally divided. When we increase or decrease the weight of a certain graph, the weights of the other two graphs will also decrease or increase, and the accuracy of the system will also change. Through experiments, we find that three graphs contribute equally to the overall performance. Therefore, we think that they are equally important.

We use a two-layer GCN, which can fully utilize the relations between neighbors efficiently. We choose GCN because it can capture the node relations in graphs via message passing between the nodes, which is very suitable for our parking payment application scenarios. In detail, in our parking payment behavior analysis, we treat each user as a node in the graph while their relations as edges. Furthermore, previous works [26,40,53] similar to this problem are also handled by GCN and can achieve good results. Therefore, we use GCN in payment behavior prediction. We also compare GCN with other methods such as CART, LSTM, and CNN, which is detailed in Sect. 6.2.

5.4 LSTM-based modeling

We propose an LSTM-based method in TR-GCN to model the temporal dependencies among users' historical transac-

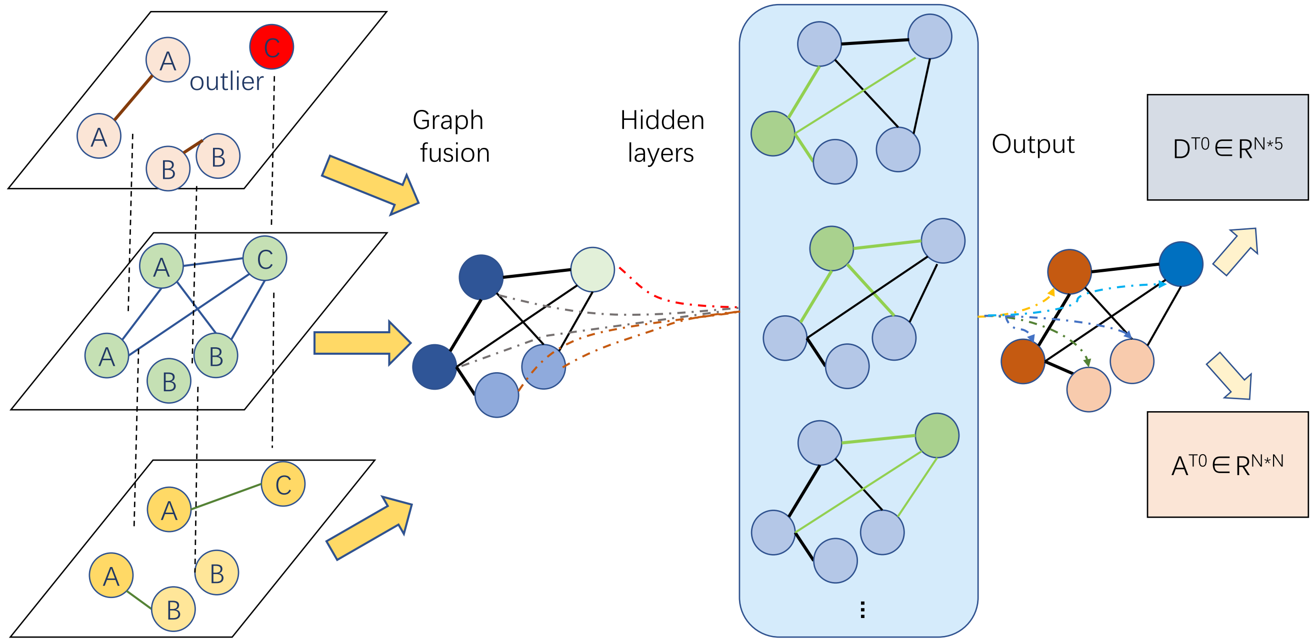
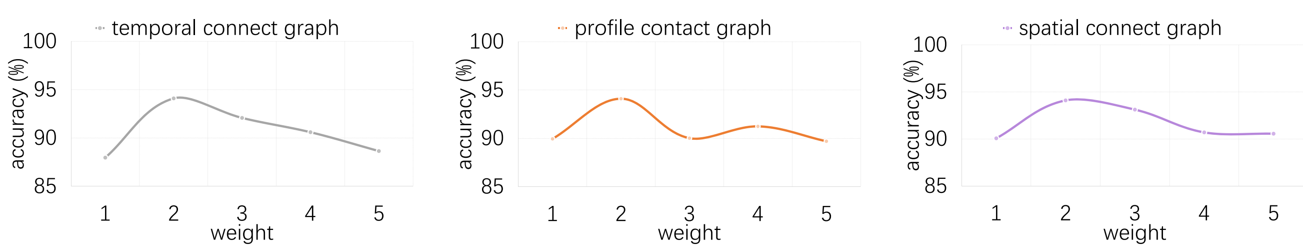


Fig. 10 Graph fusion process. "A," "B," and "C" on the left side represent different clusters in each association graph



(a) Portrait contact graph. **(b)** Temporal connection graph. **(c)** Spatial connection graph.

Fig. 11 Accuracy with weight adjustment

tions, including an LSTM layer and a fully connected layer. With LSTM, TR-GCN can learn the payment habits from the history of users, such as whether the user prefers to pay in the morning or afternoon. The platform also provide a reminder service detailed in Sect. 6.3. The reminder should also respect such habits and send notification in the time period users prefer to pay.

LSTM layer We use LSTM to capture the temporal relations among users' historical payment records instead of simply fusing them for three reasons. First, LSTM can capture the temporal information from historical events. Since LSTM has a long-term memory function, which can solve the problem of gradient disappearance or explosion by gating mechanism, LSTM can well utilize the temporal irregular historical payment data. Second, LSTM is a nonlinear neural network, which means that it not only captures the direct relations linear models can obtain, but also digs deeper into the hidden relationship between payment records, so as to improve the representation of the selected features. Third, LSTM has been proven to be suitable for parking behavior prediction [2,80]. Based on these analyses, we believe that LSTM is suitable for capturing the temporal relations in parking payment prediction.

Considering previous T step payment inputs of user i, which is $\left(x_i^{t-T+1}, x_i^{t-T+2}, \ldots, x_i^t\right)$, we denote the status of user i at time step t-1 and t as h_i^{t-1} and h_i^t, respectively. The temporal dependency between h_i^{t-1} and h_i^t can be modeled by Equation 5.

$$h_i^t = \left(1 - z_i^t\right) \circ h_i^{t-1} + z_i^t \circ \widetilde{h_i^t} \tag{5}$$

z_i^t and $\widetilde{h_i^t}$ are defined in Equation 6, where W_r, W_z, and $W_{\widetilde{h}}$ are payment parameters learnt by LSTM. $\bigoplus$ represents the concatenation operation, and $\circ$ stands for the Hadamard product. Then, the hidden payment state h_i^t obtained from LSTM is used as the input of our next fully connected layer.

$$\begin{cases} z_i^t = \sigma\left(W_z\left[h_i^{t-1} \bigoplus x_i^t\right]\right) \\ r_i^t = \sigma\left(W_r\left[h_i^{t-1} \bigoplus x_i^t\right]\right) \\ \widetilde{h}_i^t = \tanh\left(W_{\widetilde{h}}\left[r_i^t * h_i^{t-1} \bigoplus x_i^t\right]\right) \end{cases} \tag{6}$$

Fully connected layer To obtain better results, we adopt the method of adding a fully connected layer to carry out the final classification. In this part, the input of the fully connected layer is the three-dimensional hidden payment state h_i^t that we get from LSTM. To note that we do not add a normalization layer before the fully connected layer and we apply relu [56] as our activation layer because it can avoid vanishing gradient problems. There is only one hidden layer in this part and the output is two-dimensional because we need to get the probability. Similar idea has also been used by Zhang et al. [80].

Details. In our work, we use LSTM to integrate users' historical payment information. There are many kinds of temporal neural networks, such as TCN [37], RNN [49], and GRU [11]. We choose LSTM to process time series data for three reasons. First, LSTM is a powerful variant of RNN that can avoid serious problems like vanishing gradient. Second, LSTM provides us more space for adjusting parameters. Third, LSTM has been proved to be an effective model in shared parking behavior prediction [80]. Based on these reasons, we use LSTM in our work to integrate users' information. To further extract the features for a more precise prediction result, we add a single-layer fully connection network after the LSTM. In the fully connected layer, we set the number of neurons in the hidden layer to five by experience.

6 Evaluation

In this section, we first introduce the experimental setup, including our methodology and dataset. Second, we show our evaluation results with detailed analysis. Third, we analyze our graph fusion process.

6.1 Experimental setup

Methodology We use accuracy as the evaluation metric and compare our TR-GCN with the state-of-the-art payment prediction method [74] which applies CART model [63] in parking payment prediction, and the LSTM model without hidden user relationships.

In TR-GCN, we set the parameters mainly based on grid search [38], which we find powerful in the parameter settings. In both training and testing cases, T values represent different timestamps at day granularity. As to the threshold γ, if γ is too low, the connections between users are unstable and imprecise. In contrast, if γ is too high, the graph to be built will become too sparse and we cannot utilize all useful information. Hence, we set the range of γ from 0 to 5 (we find users rare to have connections more than five times). After exploration, we set γ to 2 to build the parking period pattern graph and the spatial connection graph. We leverage a two-layer GCN due to its effectiveness in modeling graph data. The dimension of the first layer of GCN is fixed to 10, and the dimension of the second layer is fixed to 5. The dimension of the hidden layer of LSTM is set to 52, and the dimension of the output layer of LSTM is set to 3. At last, we use a fully connected neural network with the dimension of the hidden layer set to 5.

The accuracy is defined in Eq. 7. According to [62], TP denotes the number of correctly predicted delayed payments, and TN denotes the number of correctly predicted on-time payments. FP denotes the number of orders mispredicted as delayed payment, and FN denotes the number of orders mispredicted as on-time payment.

$$accuracy = \frac{TP + TN}{FP + TP + FN + FP} \tag{7}$$

Platform TR-GCN has been deployed on a node with Intel Core i7-10750H CPU equipped with a NVIDIA GeForce RTX 2070 GPU.

Dataset We collect the payment record from 50 parking lots in nine cities. We randomly show ten parking lots in Table 1. These parking lots have various surroundings, including hospitals, commercial streets, residential areas, and so on. The dataset spans 36 months, from October 20, 2017, to October 20, 2020, with a total of 705,155 payment records. After data cleaning, we have 655,008 orders in total, with 451,474 orders belong to the case of a user with more than 10 records. There are 104,442 users in the dataset, with 10,144 users have more than 10 records. For each record, the attributes we extract include the order amount, the time interval from the previous order (0 if there is only one order), parking lot ID, coupon usage, membership, and date. We use 20% of our dataset as test set, and the remaining records as the training set. For users who have less than 10 records, we still use the traditional prediction method since TR-GCN relies on historical records.

6.2 Prediction accuracy evaluation

Accuracy We show the prediction accuracy of the fifty parking lots in Fig. 12, and we have the following observations. First, our TR-GCN achieves the highest accuracy of 91.2%, which is about 7.1% higher than the state-of-the-art method and 4% higher than the LSTM method. Second, our model performs well in all kinds of surroundings, which proves the practicality of our model. TR-GCN performs the best

Table 1 Parking lot information (50 parking lots in total)

Parking Lot	Record#	Surrounding	Location
P1	13446	Business, residential district	City A
P2	10122	Business	City B
P3	4872	Business	City B
P4	7990	Business	City C
P5	125409	Second-hand car mall	City D
P6	19326	Hospital	City D
P7	38599	Business	City E
P8	90627	Business, office	City E
P9	8374	Business, office	City E
P10	11911	Office	City F
P11	9216	Business	City C
P12	18370	Uptown	City D
P13	1769	Office	City C
P14	9621	Business	City B
P15	11315	Business	City F
P16	9940	Business	City E
P17	8942	Uptown	City F
P18	36079	Uptown	City A
P19	13332	Business	City G
P20	40926	Office	City C
...	...	...	...

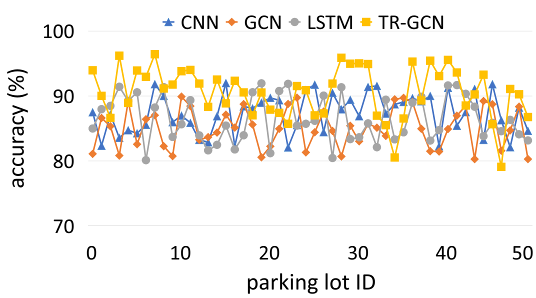


Fig. 12 Prediction accuracy

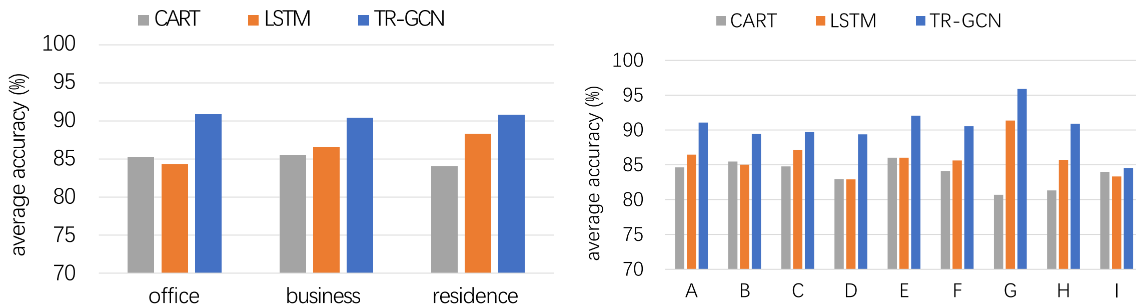
Table 2 Prediction analysis of different approaches

Approach	Accuracy	F1	Recall	Precision
TR-GCN	91.2%	65.9%	73.7%	53.1%
CNN	86.3%	39.8%	59.7%	30.0%
CART	84.1%	30.9%	63.6%	20.4%
LSTM	87.8%	42.0%	54.8%	28.6%

in 40 out of 50 parking lots. TR-GCN performs worse in those parking lots mainly because of the data distribution. For example, in parking lot P35, only 35 out of 16,282 orders are late payment orders. The two baseline models, LSTM and CART, tend to predict all orders as in-time payment, while our TR-GCN can detect more delayed payments. Third, compared with the traditional method that depends on feature engineering and the LSTM that needs a lot of users' historical data for training, TR-GCN utilizes the various hidden relations between users, which greatly increases its accuracy. Moreover, we measure the throughput of TR-GCN. TR-GCN handles 235 orders per second, which satisfies the current situation.

Prediction analysis We further analyze the overall indicators in Table 2. The *precision* is defined as "$TP/(FP + TP)$," which shows that TR-GCN can better avoid mispredicting on-time payment as delayed payment. The *recall* is defined as "$TP/(FN + TP)$," which shows that TR-GCN can better avoid mispredicting delayed payment as on-time payment. The $F1$ score is defined as "$2 \times (precision \times recall)/(precision + recall)$," which shows that TR-GCN achieves significant advantages over current strategies. Moreover, compared with the baseline, TR-GCN does not need to carry out special feature engineering processing, which greatly increases its adaptability.

Precision versus accuracy TR-GCN achieves a high accuracy score, which means that our prediction results are accurate and precise. However, accuracy represents all correctly predicted samples. Although accuracy is commonly used, it cannot fully meet all the requirements of classifi-



(a) Different types of parking lots.

(b) Different cities.

Fig. 13 Average accuracy

cation tasks, especially when data are unbalanced. Hence, we also analyze the results from the precision perspective. Precision describes the proportion of true positive examples divided by all positive examples, predicted by the classifier, that is, the ratio of accurate positive examples predicted by the classifier. In our scenario, we regard the positive examples as the delayed payments.

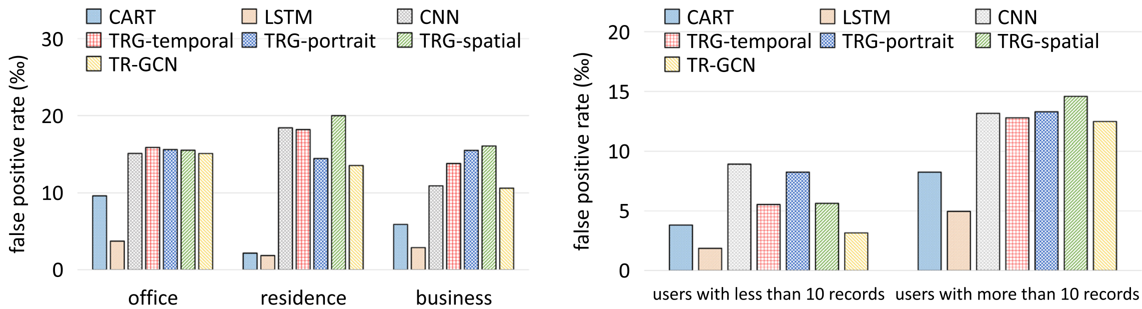
Considering the unbalanced data, we use several metrics including accuracy, F1, recall, and precision, to evaluate our experimental results. As shown in Table 2, our method, TR-GCN, achieves 91.2% in accuracy while the precision score is 53.1%, which is relatively low. The reasons are as follows. First, TR-GCN uses unbalanced data, so the precision result can be influenced. Second, we add weight adjustment with upsampling, so TR-GCN tends to predict more orders as delayed payments. Third, although the precision result is not high, we obtain 73.7% recall, which means we can find more than 70% of the delayed payments and is satisfying. Note that precision and recall are two opposite indicators.

Detailed analysis We also exhibit the average accuracies of different types of parking lots and cities, as shown in Fig. 13. We classify parking lots into three major categories in Fig. 13 a. The office, business, and residence categories include 9, 31, and 10 parking lots separately. We can see that TR-GCN achieves clear advantages in all cases. Figure 13 b shows the average accuracies of different cities, and TR-GCN achieves the highest accuracy in all cities. For city I, it has only five parking lots with 5,061 records in total spanning six months, and the connection graphs are also sparse, which is insufficient for training. Therefore, TR-GCN achieves minor advantage over CART. We report in Table 3 the F1 score in different cities. From Table 3, we can see that in most cities, TRGCN outperforms the other three models, but in cities with small number of orders and users, our model is affected by the sparsity of data. However, with the growing number of users and orders in all cities, we believe that our model can satisfy the need for all cities.

We also analyze the false positive rate in different user groups, as shown in Fig. 14. The false positive rate is defined as $FP/(FP + TN)$. FP denotes the number of orders mispredicted as delayed payment. TN denotes the number of correctly predicted on-time payments. We first analyze the false positive rates (FPR) of users in different types of parking lots, as shown in Fig. 14a. LSTM and CART models tend to predict all orders as being paid on time, while our TR-GCN model can detect more delayed payment orders. Note that the false positive rate of TR-GCN with three association graphs is lower than that with only one association graph (TRG-temporal, TRG-portrait, and TRG-spatial), which means that our method obtains more useful information making the prediction more accurate. Then, we analyze the false positive rates of user groups with more than and less than ten parking records, as shown in Fig. 14b. For users with less than ten orders, the false positive rate of TR-GCN is satisfying, which is 3.15‱. For the user group with more than ten orders, TR-GCN achieves 12.4‱ false positive rate. The reason why we measure FPR is that in our parking scenario, the proportion of users who have delayed payment is very small, which is less than 4.5%, and the distribution of users is unbalanced. In this case, it is inappropriate to use accuracy as the main indicator of the evaluation model. On the other hand, FPR focuses on examining the number of samples that are incorrectly predicted in negative samples. Therefore, as a comprehensive evaluation indicator, FPR can better reflect

Table 3 F1 in different cities

Approach (#Records)	CART	LSTM	CNN	TR-GCN
A (887944)	31.4%	44.2%	58.0%	62.1%
B (44742)	36.9%	42.7%	40.5%	55.3%
C (73332)	32.8%	47.0%	49.8%	60.6%
D (37696)	32.2%	41.9%	53.7%	70.1%
E (184835)	26.1%	51.9%	68.6%	73.4%
F (61572)	38.0%	45.1%	53.3%	59.6%
G (204002)	25.5%	46.6%	60.5%	76.0%
H (5908)	48.4%	38.2%	48.6%	56.2%
I (5124)	52.9%	43.0%	50.8%	52.6%



(a) Different types of parking lots. **(b)** Users with more than and less than ten records.

Fig. 14 False positive rates in different user groups

the ability of models to identify users who have delayed payments. Experiments in our paper show that the final FPR index of TR-GCN reaches 3.52%, which is much lower than the FPR indexes of the other models. Furthermore, this result shows that our model has a low probability of having disparity and does not hurt interests of certain groups. Accordingly, it is less likely to have wrong predictions and less likely to harm the interests of users that are wrongly predicted with delayed payment.

6.3 TR-GCN-based payment reminder

Challenges in reminder service Providing reminder service involves three challenges to handle. First, every user has her own payment habits and needs to be respected. Second, few users have more than three reminding records, which makes us lack of training data. Third, payment may be related to many factors, such as the day of the week.

Solution Our TR-GCN can solve the above challenges and be used to provide payment reminder service. We divide users into different user groups. Users in each group have similar payment habits, and we remind them uniformly. For users lacking records, we use rule-based methods to assist in reminder, such as reminding users at their average payment time. Additionally, we propose some relevant factors by experience, including week day and time. Finally, we verify them and involve them to our model. We also need to consider the data security as well as the training efficiency, given that the system needs to be placed in the cloud [51,52,79,82,84].

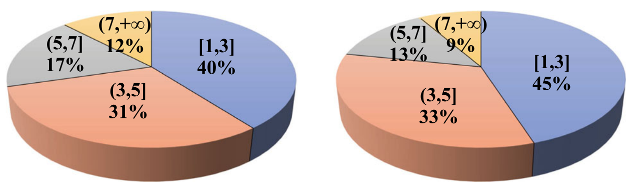
Effectiveness 4.43% users have delayed payment more than three days, and TR-GCN reminder makes 63.1% of these users pay within a day. Previously, 3.83% users have late payment more than five days, and TR-GCN reminder reduces this ratio by 13.8% to 3.32%. Moreover, 2.9% users have late payment more than a week, and TR-GCN reminder can make more than half of these users pay. Accordingly, TR-GCN can help these users establish good payment habits and help the intelligent platform recover 1.9% profit loss. We show our payment reminding results from November 11, 2020 to November 16, 2020, in Table 4. Approximately half of the delayed payments can be paid after being reminded. We have the following findings. First, if a user clicks on our payment reminder, then the user is very likely to pay (95.6% accuracy). Second, the payment also relates to the day of the week. For example, we send messages around nine on November 12 and November 14, but the results vary greatly. The reason is that November 12 is Thursday; users could be working and miss our message. November 14 is Saturday so they are available to pay. Third, after receiving our reminder, some users still need four hours to delay payment.

Payment behavior comparison before and after reminder We specify the distribution of users' payment time longer than one day with and without our reminder system in Fig. 15. Figure 15a shows the distribution before using our reminder service while Fig. 15b shows the distribution after using TR-GCN reminder. We can see that the proportion of the users who pay for more than five days has decreased significantly, which proves the effectiveness of TR-GCN.

In detail, before our reminder service, there are about 4% orders that belong to the late payment. We further analyze these late payment orders. Among these orders, 40% orders are paid within one to three days, 31% orders are paid within three to five days, and 29% orders are paid after five days, as illustrated in Fig. 15a. Among these orders, 40% orders are paid within one to three days, 31% orders are paid within three to five days, and 29% orders are paid after five days. In contrast to the original payment distribution, Figure 15b shows the benefits of our reminder service. The orders that belong to late payment are reduced to around 2.5% after we apply our reminder system. Specifically, we can see from Fig. 15b that the number of orders that are paid within one to five days increases, and the number of orders that are paid more than five days decreases. Additionally, 45% orders are paid within one to three days, and 33% orders are paid within three to five days. Hence, the orders that are paid for more than five days decrease to 21%. To conclude, the total number

Table 4 Effectiveness of payment reminder

Date	Sending time	#Sending orders	#User clicks	#Payments	Time range	Payment rate (%)
2020-11-11	12:23	84	52	50	12:23∼14:49	57.14
2020-11-12	9:00	252	118	116	9:23∼10:42	22.22
2020-11-13	12:50	98	62	69	12:50∼13:49	55.10
2020-11-14	9:21	116	82	92	9:21∼13:41	65.52
2020-11-15	14:49	414	127	106	14:49∼16:47	27.78
2020-11-16	13:36	125	83	60	13:36∼15:58	54.40
...	...	...	...	...	...	...



(a) Before TR-GCN reminder. **(b)** After TR-GCN reminder.

Fig. 15 User distribution in late payments of different periods before and after reminder

of late payment orders decreases. Our reminder system is effective for users who forget to pay for a long time.

Comparison with rule-based reminder We compare our TR-GCN-based reminder to the baseline of rule-based method, which was used by the parking platform before. In addition, we also add a simpler LSTM-based neural method for comparison. The baseline reminds users when their unpaid time exceeds the average payment time. The LSTM-based method utilizes LSTM to predict a reminding time. However, this method cannot utilize the hidden relationships and thus its payment rate is not as high as that of the proposed TR-GCN method. The platform adjusts the threshold every three months. We show the comparison results between TR-GCN and the baselines in Fig.16. Fig.16a shows the comparison of payment rates in different cities. On average, TR-GCN outperforms the rule-based method by 18%. Fig.16b shows the comparison of payment rates in different types of parking lots. TR-GCN outperforms the rule-based method by 14.3% on average and outperforms LSTM-based method by 11% on average. Figure 16c shows the payment rates using different methods in 50 parking lots. TR-GCN achieves the best payment rates in 44 out of 50 parking lots. Therefore, TR-GCN is an effective method that can be used in parking payment reminder systems.

Social impact We use the first ten parking lots to demonstrate the social impact on trends and unpaid rates in Fig. 17. First, we can see from Fig. 17a that the number of orders increases dramatically every year, but the unpaid rate also shows an increasing trend.

Second, we can see from Fig. 17b that for different parking lots, the unpaid rate is increasing year by year, and the difference between parking lots is relatively large. For example, for parking lot $P1$, the unpaid rate in 2020 even exceeds 10%, which seriously affects the profit of $P1$. Third, we can see that currently, the overall unpaid rate is not high, but we should not ignore such rising trends. Consequently, predicting customers' payment behavior and making corresponding reminders are of great significance.

6.4 Discussion

In this part, we summarize our contributions and discuss the biased issues and the applicability to other scenarios.

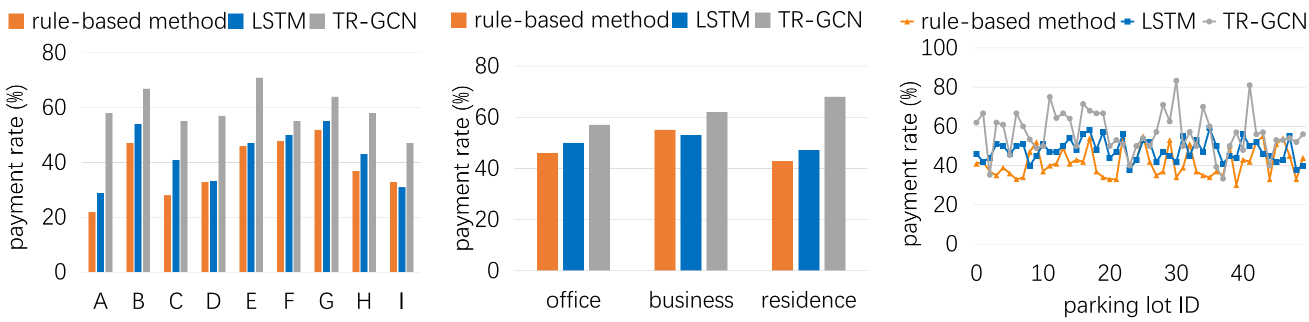
Summary of findings To better summarize our contributions and the effectiveness of our model, we list our findings learnt from our experiments.

First, our model is able to mine hidden information among users, and outperforms the state-of-the-art method [74]. Due to data imbalance, we apply four different metrics to evaluate our prediction results. Overall, our model achieves the best result.

Second, our GCN for modeling hidden relations plays an important role in payment prediction. To verify the effectiveness of GCN in delayed payment, we evaluate the model with GCN removed for comparison. Table 2 shows that without GCN, only LSTM cannot obtain satisfactory results. We believe that other hidden information can further improve the accuracy of payment prediction, which we leave to explore in the future.

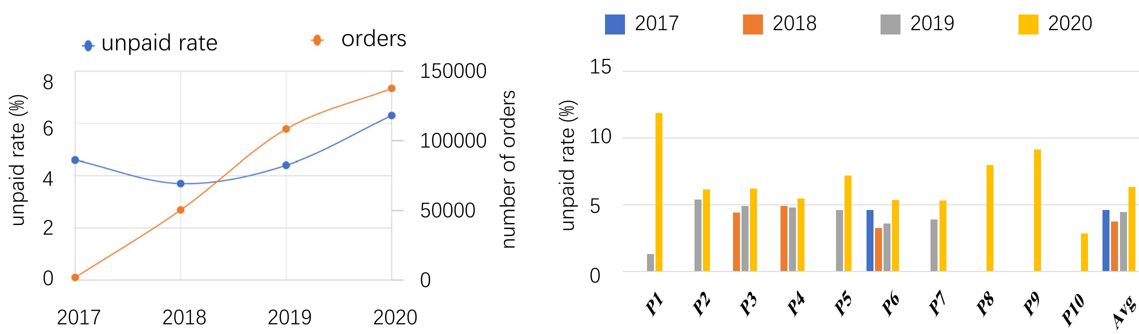
Third, TR-GCN not only predicts the delayed payments but also provides reminder service afterward. TR-GCN greatly saves the income loss of the shared parking platform and helps users to build better payment habits.

Biased issue Among the parking records, in-time payments account for the majority of the records. To identify the delayed payments, we have explored several possible solutions, such as upsampling and downsampling [22]. In our work, we combine resampling and weight adjustment to



(a) Different cities. **(b)** Different types of parking lots. **(c)** Different parking lots.

Fig. 16 Payment rate comparison



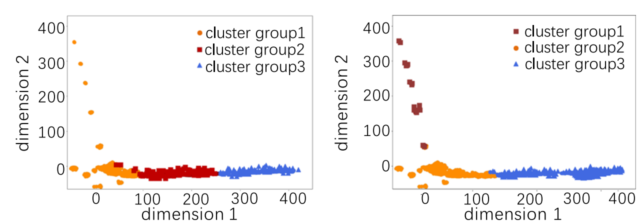
(a) Trends. **(b)** Unpaid rates.

Fig. 17 Social impact on trends and unpaid rates. P1, P2, P5, and P7 are built in 2019. P3 and P4 are built in 2018. P6 is built in 2017. P8, P9, and P10 are built in 2020

solve the biased data issue. We use upsampling instead of downsampling because it can utilize more data so that our model can be more precise and representative. First, we create a dictionary that records data from all users, including their features and the payment result. Second, for each user, we randomly select one of the user's delayed records as the upsampling data. Third, we insert the upsampling data into the user's record sequence. If the delayed payments consist of 30% of the user's records, we stop the upsampling process. We do not generate too many delayed records because this will degrade our model's ability to predict in-time payments. Additionally, we add weight adjustment to further improve TR-GCN's performance. We set the loss weight of mispredicted in-time payments as 1.5, while the loss weight of mispredicted delayed payments as 1.

Visualization The user relation graphs are changing dynamically. To visualize this changing process, we use t-SNE plot [45], which utilizes PCA [57] and KLD [36], to reduce the four dimensions to two dimensions, as shown in Fig. 18.

We can see that our clustering method successfully partitions the users into different groups. Moreover, the user relation graphs are changing dynamically. For example, Fig. 18a shows the clustering result on September 1, 2019,



(a) September 1, 2019. **(b)** September 1, 2020.

Fig. 18 Illustration for dynamic clustering results

while Fig. 18b shows the clustering result on September 1, 2020. We can see that the proportion of the three types of users has changed significantly. Our LSTM module can capture the characteristics of such changes.

Applicability to other scenarios Our model can be applied to other similar parking problems. For example, our method can be used in issuance of coupons for shared parking lots. TR-GCN successfully discovers hidden relations among users and builds association graphs without the personal information of users. With the hidden relationship mining in TR-GCN, we can identify the users with more parking choices, and then we can distribute coupons to these users to increase their favorability. Other factors, such as weather information, can also be integrated into the system.

Note that TR-GCN is not suitable for high-risk domains since models need to be interpretable in these scenarios. In a high-risk scenario, in which a positive classification means that access to a resource will be denied, incorrect positive classifications can hurt users, while incorrect negative classifications can hurt the service provider's business objectives. Regarding the interpretability, we have explored other models with better interpretability, such as CART (baseline). However, the prediction accuracy of these models is lower than that of TR-GCN. Hence, we regard TR-GCN as a good choice in the parking lot scenario, which is not high-stake.

For the limitations of the analysis, although the number of features in our model is limited, we propose a simple but effective solution to extract the hidden relationships. The model can successfully handle the input of users with less historical data. Although adding other demographic attributes can further extract hidden relations to reduce FPR, we find that the current solution can already generate qualified results that meet the requirements. Therefore, we leave the exploration for the other demographic attributes in the future.

Maximum processing capacity We perform a simulation to confirm the maximum throughput of our system. We let the system process millions of orders directly from memory for pressure testing, and find that it can process 720,349 orders per minute, which is much higher than the current data flow in peak hours. Accordingly, our system still can be efficient as the scale of workload grows. Moreover, it can be further accelerated by other optimizations such as [79,83].

Threshold setting for temporal connection graph In Sect. 5.3, we mention the thresholds for lockdown time (*threshold1*) and parking duration (*threshold2*). We demonstrate the impact of different threshold values in the Fig. 19. We evaluate nine different combinations of *threshold1* and *threshold2*, with both *threshold1* and *threshold2* range from 0.5 to 2. If both *threshold1* and *threshold2* are small, such as (0.5, 0.5), we find that this combination can incur the lowest accuracy. The reason is that the connection graph is so sparse and we can hardly obtain useful information from the connection graph. However, when the thresholds are both large, such as (2, 2), the accuracy is also low. The reason is that the scope of the time period is too large, and the extracted relationships can no longer hold. Hence, we set (1, 2) for *threshold1* and *threshold2*.

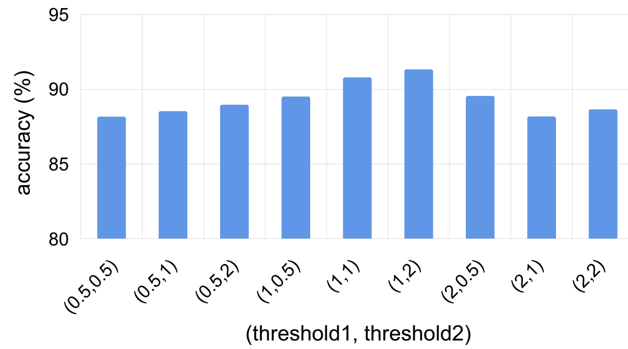


Fig. 19 Relation between threshold and accuracy

7 Related work

Parking payment prediction has become a hot research topic in recent days [6,19,30,74,80]. To our knowledge, this is the first work to mine user hidden connections in shared parking payment prediction under the circumstance of lacking users' personal information due to privacy issues. The closest work to ours is [74], which applied a simple machine learning method, LSTM, to predict late payments. Our work differs from [74] in many aspects. First, Xu et al. [74] did not solve the three challenges we mentioned in Sect. 4.2: the volume challenge, the variety challenge, and the timing challenge. They only used a small amount of data, and failed to consider timing issues and heavily relied on feature engineering. In contrast, our work uses a large amount of data and manages to store and read parking records efficiently. In addition, we abstract the association graphs between users from three hidden dimensions as well as considering the historical information. Our system can also be applied to other similar problems, such as payment prediction for shared bicycles.

Graph structured data modeling We use GCN to optimize feature vectors due to its applicability and efficiency. Previous works [26,27,40,53,71] often extract user connections by using their personal information like friend relationships or contact information. Jiang *et al.* [27] used graphical convolutional reinforcement learning to address the difficulty of rapidly changing relationships in a multi-agent environment. Xiong *et al.* [71] developed a new reinforcement learning framework for learning multi-hop relational paths. Additionally, many works used GCN and other variants of graph neural networks for different prediction scenarios [26,40,53].

User behavior modeling and feature embedding There are many works about relation modeling and feature embedding [7,20,23,46,54,67,94]. Guo et al. [23] proposed a solution for vehicle routing based on context and routing preferences. Paul et al. [54] extended the concept of semantic embedding of POIS (points of interest) Magdy et al. [46] applied automated machine learning techniques to design a simple and efficient cache-based negative sampling method, NSCaching. Zhou et al. [94] used the context of social users for video recommendations based on content and social relationships.

Time series modeling Many studies leverage time series data for prediction [12,16,18,28,39,41,95]. Li et al. [41] introduced the DCRNN (Diffusion Convolution Recurrent Neural

Network), which captures spatial dependence and temporal dependence. Donkers et al. [16] extended the RNN so that the extended RNN better adapts to the task of recommendation systems. Cini et al. [12] applied RNN to high-precision electricity demand forecasting, which is one of the main challenges for smart grids. Farha et al. [18] used a TCN-based model to classify video frames. Zhu et al. [95] proposed a spatial-temporal convolutional network for detecting human action boundaries. Li et al. [39] proposed a system that leverages historical trajectory to predict the best route on time to minimize on-road time.

8 Conclusion

In this paper, we present TR-GCN, which is a temporal relational graph convolutional network for payment prediction on shared parking lots. We first propose a novel approach to capture the multi-dimensional relations between users to construct user relational graphs according to user behaviors. Second, we construct a neuron-based model with GCN and LSTM to learn users' hidden relationships. Third, TR-GCN has been integrated into a real intelligent parking system. We input all historical records in TR-GCN for training and provide payment reminder service.

Experiments based on 655 thousand real payment records show that from 50 intelligent shared parking lots, our TR-GCN achieves 91.2% prediction accuracy, which is about 7% higher than the state-of-the-art method. Moreover, the TR-GCN-based payment reminder service generates large social impact by helping users establish good payment habits and reducing losses for parking platforms.

Acknowledgements This work is supported by the National Key R&D Program of China (2020AAA0105200), National Natural Science Foundation of China (No. 61732014, 62172419, U20A20226, and 61802412), Tsinghua University Initiative Scientific Research Program (20191080594), and GHfund A (No. 20210701). This work is also sponsored by CCF-Tencent Open Research Fund. Bingsheng's work is in part supported by a research project grant under NUS Centre for Trusted Internet and Community. Feng Zhang is the corresponding author of this paper.

References

1. Azayite, F. Z., Achchab, S.: The impact of payment delays on bankruptcy prediction: A comparative analysis of variables selection models and neural networks. In: International Conference of Cloud Computing Technologies and Applications, (2017)
2. Babaev, D., Savchenko, M., Tuzhilin, A., Umerenkov, D.: ET-RNN: Applying Deep Learning to Credit Loan Applications. In SIGKDD, (2019)
3. Barceló, P., Kostylev, E. V., Monet, M., Pérez, J., Reutter, J. L., Silva, J.-P.: The expressive power of graph neural networks as a query language. SIGMOD Rec, (2020)
4. Bergman, M., Milo, T., Novgorodov, S., Tan, W.-C.: Query-oriented data cleaning with oracles. In SIGMOD, (2015)
5. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach Learn. Res. **13**(2), (2012)
6. Chen, C., Chiang, R., Wu, T., Chu, H.: A combined mining-based framework for predicting telecommunications customer payment behaviors. Expert Systems Appl. (2013)
7. Chen, L., Koutris, P., Kumar, A.: Towards model-based pricing for machine learning in a data marketplace. In: International Conference on Management of Data, (2019)
8. Chen, Y.-L., Kuo, M.-H., Wu, S.-Y., Tang, K.: Discovering recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data. Electron. Commerce Res. Appl., (2009)
9. Cho, Y. S., Moon, S. C., Noh, S. C., Ryu, K. H.: Implementation of Personalized recommendation System using k-means Clustering of Item Category based on RFM. In: IEEE International Conference on Management of Innovation & Technology, (2012)
10. Chu, K.-F., Lam, A. Y., Loo, B. P., Li, V. O.: Public Transport Waiting Time Estimation Using Semi-Supervised Graph Convolutional Networks. In IEEE Intelligent Transportation Systems Conference, (2019)
11. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, (2014)
12. Cini, A., Lukovic, S., Alippi, C.: Cluster-based Aggregate Load Forecasting with Deep Neural Networks. In: 2020 International Joint Conference on Neural Networks (IJCNN), (2020)
13. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE transactions on neural networks and learning systems, (2017)
14. Dolatshah, M.: Cleaning crowdsourced labels using oracles for statistical classification. PhD thesis, Applied Sciences: School of Computing Science, (2018)
15. Dong, G., Liu, H.: Feature engineering for machine learning and data analytics. CRC Press, NY (2018)
16. Donkers, T., Loepp, B., Ziegler, J.: Sequential user-based recurrent neural network recommendations. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, (2017)
17. Fan, W., Xu, J., Wu, Y., Yu, W., Jiang, J.: GRAPE: Parallelizing sequential graph computations. In: Proceedings of the VLDB Endowment, (2017)
18. Farha, Y. A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2019)
19. Feng, N., Zhang, F., Lin, J., Zhai, J., Du, X.: Statistical Analysis and Prediction of Parking Behavior. In: IFIP International Conference on Network and Parallel Computing, pages 93–104. Springer, (2019)
20. Gao, J., Chen, J., Li, Z., Zhang, J.: ICS-GNN: lightweight interactive community search via graph neural network. PVLDB, (2021)
21. Gers, F. A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with LSTM. Neural Comput. **12**(10), 2451–2471 (2000)
22. Gopinath, R. A., Burrus, C. S.: On upsampling, downsampling, and rational sampling rate filter banks. IEEE Transactions on Signal Processing, (1994)
23. Guo, C., Yang, B., Hu, J., Jensen, C. S., Chen, L.: Context-aware, preference-based vehicle routing. VLDB J. **29**(5), 1149–1170 (2020)
24. Han, S., Kang, J., Mao, H., Hu, Y., Li, X., Li, Y., Xie, D., Luo, H., Yao, S., Wang, Y., et al.: Ese: Efficient speech recognition engine with sparse LSTM on FPGA. In: Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, (2017)
25. Hovanesyan, A.: Late payment prediction of invoices through graph features. Master thesis, TDelft University of Technology, (2019)

26. Huang, Z., Li, X., Ye, Y., Ng, M. K.: MR-GCN: Multi-Relational Graph Convolutional Networks based on Generalized Tensor Product. In IJCAI, (2020)
27. Jiang, J., Dun, C., Huang, T., Lu, Z.: Graph convolutional reinforcement learning. arXiv preprint arXiv:1810.09202, (2018)
28. Jin, Y., Guo, W., Zhang, Y.: A time-aware dynamic service quality prediction approach for services. Tsinghua Sci. Technol. **25**(2), 227–238 (2019)
29. Jou, R.-C., Chiou, Y.-C., Ke, J.-C.: Impacts of impression changes on freeway driver intention to adopt electronic toll collection service. Transp. Res. Part C Emerg. Technol., (2011)
30. Kim, J., Kang, P.: Late payment prediction models for fair allocation of customer contact lists to call center agents. Decis. Support Syst. **85**, 84–101 (2016)
31. Kim, J.-Y., Cho, S.-B.: Predicting repayment of borrows in peer-to-peer social lending with deep dense convolutional network. Expert Syst. **36**(4), e12403 (2019)
32. Kipf, T. N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, (2016)
33. Krishna, K., Murty, M. N.: Genetic K-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), (1999)
34. Krishnan, S., Wang, J., Wu, E., Franklin, M. J., Goldberg, K.: Activeclean: Interactive data cleaning for statistical modeling. PVLDB, (2016)
35. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, (2012)
36. Kullback, S., Leibler, R. A.: On information and sufficiency. Ann. Math. Statist. **22**(1), 79–86 (1951)
37. Lea, C., Flynn, M. D., Vidal, R., Reiter, A., Hager, G. D.: Temporal convolutional networks for action segmentation and detection. In CVPR, (2017)
38. Lerman, P.: Fitting segmented regression models by grid search. J. R. Statist. Soc. Ser. C (Applied Statistics), (1980)
39. Li, L., Zheng, K., Wang, S., Hua, W., Zhou, X.: Go slow to go fast: minimal on-road time route scheduling with parking facilities using historical trajectory. VLDB J. **27**(3), 321–345 (2018)
40. Li, S., Li, W., Wang, W.: Co-GCN for Multi-View Semi-Supervised Learning. In AAAI, (2020)
41. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926, (2017)
42. Lin, Z., Li, M., Zheng, Z., Cheng, Y., Yuan, C.: Self-Attention ConvLSTM for Spatiotemporal Prediction. In AAAI, (2020)
43. Liu, H., Lu, S., Chen, X., He, B.: G3: when graph neural networks meet parallel graph processing systems on GPUs. In: Proceedings of the VLDB Endowment, (2020)
44. Liu, Z., Chen, C., Yang, X., Zhou, J., Li, X., Song, L.: Heterogeneous graph neural networks for malicious account detection. In CIKM, (2018)
45. Maaten, L. V. d., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Rese (2008)
46. Magdy, A., Abdelhafeez, L., Kang, Y., Ong, E., Mokbel, M. F.: Microblogs data management: a survey. VLDB J. **29**(1), 177–216 (2020)
47. Manessi, F., Rozza, A., Manzo, M.: Dynamic graph convolutional networks. Pattern Recognit. **97**, 107000 (2020)
48. Meehan, J., Aslantas, C., Zdonik, S., Tatbul, N., Du, J.: Data Ingestion for the Connected World. In CIDR, (2017)
49. Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh annual conference of the international speech communication association, (2010)
50. Nainggolan, R., Perangin-angin, R., Simarmata, E., Tarigan, A. F.: Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) optimized by using the Elbow Method. In: Journal of Physics: Conference Series, (2019)
51. Pan, Z., Zhang, F., Li, H., Zhang, C., Du, X., Deng, D.: G-SLIDE: A GPU-Based Sub-Linear Deep Learning Engine via LSH Sparsification. IEEE TPDS, (2021)
52. Pan, Z., Zhang, F., Zhou, Y., Zhai, J., Shen, X., Mutlu, O., Du, X.: Exploring data analytics without decompression on embedded GPU systems. IEEE TPDS **33**(7), 1553–1568 (2021)
53. A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In AAAI, (2020)
54. Paul, D., Li, F., Phillips, J. M.: Semantic embedding for regions of interest. The VLDB J., (2021)
55. Pfohl, H.-C., Gomm, M.: Supply chain finance: optimizing financial flows in supply chains. Logisti. Res. **1**(3-4), 149–161 (2009)
56. Ramachandran, P., Zoph, B., Le, Q. V.: Searching for activation functions. arXiv preprint arXiv:1710.05941, (2017)
57. Ringnér, M.: What is principal component analysis? Nature Biotechnology, (2008)
58. Salha, G., Hennequin, R., Vazirgiannis, M.: Keep it simple: Graph autoencoders without graph convolutional networks. arXiv preprint arXiv:1910.00942, (2019)
59. Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference, (2018)
60. Sikka, N., Vila, C., Stratton, M., Ghassemi, M., Pourmand, A.: Sharing the sidewalk: a case of E-scooter related pedestrian injury. Am. J. Emergency Med., (2019)
61. Song, K., Jang, J., Shin, S., Moon, I.: Bivariate Beta-LSTM. In AAAI, (2020)
62. Stehman, S. V.: Selecting and interpreting measures of thematic classification accuracy. Remote Sens. Environ. **62**(1), 77–89 (1997)
63. Steinberg, D., Colla, P.: Cart: classification and regression trees. The top ten algorithms in data mining, (2009)
64. Sundararajan, A.: From Zipcar to the sharing economy. Harvard business Review, **1**(1), 1–2 (2013)
65. Tam, D. S. H., Lau, W. C., Hu, B., Ying, Q. F., Chiu, D. M., Liu, H.: Identifying Illicit Accounts in Large Scale E-payment Networks–A Graph Representation Learning Approach. arXiv:1906.05546, (2019)
66. Wang, C., Caja, J., Gómez, E.: Comparison of methods for outlier identification in surface characterization. Measurement, **117**, 312–325 (2018)
67. Wang, W., Guo, J., Li, Z., Zhao, R.: Behavior model construction for client side of modern web applications. Tsinghua Sci. Technol. **26**(1), 112–134 (2020)
68. Wen, Y. T., Yeh, P. W., Tsai, T. H., Peng, W. C., Shuai, H. H.: Customer Purchase Behavior Prediction from Payment Datasets. In: Eleventh ACM International Conference on Web Search & Data Mining, (2018)
69. Weytjens, H., Lohmann, E., Kleinsteuber, M.: Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. Electronic Commerce Research, (2019)
70. Wu, J., Si, S., Yan, H.: Reducing poverty through the shared economy: creating inclusive entrepreneurship around institutional voids in China. Asian Business Management, 1–29 (2020)
71. Xiong, W., Hoang, T., Wang, W. Y.: Deeppath: A reinforcement learning method for knowledge graph reasoning. arXiv:1707.06690, (2017)
72. D. Xu, W. Cheng, B. Zong, D. Song, J. Ni, W. Yu, Y. Liu, H. Chen, and X. Zhang. Tensorized LSTM with Adaptive Shared Memory for Learning Trends in Multivariate Time Series. In AAAI, (2020)
73. Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K.: Inductive Representation Learning on Temporal Graphs. arXiv:2002.07962, (2020)

74. Xu, Q., Zhang, F., Zhang, M., Zhai, J., Lin, J., Liu, H., Du, X.: Payment Behavior Prediction and Statistical Analysis for Shared Parking Lots. In: IFIP International Conference on Network and Parallel Computing, (2020)
75. Xu, Y., Yang, J., Du, S.: CF-LSTM: Cascaded Feature-Based Long Short-Term Networks for Predicting Pedestrian Trajectory. In AAAI, (2020)
76. Yang, S., Ma, W., Pi, X., Qian, S.: A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. Transp. Res. Part C Emerg. Technol. **107**, 248–265 (2019)
77. Yang, Y., Liu, Z., Tan, C., Wu, F., Zhuang, Y., Li, Y.: To stay or to leave: Churn prediction for urban migrants in the initial period. In WWW, (2018)
78. Yu, W., He, X., Pei, J., Chen, X., Xiong, L., Liu, J., Qin, Z.: Visually aware recommendation with aesthetic features. VLDB J. 1–19 (2021)
79. Zhang, F., Chen, Z., Zhang, C., Zhou, A.C., Zhai, J., Du, X.: An efficient parallel secure machine learning framework on GPUs. IEEE Trans. Parallel Distrib. Syst. **32**(9), 2262–2276 (2021)
80. F. Zhang, N. Feng, Y. Liu, C. Yang, J. Zhai, S. Zhang, B. He, J. Lin, and X. Du. PewLSTM: Periodic LSTM with Weather-Aware Gating Mechanism for Parking Behavior Prediction. In IJCAI, 2020
81. F. Zhang, Y. Liu, N. Feng, C. Yang, J. Zhai, S. Zhang, B. He, J. Lin, X. Zhang, and X. Du. Periodic Weather-Aware LSTM with Event Mechanism for Parking Behavior Prediction. TKDE, 2021
82. Zhang, F., Zhai, J., He, B., Zhang, S., Chen, W.: Understanding co-running behaviors on integrated CPU/GPU architectures. IEEE Trans. Parallel Distrib. Syst. **28**(3), 905–918 (2016)
83. Zhang, F., Zhai, J., Shen, X., Mutlu, O., Du, X.: POCLib: a high-performance framework for enabling near orthogonal processing on compression. IEEE Trans. Parallel Distrib. Syst. **33**(2), 459–475 (2022)
84. Zhang, F., Zhai, J., Shen, X., Wang, D., Chen, Z., Mutlu, O., Chen, W., Du, X.: TADOC: text analytics directly on compression. VLDB J **30**(2), 163–188 (2021)
85. Zhang, H., Wang, X., Chen, J., Wang, C., Li, J.: D2D-LSTM: LSTM-Based Path Prediction of Content Diffusion Tree in Device-to-Device Social Networks. In AAAI, (2020)
86. Zhang, J., Zhou, K., Li, G., Liu, Y., Xie, M., Cheng, B., Xing, J.: CDBTune+: An efficient deep reinforcement learning-based automatic cloud database tuning system. The VLDB Journal, (2021)
87. Zhang, L., Ma, D., Zhang, X., Yan, X., Wang, H.: Graph LSTM with Context-Gated Mechanism for Spoken Language Understanding. In AAAI, (2020)
88. Zhang, W., Liu, H., Liu, Y., Zhou, J., Xiong, H.: Semi-Supervised Hierarchical Recurrent Graph Neural Network for City-Wide Parking Availability Prediction. In AAAI, (2020)
89. Zhang, W., Miao, X., Shao, Y., Jiang, J., Chen, L., Ruas, O., Cui, B.: Reliable Data Distillation on Graph Convolutional Network. In SIGMOD, (2020)
90. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H.: T-GCN: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, (2019)
91. Zhao, Z., Chen, W., Wu, X., Chen, P. C., Liu, J.: LSTM network: a deep learning approach for short-term traffic forecast. IET Intell. Transp. Syst. **11**(2), 68–75 (2017)
92. Zheng, L., Li, Z., Li, J., Li, Z., Gao, J.: AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN. In IJCAI, (2019)
93. Zhong, W., Yu, N., Ai, C.: Applying big data based deep learning system to intrusion detection. Big Data Min. Anal. **3**(3), 181–195 (2020)
94. Zhou, X., Chen, L., Zhang, Y., Qin, D., Cao, L., Huang, G., Wang, C.: Enhancing online video recommendation using social user interactions. VLDB J. **26**(5), 637–656 (2017)
95. Zhu, H., Vial, R., Lu, S.: Tornado: A spatio-temporal convolutional regression network for video action proposal. In: Proceedings of the IEEE International Conference on Computer Vision, (2017)
96. Zhu, R., Zhao, K., Yang, H., Lin, W., Zhou, C., Ai, B., Li, Y., Zhou, J.: AliGraph: A Comprehensive Graph Neural Network Platform. Proc. VLDB Endow., (2019)
97. Zuo, Y., Liu, G., Lin, H., Guo, J., Hu, X., Wu, J.: Embedding temporal network via neighborhood formation. In SIGKDD, (2018)