Complex & Intelligent Systems (2023) 9:81–97
https://doi.org/10.1007/s40747-022-00781-y

ORIGINAL ARTICLE

RST-Net: a spatio-temporal residual network based on Region-reConStruction algorithm for shared bike prediction

Yanyan Tan[1,2] · Bin Wang[1] · Zeyuan Yan[1] · Haoran Liu[1] · Huaxiang Zhang[1,2]

Received: 13 August 2021 / Accepted: 1 May 2022 / Published online: 18 June 2022

Abstract

As a new form of public transportation, shared bikes have greatly facilitated people's travel in recent years. However, in the actual operation process, the uneven distribution of bicycles at each shared bicycle station has limited the travel experience. In this paper, we propose a deep spatio-temporal residual network model based on Region-reConStruction algorithm to predict the usage of shared bikes in the bike-sharing system. We first propose an Region-reConStruction algorithm (RCS) to partition the shared bicycle sites within a city into separate areas based on their geographic location information as well as bikes' migration trends between stations. We then combine the RCS algorithm with a deep spatio-temporal residual network to model the key factors affecting the usage of shared bicycles. RCS makes good use of the migration trend of shared bikes during user usage, thus greatly improving the accuracy of prediction. Experiments performed on New York's bike-sharing system show that our model's prediction accuracy is significantly better than that of previous models.

Keywords Spatio-temporal data mining · Urban computing · Gaussian mixture model cluster · Citywide bike usage prediction · Deep learning

Introduction

With growing air pollution and road congestion problems, many cities around the world are exploring greener and more convenient ways of getting around. One of the solutions that has been proposed and implemented during the past few years is the bicycle-sharing system (BSS) [1]. As a new mode of public transportation in the context of the "Internet+" and the sharing economy, the emergence of bicycle sharing has solved the "last mile" problem for residents and has greatly improved the utilization of public transportation [2]. It has also effectively alleviated traffic congestion on urban roads [3].

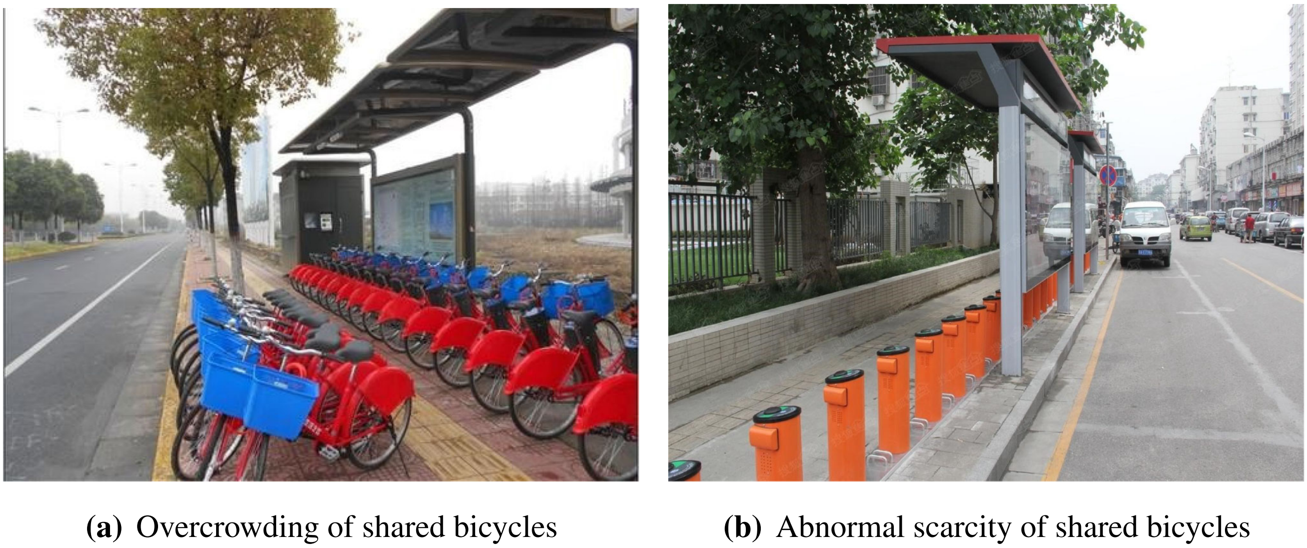
The BSS can be implemented in two main ways: a free-floating system and a station-based system [4]. In the free-floating system, bicycles are scattered in various areas of the city, and users can pick up and return the bicycles at any specified area. In comparison, the station-based system allows for better control of shared bicycles. First, the station maintains strict control over the use of shared bicycles, so the shared bicycles there are safer. Second, when electric bicycles are deployed in the station-based system, the station can act as a charging point for bicycle batteries. For this reason, station-based BSSs have always dominated the market [5].

The station-based BSS, however, does also have some shortcomings that seriously affect the user experience at times. One of these shortcomings is the unbalanced distribution of shared bicycles at each station [6]. As shown in Fig. 1a, some stations have large numbers of users returning bikes, which causes all of the bikes' stakes to be used. These stations thus cannot provide free stakes for subsequent users to use for performing return operations. As shown in Fig. 1b, some stations have very low numbers of shared bikes because large numbers of users are borrowing the bikes, so they cannot provide enough shared bikes for subsequent users. This problem of the unbalanced distribution of shared bikes mainly stems from the travel patterns of users [7], who generally leave their homes in the morning and go to work. This is when bikes located in residential areas are overused; conversely, bikes located at people's workplaces pile up in the morning, thus creating an oversupply. The exact opposite is true in the after-

✉ Bin Wang
binwangcs@163.com

[1] School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China

[2] Institute of Data Science and Technology, Shandong Normal University, Jinan 250358, Shandong, China


(a) Overcrowding of shared bicycles

(b) Abnormal scarcity of shared bicycles

Fig. 1 Unbalanced distribution of shared bicycles

noon. Both scenarios lead to an imbalance in the distribution of bikes at different stations at different times of the day in the city. In many BSSs, the problem of the uneven distribution of bikes is solved using trucks to manually transfer shared bikes between stations in a constant stream. This solution has two main drawbacks, though: (1) trucking is expensive and raises environmental concerns; (2) it is an afterthought, as the transfer of bicycles is always done after the problem occurs. This affects the proper functioning of the BSS. This solution is clearly not a long-term solution.

For this reason, researchers hope to solve this problem with the help of the effective prediction of the future usage of BSSs. Lin et al. [8] presented a bicycle-sharing strategy design problem that included a bicycle garage storage system and a model based on an inventory center. They took into consideration the number and locations of BSS stations, as well as the creation of bicycle lanes. Li et al. [9] proposed a level forecast model for predicting the number of future shared bicycles rented or returned. The model focuses more on macroscopic traffic flow in the BSS. This is an extremely important reference for research methods in system analysis and travel forecasting. The prediction model proposed in their literature first clusters bicycle stations using unified geographic grid clustering (GC) and the K-means clustering method in a two-layer process. Then, it uses a multi-similarity reference model to predict the number of rented and returned bicycles. However, the prediction accuracy still has much room for improvement due to the limitations of the multi-similarity reference model [10]. Jia et al. [11] combined affinity propagation (AP) clustering with a multi-similarity reference model (MSI) based on the work of Li to predict the number of borrowed and returned bicycles at a future time. The clustering results they obtained are more stable compared with previous studies because the number of clusters does not need to be specified at the time of clustering [12]. But in the actual use process the prediction of the model is limited by the time and space complexity of the model, which takes a lot of time. Yang et al. [13] use random forest (RF) to build a spatiotemporal dynamic network to evaluate and predict station and city bike demand. It should be noted that, random forests are not able to make predictions beyond the range of the training set data in the shared bicycle usage prediction task, which may lead to overfitting when modeling data with some specific noise. Also for shared bicycle prediction tasks with small data or low-dimensional data (data with few features), RF is ineffective. Therefore, the effectiveness of the spatio-temporal dynamic network created using RF has some limitations in the prediction task of shared bicycle usage.

In recent years, with the rapid development of deep learning, artificial neural networks have also been applied to shared bicycle usage prediction. Liu et al. [14] proposes a weight correlation network (WCN) to model the relationship among bike stations and dynamically group neighbouring stations with similar bike usage patterns into clusters, followed by artificial neural network (ANN) and Monte Carlo (MC) simulation to predict the over-demand probability of each cluster, looking at station-and cluster-level dimensions. However, because artificial neural networks do not take into account the time dependence in the model structure, it cannot fully capture the features of time series data [15,16].To overcome ANN's shortcomings, Pan et al. [17] train a deep long short term memory (LSTM) model with two layers to predict bike renting and returning by making use of the gating mechanism of LSTM and the ability to process sequence data of recurrent neural network(RNN). Their prediction accuracy has improved considerably, but because of traditional RNNs rely too much on predefined time lags to learn time series processing, it is difficult to find the optimal window size when modeling time series data. Models constructed based on RNNs alone do not capture temporal correlation well.

Based on the above study, in our work we present a deep spatio-temporal residual network model based on the Region-reConStruction algorithm (RCS) algorithm, called RST-Net, for predicting the usage of the urban BSS at a future time. We first propose an RCS algorithm to zone the shared bike sites in the city. The algorithm first clusters the sites into various regions based on their geographic location information via the Gaussian mixed-model clustering algorithm. Then, it calculates the "site→region" migration trend matrix of every site. Afterward, the Gaussian mixed-model clustering algorithm is used again to obtain more accurate regional classification results by combining each site's geographic location information with its "site→region" migration trend matrix. Unlike the previous clustering of the sites' geographic location information, the introduction of the "site→region" migration trend matrix can effectively capture the migration trends among each shared-bicycle site, which significantly improves the clustering accuracy. Following the RCS of shared-bicycle sites, we calculate the number of borrowed and returning bikes in each region based on the clustering results, and we obtain the inter-regional shared-bicycle borrowing and returning matrix. Based on this, we use the deep spatio-temporal residual network [18] to make a batch prediction of the number of shared bikes borrowed and returned in each region of the city in a future period. In addition, we design the residual network to model the traffic flow characteristics according to the short-term temporal impact and the long-term temporal impact. Then, we dynamically aggregate the two residual networks. Finally, we combine the effects of external factors, such as meteorology, to obtain the final prediction results.

Our contribution is in three main areas:

(1) The RCS algorithm is used to reconfigure the shared-bicycle sites. The RCS algorithm can effectively capture the bike migration trend among shared bike sites, and the sites are more representative after the regional reconstruction, which is more conducive to the prediction for bike borrowing and returning.

(2) Combining the RCS algorithm with the deep spatio-temporal residual network. While capturing the shared-bike migration trend, we identify four factors that influence the use of shared bicycles: the short-term temporal impact, the long-term temporal impact, the external factor impact (EFI) and the regional association impact (RAI). Then, we overlay deep spatio-temporal residual networks on top of RCS to capture these factors.

(3) Experimental results in the New York BSS show that our model performs well, and the prediction accuracy is greatly improved compared with the previous model.

Overview

In this section, we first define the terminologies used in this paper. Then, we describe the key characteristics of the BSS. Finally, we provide an overview of our proposed prediction model.

Problem definition

Definition 1 Trip. A trip $\mathrm{Tr} = (S_{\mathrm{o}}, S_{\mathrm{d}}, \tau_{\mathrm{o}}, \tau_{\mathrm{d}})$ is a usage record of shared bike, the starting station is denoted as S_{o}, consist of latitude $S_{\mathrm{o.lat}}$ and longitude $S_{\mathrm{o.lon}}$; the destination station is denoted as S_{d}, consist of latitude $S_{\mathrm{d.lat}}$ and longitude $S_{\mathrm{d.lon}}$ too; τ_{o} and τ_{d} are the time when the bike is borrowed at S_{o} and returned at S_{d} respectively.

Definition 2 Region. Location has many different definitions depending on the granularity and semantic meaning. In this paper, according to the clustering results in "Evaluation", we divide the shared bike stations into m regions, each region representing a group of shared bike sites with similar migration trends.

Definition 3 "site→region" migration trend matrix. "site →region" migration trend matrix is the bike migration matrix from each shared bike site to each region based on the clustering results, a "site→region" migration trend matrix in t is represented by a matrix $\mathrm{MT}_{t,k\times m}$, which means the number of shared bicycles from station S_i to region R_j in time t.

Definition 4 Meteorology. The meteorology $M_t = (w_t, p_t, v_t)$ is a vector corresponding to period t, where w_t, p_t and v_t stand for the weather condition, temperature and wind speed in t respectively.

Key features

Short-term time effects, long-term time effects, the regional association impact and some external factors usually influence the use of shared bicycles as a mainstream mode of public transportation. These factors are visually depicted in Fig. 2.

(1) Short-term impact (STI): The use of shared bikes at a certain moment is usually influenced by the use of shared bikes at a time close to that moment [19]. For example, the number of bikes returned at 8:00 a.m. to 9:00 a.m. is closely related to the number of bikes borrowed at 7:00 a.m. to 8:00 a.m. Similarly, the number of bikes borrowed at a site at this time is also closely related to the number of bikes returned

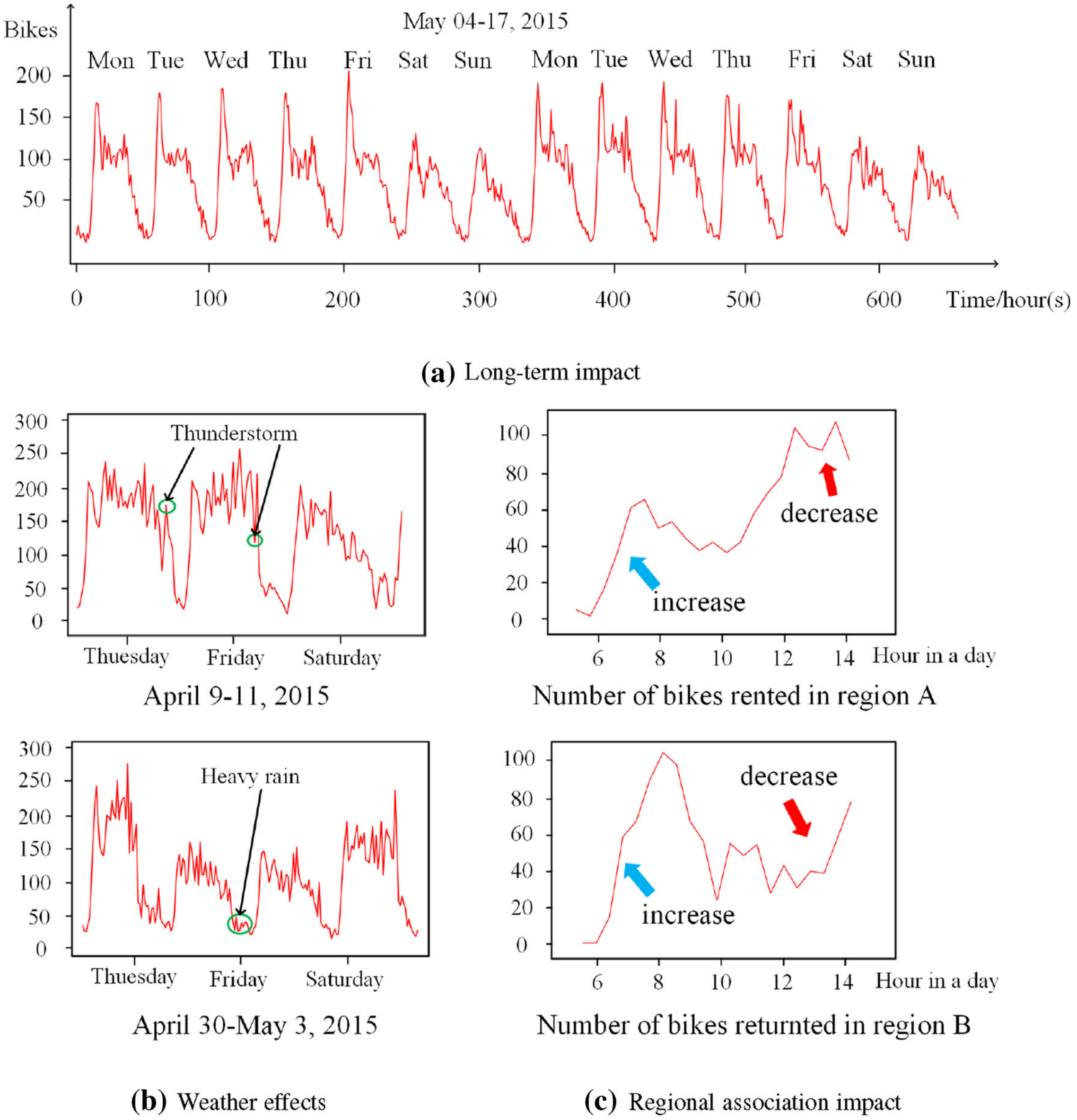


Fig. 2 Factors affecting the use of shared bicycles

at the previous time, as the number of bikes returned at a site directly determines whether the site has enough bikes for users to use at a following time period.

(2) Long-term impact (LTI): Shared bicycle usage at the same moment in a fixed period is usually similar as shown in Fig. 2a. Wednesday's shared bike usage is very similar to that of Tuesday and Thursday of this week and Wednesday of last week. These users generally have the same bicycle-using habits, such as commuting to work by bicycle, running to the park by bicycle, etc.

(3) External factors impact (EFI): First, as shown in Fig. 2b, the use of shared bicycles is highly susceptible to weather conditions [20]: in some bad weather conditions—such as too high or too low of a temperature, high wind speeds and rain—the number of shared bicycles used has a clear tendency to decline. Second, the usage of shared bicycles during holidays and double holidays is also significantly different from that in normal times, etc.

(4) Regional association impact (RAI): Many regions are interrelated in terms of shared-bike usage, which is mainly reflected in two aspects: geographic interconnection and regional function links. On the one hand, as shown in Fig. 2c, the number of bikes borrowed from a site at 6:00 a.m. has a direct impact on the number of bikes returned to the site after 6:00 a.m. Due to the close geographical connection, the flow of shared bicycles in adjacent areas will affect each other directly. On the other hand, a city can be divided into many areas according to their functions, such as work areas, living areas, and entertainment areas [21,22]. Many areas are directly related to one another in terms of transportation [23,24], such as work areas and living areas: during the work day, many workers go from their living areas to their work areas in the morning, and they return to the living areas to rest in the afternoon. Therefore, the number of borrowed bikes in the living areas in the morning will directly affect the number of returned bikes in the work areas. Meanwhile, the opposite is true in the afternoon: the number of borrowed bicycles in the work areas will directly affect the number of returned bicycles in the living areas.

Framework of our prediction model

We propose a deep spatio-temporal residual network model based on the RCS algorithm, called RST-Net, to predict the number of rented bikes and returned bikes in a following period. As illustrated in the Fig. 3, the entire prediction model consists of three parts:

(1) RCS model for shared bicycles: The distribution of shared-bike sites within the city is very disorganized, and the shared-bike usage data of many sites is very sparse, so it is not necessary to predict each site in the city individually. In this paper, we propose an RCS algorithm for the region classification of shared bike sites in a city. The algorithm first clusters all sites into different regions based on their geographic location information via the Gaussian mixed-model clustering algorithm. It then calculates the "site→region" migration trend matrix of each site. Finally, it combines the geographic location information and the "site→region" migration trend matrix for clustering to obtain more accurate regional classification results. Unlike the previous clustering of the geographic location information of the bicycle sites, the introduction of the "site→region" migration trend matrix can capture the bicycle migration trends among the shared bicycle sites, which can significantly improve the clustering accuracy.

(2) Deep spatio-temporal residual networks based on RCS: After RCS, we use the deep spatio-temporal residual network (ST-ResNet) to make a batch prediction of the number of bikes borrowed and returned in each region of the city at a coming time. Based on the STI and the LTI, we design a residual convolution unit to model the traffic flow characteristics. Then, we dynamically aggregate the two residual networks and combine them with meteorological factors to obtain the final prediction results. The details are described in "Deep spatio-temporal residual networks based on Region-reConStruction algorithm".

Here is the structure of the remainder of this paper: "Preparation work" describes the methods we need to use, "Proposed algorithm" provides a detailed description of the algorithm proposed, the experimental part is in "Evaluation" and "Conclusion" provides the conclusion of the thesis.

Preparation work

Gaussian mixture model

The Gaussian mixture model (GMM) uses the Gaussian probability density function (normal distribution curve) to accurately quantify things and to decompose a thing into a number of Gaussian probability density functions (normal distribution curve) based on the formation of the model [25]. Each GMM is composed of K Gaussian distributions, each Gaussian is called a "component" and these components are added together linearly to form the probability density function of the GMM as shown in Eq. (1):

$$p(x) = \sum_{k=1}^{k} p(k)p(x|k) = \sum_{k=1}^{k} \pi_k N(\mu_k, \Sigma_k) \tag{1}$$

where K components of GMM correspond to K clusters, π_k is the impact factor of each Gaussian distribution (component) on data points, μ_k is the mean value of each class, Σ_k is the covariance matrix.

Now there are N data points and assume that they obey a certain distribution (denoted as $p(x)$), now to determine the values of the set of parameters π_k, μ_k, Σ_k inside. The probability distribution it determines generates these given data points with maximum probability, and this probability is actually shown in Eq. (2):

$$p(x) = \prod_{i=1}^{N} p(x_i) \tag{2}$$

This product is called the likelihood function. The probability of a single point is usually very small. If many very small numbers are multiplied together, this can easily cause a floating point overflow in the computer. This leads to taking the logarithm of $p(x)$ and converting the product to a summation as shown in Eq. (3):

$$p(x) = \sum_{i=1}^{N} \log p(x_i) = \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \Sigma_k) \right\} \tag{3}$$

This yields the log-likelihood function, which is then maximized by finding a set of values for the parameters of π_k, μ_k and Σ_k such that the likelihood function takes its maximum value.

Because the logarithmic function has a summation inside, the maximum value cannot be found directly by solving the equation directly. To solve this issue, researchers try to randomly select points in the GMM. This process is divided into two steps: the E step and the M step (i.e., the EM algorithm [26]). The EM algorithm is a maximum likelihood

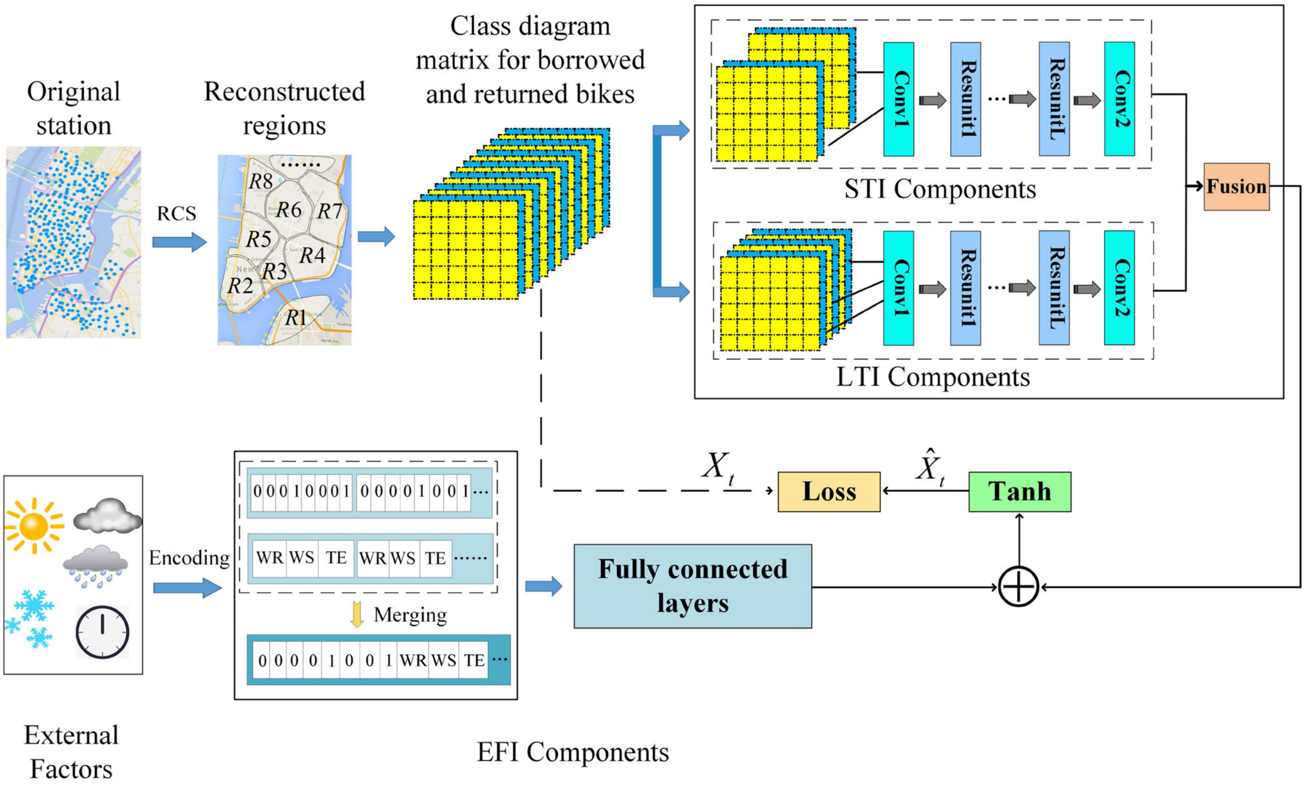


Fig. 3 The framework of RST-Net

estimation method for solving the parameters of a probability model from incomplete data or data sets with missing data (the presence of hidden variables). The E step uses the existing estimates of the hidden variables to calculate their maximum likelihood estimates; meanwhile, the M step maximizes the maximum likelihood values found in the E step to calculate the values of the parameters, and it iterates until convergence.

EM algorithm flow:

(1) Estimate the probability of the data that each component is generating (not the probability that each component will be selected): For each datum x_i, the probability that it is generated via the k-th component is:

$$\gamma(i,k) = \frac{\pi_k N(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i|\mu_j, \Sigma_j)} \tag{4}$$

where $N(x_i|\mu_k, \Sigma_k)$ is the posterior probability:

$$N(x_i|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right\} \tag{5}$$

(2) The values of the parameters of μ and Σ can be obtained via maximum likelihood estimation by taking the derivative and making the parameters equal to 0:

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k) x_i \tag{6}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k)(x_i - \mu_k)(x_i - \mu_k)^T \tag{7}$$

where $N_k = \sum_{i=1}^{N} \gamma(i,k)$, and π_k can be estimated as N_k/N,

(3) Repeat the first two steps of the iteration until the value of the likelihood function converges.

Gaussian mixture model cluster

The GMM cluster is the "inverse process" of generating data samples based on the GMM: the number of clusters (K), the parameters (μ) (i.e., mean vector), the covariance matrix (Σ) and the weights (π) of each mixed component are derived via a parameter estimation method, and each multivariate Gaussian distribution component corresponds to a cluster after clustering [27]. When the parameter estimation process is completed, for each sample point, the posterior probability of belonging to each cluster is calculated according to Bayes's

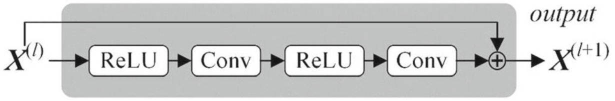


Fig. 4 Residual deep convolutional neural network

theorem, and the sample is classified with the cluster with the largest posterior probability. Compared with clustering methods such as K-means, which directly provides a cluster division of sample points, the GMM, which gives the probability that a sample point belongs to each cluster, is called soft clustering [28].

Deep spatio-temporal residual networks

The deep spatio-temporal residual network (ST-ResNet) is a neural network model with an end-to-end structure that designs neural networks based on the unique characteristics of spatial-temporal data. Zhang et al. proposed this in 2016 [18]. The ST-ResNet first grids the entire city region [29] and then uses a convolution-based residual network to mine the intra-city local area and its surrounding area. It has excellent performance in traffic flow prediction.

The ST-ResNet consists of four main components: the closeness component, periodicity component, trend component and external component. These are used to model four characteristics of traffic data, such as the temporal proximity, periodicity, trend and external factors affecting the traffic flow.

Specifically, the deep spatio-temporal residual network first converts the number of crowd inflows and outflows [30] for the entire city into an image-like two-channel matrix based on a given time interval. Then, the time axis is divided into three segments representing recent time, recent history and distant history, respectively. The two-channel flow matrix in each time interval segment is fed into the first three components, which are used to model each of the three temporal attributes mentioned above: the proximity, periodicity and trend. The first three components are composed of a network structure of a convolutional neural network connecting a sequence of residual units as shown in Fig. 4, with the specific description given by Eq. (8):

$$X_c^{(l+1)} = X_c^{(l)} + F(X_c^l; \theta_c^l), l = 1, \dots, L \tag{8}$$

The learnable parameters in the l-th residual unit is denoted by θ, and $F(\bullet)$ is the residual function. A convolutional layer is superimposed at the top of the L-th residual unit; thus, each component is composed of a number of convolutional layers and residual units. The outputs of these three components are X_c^{l+2}, X_p^{l+2} and X_q^{l+2}. respectively. This network structure captures the spatial dependence between nearby and faraway regions.

In the external component, the ST-ResNet manually extracts some features from external data, such as weather conditions and events, and feeds them into a two-layer fully connected neural network. The output of the external component is represented by X_{Ext}. Furthermore, the ST-ResNet fuses the outputs of the first three components—X_c^{l+2}, X_p^{l+2} and X_q^{l+2}—by means of parameter matrix-based fusion as shown in Eq. (9):

$$X_{\text{Res}} = W_c \circ X_c^{(L+2)} + W_p \circ X_p^{(L+2)} + W_q \circ X_q^{(L+2)} \tag{9}$$

where $\circ$ is the Hadamard product (i.e., element-wise multiplication), and where W_c, W_p and W_q are the parameters used to adjust the degree of influence by proximity, period and trend, respectively. They assign different weights to the results of different components in different regions; X_{Res} is then fused directly with the output of external component X_{Ext}. The predicted value, $\hat{X}_t$, for the t-th time interval is defined as:

$$\hat{X}_t = \tanh(X_{\text{Res}} + X_{\text{Ext}}) \tag{10}$$

where tanh is the hyperbolic tangent, ensuring that the output value is between − 1 and 1.

Proposed algorithm

Region reconstruction model for shared bicycles

The main reasons for our regional reconstruction of urban shared-bike sites are as follows:

(1) The distribution of shared-bicycle sites within the city is very disorganized [31], and the shared-bicycle data of a single site is not very regular, which makes it difficult to make modeling predictions.
(2) Due to the remote locations of some shared-bike sites and the relatively low bicycle usage there, the bike-sharing usage data of many sites are very sparse. In addition, due to some force majeure, the data collected via the sensors at some of the sites may be null at some moments, resulting in a lack of corresponding data [32];
(3) Many shared-bicycle stations are interconnected [33]. When one site is full of bicycles, if a user needs to return a bicycle, he or she must go to a nearby site to return the bicycle; similarly, when one site is in high demand and no bikes are available to borrow, the user must go to a nearby site. In addition, once an accident occurs that affects bike use at one station, this usually affects many shared bike stations in the immediate area.

Therefore, predicting shared-bicycle usage at individual sites in the city is not necessary. We consider using the RCS algorithm to reconstruct the shared-bicycle sites in the city in some regions. The algorithm first clusters all sites into different regions based on their geographic location information via the GMM clustering algorithm. Then, it calculates the bike migration matrix from each site to each region based on the clustering results, which is called the "site→region" migration trend matrix. Finally, it combines the sites' own geographic location information and the obtained migration trend matrix to obtain a more accurate region classification. Unlike the previous clustering of the geographic location information of shared-bicycle sites, the introduction of the "site→region" migration trend matrix is good for capturing the bike migration trends among shared-bicycle sites.

> **Step 1: Initial reconstruction** (in the first level) Apply GMM clustering on shared bike stations based on their geographical locations. Consequently, all stations are divided into some classes, each class is labeled as one region.
>
> **Step 2: Iterated reconstruction** (in the second level)
>
> > **Step 2.1**: Based on the reconstruction results at Step 1, compute the "site→region" migration trend matrix of each station.
> >
> > **Step 2.2**: Re-perform GMM clustering on shared bike stations based on their geographical locations and migration trend matrices. The new reconstruction results are obtained.
> >
> > **Step 2.3**: Repeat Step 2.1 and Step 2.2 until the reconstruction results are no longer changed, or the number of iterations gets the maximal limit.

Figure 5 visually portrays the RCS procedure. In the first step of the algorithm, the RCS algorithm clusters the individual sites by their geographic location information only:

$$\begin{bmatrix} S_1.\text{lat} & S_1.\text{lon} \\ S_2.\text{lat} & S_2.\text{lon} \\ \vdots & \vdots \\ S_n.\text{lat} & S_n.\text{lon} \end{bmatrix}_{n\times 2}$$

where n is the number of stations, $S_{x.\text{lat}}$ and $S_{x.\text{lon}}$ represent the latitude and longitude information of each shared bike station respectively. $\{R_k^0\}_{k=1}^m$ are the generated regions, m is the number of regions. In the second step of the algorithm, a value $\left\|\text{MT}_x^t\right\|_F$ is added to the latitude and longitude information of the site, which is represents the migration trend of station S_x in the t-th iterated.

Algorithm 1: Region-reConStruction algorithm

```
Input:
Stations $\{S_i\}_{i=1}^n$, trip history $\{T_{r_i}\}_{i=1}^H$, iterations number $T$,
Number of regions $k$, $t = 0$.
Output:
Regions $\{R_k\}_{k=1}^m$
Step 1: Initial reconstruction
$\{R_k^0\}_{k=1}^m = \text{GMM}\ (\{S_i\}_{i=1}^n)$; /* Perform GMM on bike stations
based on geographic position features */
Step 2: Iterated reconstruction
while $t \leq T$ do
    $t++$;
    Calculate/update $\{MT_i^t\}_{i=1}^n$ based on $\{R_k^{t-1}\}_{k=1}^m$; /*
    $\{MT_i^t\}_{i=1}^n$ is the migration tend matrices of stations under the
    $t$-th iteration */
    $\{R_k^t\}_{k=1}^m = \text{GMM}\left(\{S_i\}_{i=1}^n, \|MT_i^t\|_F\right)$; /* Use geographical
    locations and migration trends to re-reconstruction, $\|MT_i^t\|_F$
    is the Frobenius norm of $R_i^t$ */
    if $\left(\{R_k^t\}_{k=1}^m == \{R_k^{t-1}\}_{k=1}^m\right)$ then
        return;
    end
end
```

$$\begin{bmatrix} S_1.\text{lat} & S_1.\text{lon} & \|\text{MT}_1^t\|_\text{F} \\ S_2.\text{lat} & S_2.\text{lon} & \|\text{MT}_2^t\|_\text{F} \\ \vdots & \vdots & \vdots \\ S_n.\text{lat} & S_n.\text{lon} & \|\text{MT}_n^t\|_\text{F} \end{bmatrix}_{n\times 3}$$

It must be noted that in Step 2.2 of Algorithm 1, we have a problem (i.e., how to add migration trend information to the clustering process). Our solution is to transform the migration trend matrix into a trend array feature, which is described below:

We introduce the Frobenius norm [34] in the data processing step. Assume that MT is the migration trend matrix for a station. MT_{ij} represents the migration information (expressed in terms of the number of returned bikes) from the current station to the j-th region during the i-th hour. The Frobenius norm of $\text{MT} \in \Re^{l\times m}$ is defined as:

$$\|\text{MT}\|_\text{F} = \sqrt{\sum_{i=1}^{l}\left(\sum_{j=1}^{m}\left|A_{ij}\right|^2\right)} \tag{11}$$

where m is the number of regions, and l is the time interval set in the calculation of the migration trend (here, l is set to one hour) [35]. Take one day as an example: the training data period is 24 h, and the number of rows in matrix MT is 24. $\left\|\text{MT}_1^1\right\|_\text{F}$ and $\left\|\text{MT}_1^t\right\|_\text{F}$ represent the Frobenius norm of the migration trend matrix of station S_x used in the first and t-th GMM clustering with geographical locations and migration trend information. Figure 5 provides an example of a station's

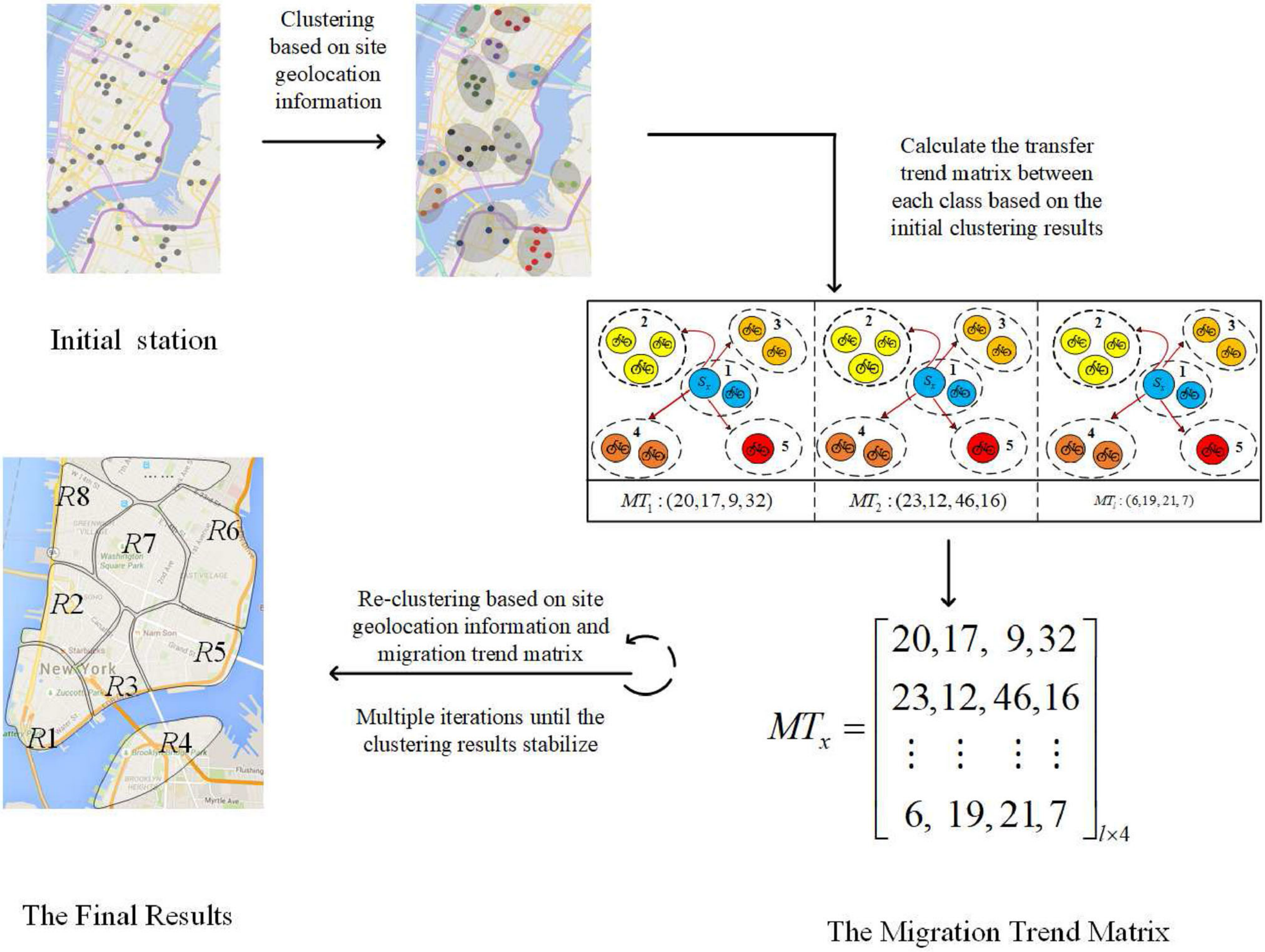

Clustering based on site geolocation information
Initial station
Calculate the transfer trend matrix between each class based on the initial clustering results
2
3
1
4
5
S_x
MT_1 : (20,17,9,32)
MT_2 : (23,12,46,16)
MT_l : (6,19,21,7)
R8
R7
R6
R2
R5
R3
R1
R4
Re-clustering based on site geolocation information and migration trend matrix
Multiple iterations until the clustering results stabilize
$MT_x = \begin{bmatrix} 20,17, & 9,32 \\ 23,12, & 46,16 \\ \vdots & \vdots \\ 6, 19, & 21,7 \end{bmatrix}_{l\times 4}$
The Final Results
The Migration Trend Matrix

Fig. 5 Region-reConStruction diagram

migration trends. As shown in Fig. 5, the city has 10 stations, and all of the stations are reconstructed into five regions. For station S_1 in region R_1, the number of shared bicycles at this site that are returned to the four other regions during time period t_1 is denoted as $\text{MT}_1(20, 17, 9, 32)$. Similarly, the number of shared bicycles at this site that are returned to the four other regions during time period t_2 is denoted as $\text{MT}_2(23, 12, 46, 16)$. The time period during the t_l hour is denoted as $\text{MT}_l(6, 19, 21, 7)$. Therefore, the migration trend matrix of station S_1 can be denoted as a matrix, which represents the historical migration trends of station S_1 during the past l hours:

$$\text{MT}_1 = \begin{bmatrix} 20 & 17 & 9 & 32 \\ 23 & 12 & 46 & 16 \\ \vdots & \vdots & \vdots & \vdots \\ 6 & 19 & 21 & 7 \end{bmatrix}_{l\times 4}$$

Following the regional reconfiguration of shared-bicycle stations, we calculate the number of shared bicycles that are borrowed and returned in each region based on the reconstruction results to get the inter-regional shared-bicycle borrowing and returning matrix.

Deep spatio-temporal residual networks based on Region-reConStruction algorithm

After reconstructing the shared-bicycle stations in the city and obtaining the inter-regional bicycle borrowing and returning matrix, we fuse it with the deep spatio-temporal residual network to predict the number of bicycles borrowed and returned at a future time.

A description of the key characteristics that influence the use of shared bicycles has been given in “Key features”. In this section, we use a deep spatio-temporal residual network based on the RCS algorithm to model three factors that influence bike-sharing usage—short-term temporal influence, long-term temporal influence and the influence of external factors—in the following steps:

Step 1: Calculate the inter-regional shared-bicycle borrowing and returning matrix. Based on the results of the RCS algorithm, the bicycle borrowing and returning matrix between regions is calculated with a time interval of one hour. The bicycle borrowing and returning matrix is similar to a picture with different pixel values. The number of bicycles borrowed and returned among different regions in a hour is composed of a two-dimensional array representing the number of bicycles that users have borrowed and the number of bicycles that users have returned in each region, respectively. All of the time intervals of the bicycle borrowing and returning matrix are stacked together to form a four-dimensional bicycle renting and returning matrix. Specifically, for four-dimensional array A, $A[0]$ represents all regions, $A[1]$ represents the number of rented bikes in each region, A[2] represents the number of returned bikes in each region and $A[3]$ represents the time series. A visual depiction is given in Fig. 6.

Step 2: Extraction of time segments. As shown in Fig. 7, according to the given predicted time slots, the renting and returning data corresponding to the two time slots are extracted on top of the regional renting and returning matrix: short-term time slice X_{Sti} and long-term time slice X_{Lti}. The details of the extraction are shown below:

$$X_{\mathrm{Sti}} = [X_{t-l_{\mathrm{sti}}}, X_{t-(l_{\mathrm{sti}}-1)}, \ldots, X_{t-1}] \tag{12}$$

$$\begin{aligned} X_{\mathrm{Lti}} = [&X_{t-l_{\mathrm{lti}}*7*24}, X_{t-(l_{\mathrm{lti}}-1)*7*24},\\ &X_{t-(l_{\mathrm{lti}}-2)*7*24}, \ldots, X_{t-7*24}] \end{aligned} \tag{13}$$

In Eqs. (12) and (13), l_{sti} denotes the length of the short-term time slice, and l_{lti} denotes the length of the long-term time slice. We use these two formulas to extract the short-term time series and long-term time series of the matrix of the number of bikes borrowed and returned. Using Eq. (13) as an example, we assume that the time of the historical data set is 100 weeks. Through this formula, we can use a week as a unit of measure to extract the matrix of the number of borrowed and returned bikes from the initial time to the current time.

Step 3: Constructing convolutional neural networks. As mentioned above, in the bicycle-sharing system, complex interactions take place between many stations in terms of geographic location, which is also the case in the BSS following region reconfiguration. In an inter-regional shared-bike renting and returning matrix, many related regions are in different locations: some are close to one another, and some are far away from one another. In this paper, we hope to capture the spatial dependencies among all of the regions in the grid via convolutional neural networks. Due to the local connectivity of the convolutional operation, a node in the higher-level feature map depends on multiple adjacent nodes in the lower-level feature map, and the adjacent feature nodes in the lower-level feature map are obtained by convolving multiple adjacent nodes in the lower-level feature map as shown in Fig. 8.

One convolution can only capture the spatial dependencies in adjacent regions, whereas multi-layer convolution can capture spatial dependencies over longer distances [36]. Thus, a multi-layer convolutional neural network needs to be designed to capture the spatial dependencies among all routes in the grid. First of all, the convolutional neural network is constructed separately to capture the dependencies of the data in different time dimensions based on the time segments of the borrowing and returning matrix extracted in the first step for two time periods. The multilayer convolution in this network structure is able to capture the interdependencies of all regions in the spatial dimension after the spatial reconstruction has been performed.

Denote transient correlation sequence by $(X_{\mathrm{Sti}})^{(0)} = X_{\mathrm{Sti}} \in \mathbb{R}^{l_{\mathrm{sti}} \times N \times N}$, the conversion formula is as follows:

$$(X_{\mathrm{Sti}})^{(1)} = f(W_{\mathrm{sti}}^{(1)} * (X_{\mathrm{Sti}})^{(0)} + b_{\mathrm{sti}}{}^{(1)}) \tag{14}$$

where $*$ denotes convolution. For ensuring that the output after the convolution operation matches the size of the input tensor, it is necessary to fill in the zero values around the input tensor. $W_{\mathrm{sti}}^{(1)}$ and $b_{\mathrm{sti}}^{(1)}$ are the learning parameters for the first layer of convolution, and $f(\bullet)$ denotes the activation function. Here, ReLU [37] is used as the activation function (i.e., $f(x) = \max(x, 0)$).

Step 4: Incorporating residual networks. As an easier-to-optimize neural network, the residual network can improve accuracy by increasing the depth of the network [38]. Its internal residual block uses jump connections to alleviate the gradient disappearance problem associated with increasing depth in deep neural networks, so in this paper's model, we use a residual learning strategy to capture spatial dependencies at longer distances. In our proposed model, L residual units are added consecutively following convolution operation Conv1, and formally, a residual unit can be defined as:

$$\begin{aligned} (X_{\mathrm{Sti}})^{(l+1)} &= (X_{\mathrm{Sti}})^{(l)}\\ &+F((X_{\mathrm{Sti}})^{(l)}; (\theta_{\mathrm{Sti}})^{(l)}, l = 1, 2, \ldots, L \end{aligned} \tag{15}$$

where $(X_{\mathrm{Sti}})^{(l)}$ and $(X_{\mathrm{Sti}})^{(l+1)}$ denote the input and output of the l-th residual unit, respectively; $F(\bullet)$ denotes residual mapping; $(\theta_{\mathrm{Sti}})^{(l)}$ denotes the learning parameter of the l-th residual unit.

After the first residual unit, convolution operation Conv2 is designed to be added, and the output of the short-term time-influenced part $(X_{\mathrm{Sti}})^{(l+2)}$ is obtained following the entire residual network calculation. The long-term time impact component uses the same network construction as the short-term time impact component, and the corresponding output is $(X_{\mathrm{Lti}})^{(l+2)}$.

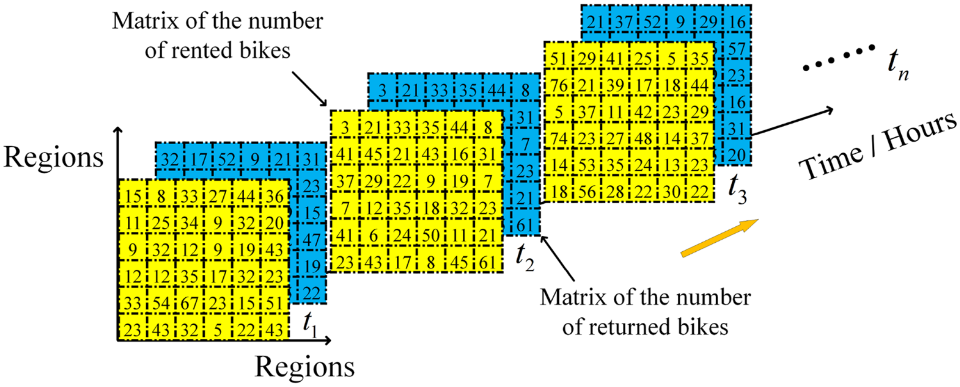

Matrix of the number of rented bikes
Regions
Regions
t_1
t_2
t_3
t_n
Time / Hours
Matrix of the number of returned bikes

Fig. 6 Four-dimensional array of rented and returned bikes

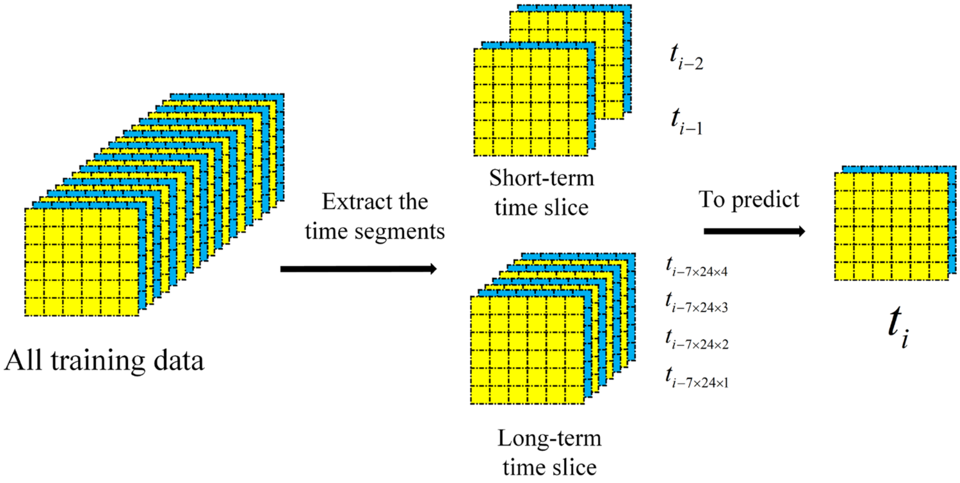


Fig. 7 Extracting time segments to form training data

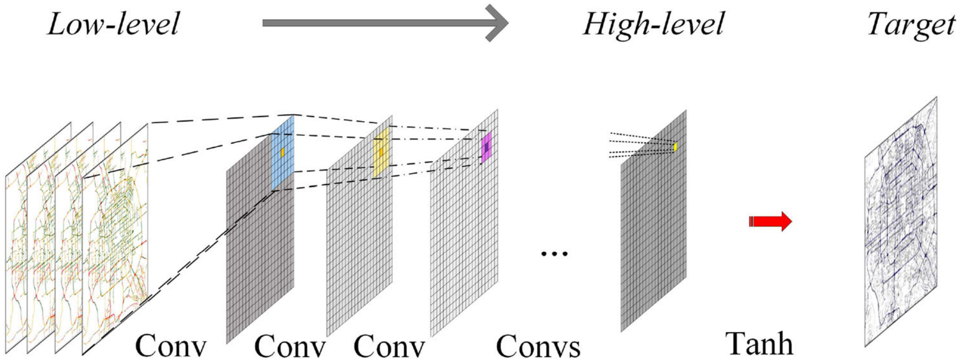
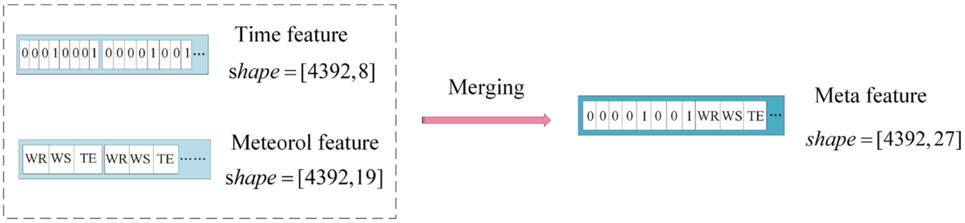


Fig. 8 Capturing spatial dependencies through convolutional networks

Step 5: Adding external factor components. Combined with the practical application context of the BSS, the main external factors considered in this paper are the weekday attribute of the corresponding date and the weather state of the corresponding moment. We model the influence of external factors on the BSS through a fully connected network [39]. The relevant experimental data are first obtained from the outside, and then, the corresponding features are extracted manually. For the weekday attribute, we use the time feature vector. The time feature of each moment consists of eight dimensions; the first seven dimensions are in one-hot form, and the last dimension indicates whether the day is a weekday. Taking Fig. 8 as an example, the meaning of the time feature is that the time slice corresponds to the date of Thursday and is a weekday. For weather conditions, we use the meteorol feature vector to represent. The meteorol feature of each moment consists of 19 dimensions. The first 17 are also in one-hot form, indicating one of the weather types, and the last two dimensions indicate wind speed and temperature, respectively. Finally, the two vectors are merged into a vector of 27 dimensions, represented as a meta-feature. The prediction moment is assumed to be t, and the external factors are represented by feature vector X_{ext}.

Fig. 9 Encoding external factor



Here, a two-layer, fully connected neural network is designed as an external factor network in this paper. The first layer can be regarded as an embedding layer, whose main role is to consider these external features together and to map them from high to low dimensions. In general, the number of neurons in this layer should not be too high, and suitable parameters should be selected according to the actual situation in the experiment. The role of the second layer is to map the low-dimensional features obtained from the first layer to the high-dimensional tensor, whose size needs to be consistent for the subsequent fusion. The output obtained after these two layers of the fully connected neural network is defined as X_{Ext}.

Step 6: Fuse the output of the residual network component with the external factor component. Based on the results obtained from the above components, this paper uses a parameter matrix fusion mechanism [18] to simultaneously fuse the outputs of each of the three components: the short-term time impact component, the long-term time impact component, and the external factor component. Through the construction of a coefficient matrix, different weights are assigned to the output results of each component and are fused to obtain a fused output result X_{Res}. The fusion equation is as follows:

$$X_{\mathrm{Res}} = W_{\mathrm{Sti}} \circ (X_{\mathrm{Sti}})^{(l+2)} + W_{\mathrm{Lti}} \circ (X_{\mathrm{Lti}})^{(l+2)} \tag{16}$$

where $\circ$ denotes the Hadamard product; $(X_{\mathrm{Sti}})^{(l+2)}$ and $(X_{\mathrm{Lti}})^{(l+2)}$ denote the output results of the short-term time impact component and the long-term time impact component, respectively; and W_{Sti} and W_{Lti} are the learning parameters denoting the importance of each of the two components to the prediction. Finally, the output results obtained from the calculation of Eq. (16) and the output results X_{Ext} obtained from the external factor part are aggregated and mapped from $[-1, 1]$ using the tanh function to obtain the final prediction results. The specific calculation formula is as follows:

$$X_{\mathrm{Net}} = \tanh(X_{\mathrm{Res}} + X_{\mathrm{Ext}}) \tag{17}$$

Our model trains the model by minimizing the mean square error between the predicted value matrix X_{Net} and the true value matrix $\hat{X}_{\mathrm{Net}}$ via the following equation:

$$L(\theta) = \left\| X_{\mathrm{Net}} - \hat{X}_{\mathrm{Net}} \right\|_2^2 \tag{18}$$

where θ denotes all learning parameters in the model.

Evaluation

Evaluation settings

Experiment environment

The experiments is performed by a 64-bit Ubuntu 18.04 computer with an Intel 3.40 GHz and an NVIDIA GTX 2080Ti GPU. We conduct RST-Net with Python 3.6, Keras 2.1.4 and Tensorflow 1.3.1.

Data sets

We have conducted an in-depth investigation of the dataset before we determined the dataset. We found that the usage data of station-based system from different shared-bike companies in various countries around the world are basically the same, mainly including the station IDs, borrowed and returned times, and the usage time of users. Considering the public nature of the dataset and the scale of the accessible data, we chose the public dataset of the New York shared bicycle system. In this section, we use shared-bike usage data and meteorological data from April 1 to September 30, 2014, in New York City as the dataset for our experimental analysis. The details of the dataset are shown in Table 1. The shared-bicycle usage data consist of 5,359,995 records. Each piece of data is composed of the following contents: riding time, borrowing station identification (ID), returning station ID, borrowing time and returning time. They are transferred into a trip set as $\{(S_{i,\mathrm{o}}, S_{i,\mathrm{d}}, t_{i,\mathrm{o}}, t_{i,\mathrm{d}})\}_{i=1}^{5359995}$. All shared-bicycle usage data from the New York BSS from July 2013 to now can be downloaded from the official websit.[1] The meteorology data are composed of the following contents: the weather conditions, temperature and wind speed at the corresponding time. Like the shared-bicycle usage data, all of them are

[1] http://www.citibikenyc.com/system-data.

Table 1 Statistics of data sets

Data source		New York City	
Time span		Apr 1, 2014–Sep 30, 2014	
Bike data	Station	344	
	Bike	6800+	
	Records	5,359,995	
Meteorology data	Weather (h)	Snowy	2
		Rainy	231
		Foggy	303
		Sunny	3856
	Temperature (°C)		[0,33]
	Wind speed (mph)		[0,18]

transferred into a meteorology set $\{w_i, p_i, v_i\}_{i=1}^{4392}$. We use the data from April 1 to September 20 to train the model, and we use the data from September 21 to 30 to test the model.

Compared baselines and metrics

Compared baselines

We use the following 11 baselines to compare with our model:

- Historical average (HA) [40]: we predict the number of bikes borrowed and returned in the future according to the average value of the historical number of bikes borrowed and returned in the set time period. For example, if we want to predict the usage of shared bicycles from 12:00 to 12:30 on Friday, the data we use should be the average of the bike-sharing usage data for the same time period every Friday in the past.
- Autoregressive integrated moving average (ARIMA) model [41]: the ARIMA model is a differential autoregressive moving average model that predicts the future (assuming that the future will repeat the historical trend) by finding the autocorrelation between historical data.
- Seasonal autoregressive integrated moving average (SARIMA) model: an extension of the ARIMA model.
- Vector autoregression (VAR) model [42]: the VAR model estimates the dynamic relationship of all endogenous variables by regressing the lagged values of endogenous variables on all endogenous variables of the model in each equation of the model through multiple equation linkages. It is often considered to be an advanced spatio-temporal model due to its ability to capture the pairwise relationships between all flows.
- Spatio–temporal attentive neural network (ST-ANN) [43]: a predictive model using artificial neural networks to model temporal and spatial features.
- DeepST [24]: Based on the differences in the length of the time series considered and the types of external factors, the investigators have designed four variants: DeepST-C, DeepST CP, DeepST-CPT and DeepST-CPTM.
- Recurrent neural network (RNN) [44]: a neural network for modeling sequential data, which can be trained on time series data of an arbitrary length and is widely used to capture a temporal correlation.
- Long-short-term-memory network (LSTM) [45]: LSTM is a special type of RNN, mainly created to solve the gradient disappearance and gradient explosion problems during the training of long sequences. Compared with a normal RNN, LSTM can have better performance with longer sequences.
- Gated-recurrent-unit (GRU) network [46]: the GRU is also a type of recurrent neural network. Like LSTM, it is also proposed to solve problems such as long-term memory and the gradient in backpropagation. Compared with LSTM, training with the GRU is easier, and this can improve the training efficiency to a great extent.
- Deep spatio-temporal residual network (ST-ResNet) [18]: a neural network that uses a residual neural network framework to model the temporal proximity, periodicity and trend characteristics of traffic flows; it then incorporates the effects of external factors to forecast traffic flows at future times.
- STAR [47]: a single network is used to model a temporal correlation based on the ST-ResNet, thus reducing a large number of parameters and increasing the model's iteration speed.
- ST-3DNet [48]: Based on the ST-ResNet, the correlation of traffic data in spatial and temporal dimensions is captured by introducing three-dimensional convolution.

Evaluation metrics

We use the root mean square error (RMSE) and mean absolute error (MAE) as evaluation criteria for assessing each model's performance:

$$\text{RMSE} = \sqrt{\frac{1}{z}\sum_{i=1}^{z}(y_i - \hat{y}_i)^2} \tag{19}$$

$$\text{MAE} = \frac{1}{z}\sum_{i=1}^{z}\left|(y_i - \hat{y}_i)\right| \tag{20}$$

where y_i is the groundtruth, $\hat{y}_i$ is the corresponding predicted values, and z is the number of all groundtruth.

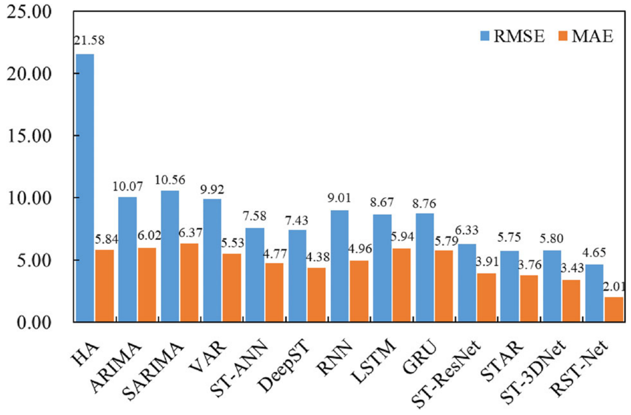


Fig. 10 Comparisons among different methods

Table 2 Prediction results of RMSE and MAE

Method	RMSE	MAE
HA	21.58	5.84
ARIMA	10.07	6.02
SARIMA	10.56	6.37
VAR	9.92	5.53
ST-ANN	7.58	4.77
DeepST	7.43	4.38
RNN	9.01	4.96
LSTM	8.67	5.94
GRU	8.76	5.79
ST-ResNet	6.33	3.91
STAR	5.75	3.76
ST-3DNet	5.80	3.43
RST-Net (ours)	**4.65**	**2.01**

Experimental results and analyses

Comparison of results with other models

In this section, we conduct comparative experiments on each model based on the shared-bike usage data from the New York City BSS from April 1, 2014, to September 30, 2014. The data from April 1 to September 20 are used as the training dataset, and the data from September 21 to 30 are used as the test dataset. Table 2 shows the RMSE and MAE of the prediction results corresponding to each method, and the best results have been highlighted in bold. Figure 10 graphically shows the results of the comparison.

The test results on the New York bike-sharing dataset show that the prediction results obtained using the RST-Net have a 54% lower RMSE than ARIMA; a 56% better RMSE than SARIMA; a 53, 39, 37, 48, 46, 47, 26, 19 and 19% better RMSE than the VAR, ST-ANN, Deepst, RNN, LSTM, GRU, ST-ResNet, STAR and ST-3DNet, respectively. In addition, the RST-NET has reduced the MAE by 41–68%. We think the usage of a spatio-temporal CNN makes the results of the DeepST model significantly better compared with other baselines. In addition, because only spatial information at short distances and temporal information at the nearest moment are considered, the ST-ANN and VAR are inferior to the DeepST, although they have exploited the relationship between spatio-temporal information and streams. In the temporal model, the RNN does not perform as well as the GRU and LSTM do, and we believe that the reason for this is that the RNN does not capture long-term temporal information as well as the GRU and LSTM do. The ST-ResNet uses spatio-temporal residual networks to capture the temporal dependence and spatial dependence of the BSS, and the obtained RMSE is reduced compared with other models that consider only temporal or spatial attributes by 27–34%. This proves that capturing both spatial and temporal attributes is critical for prediction. The results of the RMSE obtained via the STAR are the best except for in the case of the RST-Net. The reason for this is that the STAR incorporates some methods of video processing into the ST-ResNet, which improves the learning ability of the convolutional kernel in the channel dimension and makes the fusion of data more efficient. Taking all of the results together, the RMSE and MAE of the prediction results obtained using the RST-Net are reduced by 19–56% and 41–68%, respectively, compared with other models. This shows that the factors we consider here play a very important role in the prediction of shared-bike usage. This also demonstrates the necessity of using the RCS algorithm for BSS prediction. In addition, it shows that the RST-Net has good generalization performance compared with other shared-bicycle prediction models.

Analysis of the number of regions and iterations

In this section, we explore the effect of the number of regions determined for the RCS algorithm as well as the number of iterations determined for the iterative reconstruction on the final results. Figure 11 graphically shows the results.

First, we consider the number of regions m. To investigate the effect of the number of regions on the performance of the prediction model, we test the prediction results on the New York City bike-sharing dataset with a different m. It is important to note here that the number of regions should not be too large or too small. On the contrary, if the number of zones is too large, a high probability exists that few sites will be clustered into a single zone, resulting in a situation similar to that predicted for a single site, where the use of bikes in these zones is likely to be irregular. The situation is likely to be without any pattern. Combining the experience from previous studies [7] and our experimental results, we set the value of m in the range of $[0.05 * X, 0.1 * X]$, where X represents the number of stations. This constrains the intra-

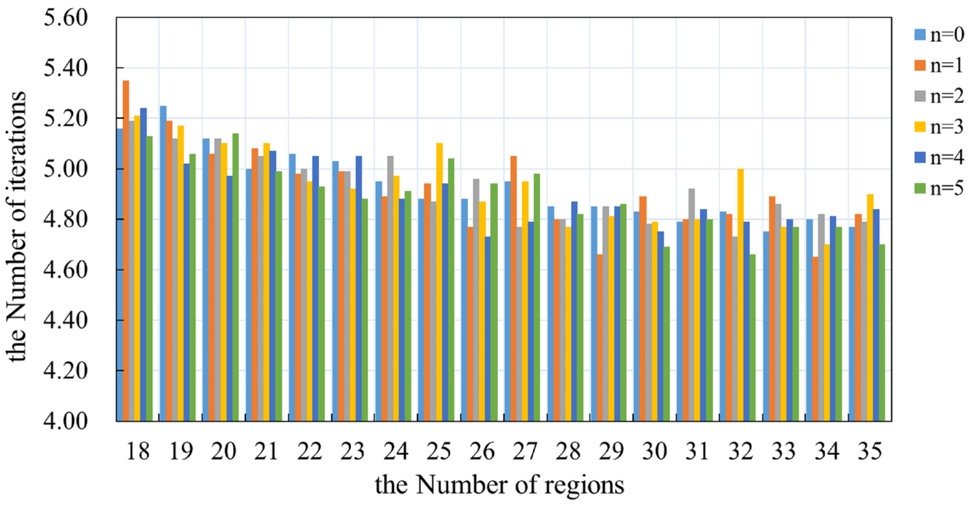


Fig. 11 Comparisons among different methods

Table 3 Different number of regions with different number of iterations

m	$n = 0$		$n = 1$		$n = 2$		$n = 3$		$n = 4$		$n = 5$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
18	5.16	4.93	5.35	4.92	5.19	4.92	5.21	**4.89**	5.24	4.93	**5.13**	4.94
19	5.25	4.52	5.19	4.54	5.12	**4.49**	5.17	4.50	**5.02**	4.53	5.06	4.50
20	5.12	**4.15**	5.06	4.17	5.12	4.16	5.10	**4.15**	**4.97**	4.17	5.14	4.17
21	5.00	**3.85**	5.08	3.88	5.05	3.87	5.10	3.89	5.07	**3.85**	**4.99**	**3.85**
22	5.06	3.63	4.98	3.60	5.00	3.61	4.95	3.60	5.05	3.62	**4.93**	**3.58**
23	5.03	3.41	4.99	**3.37**	4.99	3.38	4.92	**3.37**	5.05	3.39	**4.88**	3.40
24	4.95	3.19	4.89	3.21	5.05	3.20	4.97	3.21	**4.88**	3.18	4.91	**3.17**
25	4.88	3.03	4.94	3.05	**4.87**	3.03	5.10	**3.01**	4.94	3.02	5.04	3.03
26	4.88	2.88	4.77	2.86	4.96	2.88	4.87	**2.85**	**4.73**	2.88	4.94	**2.85**
27	4.95	2.75	5.05	2.74	**4.77**	2.75	4.95	**2.73**	4.79	**2.73**	4.98	**2.73**
28	4.85	2.63	4.80	**2.60**	4.80	2.62	**4.77**	2.62	4.87	**2.60**	4.82	2.61
29	4.85	2.51	**4.66**	2.50	4.85	**2.49**	4.81	**2.49**	4.85	**2.49**	4.86	2.51
30	4.83	2.42	4.89	**2.39**	4.78	2.38	4.79	2.40	4.75	2.40	**4.69**	**2.39**
31	4.80	2.31	4.80	**2.30**	4.92	**2.30**	**4.79**	2.32	4.84	**2.30**	4.80	2.32
32	4.83	2.25	4.82	**2.22**	4.73	2.24	5.00	2.21	4.79	2.23	**4.66**	2.23
33	4.80	2.17	4.89	2.16	4.86	2.16	4.77	**2.15**	**4.75**	**2.15**	4.77	**2.15**
34	4.80	2.09	**4.65**	2.08	4.82	2.10	4.70	2.08	4.81	**2.07**	4.77	2.08
35	4.77	2.02	4.82	**2.01**	4.79	2.03	4.90	2.03	4.84	2.03	**4.70**	2.04

cluster distance, ensuring that each cluster is neither too large nor too small. Table 3 shows the RMSEs and MAEs with different numbers of regions and different iterations. The best results have been highlighted in bold. It is clear from Table 3 that the result decreases as the number of regions increases.

Next is the number of iterations n. After determining the number of regions, we perform five iterations on them, respectively, to explore the effect of different iterations on the final prediction results. It is not difficult to see that for the same region, the prediction results obtained by clustering after adding the migration trend information between shared-bicycle sites are better than the prediction results are after clustering based only on the geographical location information of the shared-bicycle sites. This proves the rationality of the introduction of information on the migration trends of shared bicycles for RCS. In addition, we have found that the accuracy of the prediction results is not linearly correlated with the increase in the number of iterations. Generally, considering the time complexity of the algorithm, four or five iterations are sufficient.

Conclusion

In this paper, we propose a deep spatio-temporal residual network model based on the RCS algorithm to predict the bike-sharing usage of the BSS in future time. First, we use the RCS algorithm to classify the bike-sharing sites in the city. Unlike the previous regional classification based on the geographic locations of bike-sharing sites, the RCS introduces a "site→region" migration trend matrix to capture the bike migration trends among bike-sharing sites. This significantly improves the quality of the regional classification results. Based on this, this paper uses a deep spatio-temporal residual network to model the key factors affecting the usage of shared bicycles. Experiments on the New York BSS show that the results obtained using our model are significantly better compared with previous models, and the prediction accuracy has been greatly improved.

Funding The work is partially supported by the National Natural Science Foundation of China (Nos. U1836216, 61772322), and the major fundamental research project of Shandong, China(No. ZR2020MF042).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. DeMaio P (2009) Bike-sharing: history, impacts, models of provision, and future. J Public Transp 12(4):3
2. Liu Z, Jia X, Cheng W (2012) Solving the last mile problem: ensure the success of public bicycle system in Beijing. Procedia Soc Behav Sci 43:73–78
3. Midgley P (2009) The role of smart bike-sharing systems in urban mobility. Journeys 2(1):23–31
4. Patel Samir J, Patel Chetan R (2019) An infrastructure review of public bicycle sharing system (PBSS): global and Indian scenario. Innovative research in transportation infrastructure. Springer, Singapore, pp 111–120
5. Chen M et al (2020) A comparison of users' characteristics between station-based bikesharing system and free-floating bikesharing system: case study in Hangzhou, China. Transportation 47(2):689–704
6. Li Y, Zheng Yu (2019) Citywide bike usage prediction in a bike-sharing system. IEEE Trans Knowl Data Eng 32(6):1079–1091
7. Jia W et al (2019) Hierarchical prediction based on two-level Gaussian mixture model clustering for bike-sharing system. Knowl-Based Syst 178:84–97
8. Lin J-R, Yang T-H (2011) Strategic design of public bicycle sharing systems with service level constraints. Transp Res Part E Logist Transp Rev 47(2):284–294
9. Li Y et al (2015) Traffic prediction in a bike-sharing system. In: Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems
10. Pan Z et al (2020) Spatio-Temporal meta learning for urban traffic prediction. IEEE Trans Knowl Data Eng 34(3):1462–1476
11. Jia W, Tan Y, Li J (2018) Hierarchical prediction based on two-level affinity propagation clustering for bike-sharing system. IEEE Access 6:45875–45885
12. Wang K et al (2008) Adaptive affinity propagation clustering. arXiv preprint arXiv:0805.1096
13. Yang Z et al (2019) Mobility modeling and data-driven closed-loop prediction in bike-sharing systems. IEEE Trans Intell Transp Syst 1–12
14. Liu J et al (2016) Station site optimization in bike sharing systems. In: 2015 IEEE international conference on data mining (ICDM). IEEE
15. Chen L et al (2016) Dynamic cluster-based over-demand prediction in bike sharing systems. In: The 2016 ACM international joint conference ACM
16. Yang Z, Yan H (2018) Context aware flow prediction of bike sharing systems. In: 2018 IEEE international conference on big data (Big Data). IEEE
17. Pan Y et al (2019) Predicting bike sharing demand using recurrent neural networks. Procedia Comput Sci 147:562–566
18. Zhang J et al (2018) Predicting citywide crowd flows using deep spatio-temporal residual networks. Artif Intell 259:147–166
19. Hoang MX, Zheng Y, Singh AK (2016) FCCF: forecasting citywide crowd flows based on big data. In: Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems
20. Zhang X et al (2020) Traffic flow forecasting with spatial-temporal graph diffusion network
21. Yi X et al (2019) Citytraffic: modeling citywide traffic via neural memorization and generalization approach. In: Proceedings of the 28th ACM international conference on information and knowledge management
22. Huang L et al (2021) Regional logistics demand forecasting: a BP neural network approach. Complex Intell Syst 1–16
23. Qiu Y et al (2021) System dynamics mechanism of cross-regional collaborative dispatch of emergency supplies based on multi-agent game. Complex Intell Syst 1–12
24. Vitória A, Dias MS, Bacao F (2021) Machine learning approaches to bike-sharing systems: a systematic literature review. Int J Geo-Inf 10(2):62
25. McLachlan GJ, Basford KE (1988) Mixture models: inference and applications to clustering, vol 38. M. Dekker, New York
26. Bilmes JA (1998) A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Int Comput Sci Inst 4(510):126
27. Biernacki C, Celeux G, Govaert G (2000) Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans Pattern Anal Mach Intell 22(7):719–725
28. Peters G et al (2013) Soft clustering-fuzzy and rough approaches and their extensions and derivatives. Int J Approx Reason 54(2):307–322
29. Zheng Y et al (2014) Urban computing: concepts, methodologies, and applications. ACM Trans Intell Syst Technol (TIST) 5(3):1–55

30. Zhang J et al (2016) DNN-based prediction model for spatio-temporal data. In: Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems
31. Bargar A et al (2014) Interactive visual analytics for multi-city bike-share data analysis. In: The 3rd international workshop on urban computing (UrbComp 2014), New York, USA, vol 45
32. O'Mahony E, David S (2015) Data analysis and optimization for (citi) bike sharing. In: Proceedings of the AAAI conference on artificial intelligence, vol 29, no 1
33. Kaltenbrunner A et al (2010) Urban cycles and mobility patterns: exploring and predicting trends in a bicycle-based public transport system. Pervas Mob Comput 6(4):455–466
34. Böttcher A, Wenzel D (2008) The Frobenius norm and the commutator. Linear Algebra Appl 429(8–9):1864–1885
35. Sun J et al (2020) Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. IEEE Trans Knowl Data Eng
36. Geng X et al (2019) Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, no 01
37. Agarap AF (2018) Deep learning using rectified linear units (RELU). arXiv preprint arXiv:1803.08375
38. He K et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition
39. Sainath TN et al (2015) Convolutional, long short-term memory, fully connected deep neural networks. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
40. Yao H et al (2018) Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol 32, no 1
41. Contreras J et al (2003) ARIMA models to predict next-day electricity prices. IEEE Trans Power Syst 18(3):1014–1020
42. Min W, Wynter L (2011) Real-time road traffic prediction with spatio-temporal correlations. Transp Res Part C Emerg Technol 19(4):606–616
43. He Z, Chow C-Y, Zhang J-D (2018) STANN: a spatio-temporal attentive neural network for traffic prediction. IEEE Access 7:4795–4806
44. Goodfellow I et al (2016) Deep learning, vol 1, no 2. MIT Press, Cambridge
45. Cho K et al (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
46. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
47. Wang H, Su H (2019) STAR: a concise deep learning framework for citywide human mobility prediction. In: 2019 20th IEEE international conference on mobile data management (MDM). IEEE
48. Guo S et al (2019) Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting. IEEE Trans Intell Transp Syst 99:1–14