Temporal Attention-Based Graph Convolution Network for Taxi Demand Prediction in Functional Areas

Yue Wang, Jianbo Li(✉), Aite Zhao, Zhiqiang Lv(✉), and Guangquan Lu

College of Computer Science and Technology, Qingdao University, Qingdao 266071, China
lijianbo@qdu.edu.cn

Abstract. Shared travel is increasingly becoming an indispensable way of urban transportation. Accurately calculating the demand for taxis in various regions of the city has become a huge problem. In this paper, we divide the city into multiple lattices of different sizes and propose a graph convolution network based on the temporal attention mechanism for taxi demand prediction in each functional area of the city. The model includes graph convolution network (GCN), temporal convolution network (TCN), and the attention mechanism, which are respectively used to capture the spatial correlation of roads, time dependence, and highlight the characteristics of the time-series data. Extensive experiments on three datasets validate the effectiveness of the proposed method, compared against several state-of-the-art methods. Despite there are amount differences among the three datasets in our experiment, our model still has a high prediction accuracy. Our model code is available at https://github.com/qdu318/TAGCN.

Keywords: Taxi demand prediction · Attention mechanism · Graph convolution network · Temporal convolution network

1 Introduction

In recent years, although domestic private car ownership has increased substantially, due to the rapid development of the sharing economy [1], taxis are playing an increasingly significant role in urban transportation. In 2020, the taxi empty rate in Beijing has risen to nearly 40%, which undoubtedly caused a serious waste of resources and traffic congestion in the city. Therefore, a key aspect of solving the above problem is to accurately predict the demand for taxis in each region.

Traffic forecasting has received extensive attention and research in the past few decades [20, 21]. Traffic prediction models can be divided into non-parametric and parametric models. Among them, the parameters are from the data distribution. Usually, parametric models include the time series model [2], autoregressive integrated moving average model (ARIMA) [3], linear regression model [4], and Kalman filtering model [5]. The parametric models have relatively simple structures and they specify the learning method in the form of a certain function, which strictly depends on the stationary hypothesis and cannot reflect the uncertainty of the traffic state and non-linearity. The

Z. Liu et al. (Eds.): WASA 2021, LNCS 12937, pp. 203–214, 2021.
https://doi.org/10.1007/978-3-030-85928-2_16

data-driven non-parametric models can effectively solve these problems, including traditional machine learning and deep learning approaches. Without prior knowledge, they can still fit multiple functional forms of enough historical data. Traditional machine learning models include k-nearest neighbors [6], decision trees [7] (such as CART and C4.5), naive Bayes [8], support vector machines [9], and neural networks [10]. Since traditional machine learning methods usually require complex feature engineering and are not suitable for processing a large number of datasets, the deep learning algorithm has emerged. Common deep learning models include Auto-encoder [11], generative adversarial networks [12], convolutional neural network (CNN) [13], and recurrent neural network [14]. Although the Recurrent Neural Network (RNN) is designed for dealing with time-series problems, and CNN has the ability to extract spatial correlation, they are only suitable for processing structured data. Especially, CNN divides the traffic network into a two-dimensional regular lattice, which destroys the original structure of the traffic network.

Therefore, we design a temporal attention-based graph convolutional network (TAGCN) for predicting the taxi's demand in the functional area. The model can directly process traffic data on graphics, effectively capture complex spatial-temporal correlations, and more accurately predict local peaks and the values of the start and end points of the data. The main contributions of this paper are summarized as follows:

- We propose a new temporal graph convolutional network model based on the attention mechanism, which embeds Attention to highlight the characteristics of traffic data. The spatial-temporal convolution model includes graph convolutional network (GCN) and temporal convolutional network (TCN) to realize the spatial-temporal correlation of the traffic data.
- After changing the previous lattice division ways, the city is divided into multiple lattices of different sizes according to the function of area, and the spatial location is converted into a graphic structure.
- Three real-world datasets are utilized in the experiment. Compared with 7 state-of-the-art baselines, TAGCN has achieved the best prediction results.

2 Related Work

A large number of methods based on deep learning networks are widely used in traffic prediction problems without artificially synthesizing complex features and cross-domain knowledge sharing. For example, Contextualized Spatial-Temporal Network for Taxi Origin-Destination Demand Prediction (CSTN) [15] uses CNN to learn the dependence of local space and ConvLSTM to analyze the change of taxi demand over time. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction (DMVST-Net) [16] uses CNN to capture the spatial proximity and Long Short-Term Memory (LSTM) to model the time series. The short-term time dependence in Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction (STDN) [17] is obtained by LSTM, the time periodicity is obtained by Attention for the previous days, and the spatial relationship between adjacent regions is captured by CNN.

Although LSTM performs well on several sequence problems (such as speech/text recognition [18] and machine translation [19]), since the network can only handle one time step at a time, the next step must wait for the previous step to complete the operation. It means that LSTM cannot solve large-scale parallel computing problems. The data processed by LSTM and CNN belongs to the Euclidean space and has a regular structure. However, in real life, there are many irregular data structures, such as social networks, chemical molecular structures, knowledge maps, and other topological structures, which makes the prediction effect of RNN and CNN greatly ineffective. At the same time, the above work divides the region in the form of average size, which cannot well distinguish each functional area.

The non-Euclidean spatial data processing methods include graph convolutional network (GCN) [22], graph neural network (GNN) [23], DeepWalk (Online learning) [24], node2vec [25], etc. The typical model GCN has the same role as CNN as a feature extractor. However, since the spatial relationship of the traffic map is a topological structure, which belongs to non-Euclidean data, GCN can better address the map data and extract the spatial features of the topological map than CNN. The extracted features can be used in graph classification, node classification, link prediction, and graph embedding, which fully proves the ability of GCN to process highly nonlinear data in non-Euclidean space.

Therefore, we select GCN as the internal component to extract the correlation of the input spatial data. For time-series data, the temporal convolutional network (TCN) [28] is utilized to obtain the temporal dependence, meanwhile, the characteristics of the traffic demand data are highlighted by the temporal attention mechanism [34]. In terms of functional areas, we divide the city into multiple lattices of different sizes. This work aims to predict the taxi's demand of each area at the $(t+1)^{th}$ time interval, given the historical data of the previous t time intervals.

3 Algorithm Design

3.1 Data Design

We extract information about passengers getting on a taxi from the three original datasets, draw a scatter plot based on the extracted data, determine the area with the densest data, and select all the data in that area. To minimize the difference of the data, the final area of the three datasets is about 22.2 km × 9.16 km.

According to the functionality of the region, we divided the cities in the three datasets into four types: school district, recreation area, residential area, and business district. Furthermore, the city is divided into lattices according to the distribution of functional areas, and finally, 60 lattices of different sizes are divided and all lattices are numbered. To determine the scope of each lattice, we extract the boundaries of all lattices one by one and judge the lattice where the starting position of the taxi is as well as the center of each lattice. The calculation of the latitude and longitude of the center point are respectively shown in Eq. (1) and Eq. (2),

$$\varphi = \frac{|\varphi_2 - \varphi_1|}{2} \tag{1}$$

$$\lambda = \frac{|\lambda_2 - \lambda_1|}{2} \tag{2}$$

where φ and λ represent the latitude and longitude of the center point, respectively; φ_2 and φ_1 represent the latitude of the two vertices on the left and right respectively; λ_2 and λ_1 represent the longitude of the two vertices above and below, respectively.

The data should be preprocessed first that can be input into the model, i.e., the A matrix and the V matrix are calculated. The A matrix is an adjacency matrix based on the distance between lattices, used to store the mutual distance between 60 lattices. The V matrix represents the taxi's demand in each lattice. In the calculation of matrix A, we use the distance between the center points of the lattice to represent the distance between each area. The distance between each center point is calculated by Eq. (3).

$$d = 2\pi r arcsin(\sqrt{sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + cos(\varphi_1)cos(\varphi_2)sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}) \tag{3}$$

where d is the distance between two center points, and r is the radius of the earth.

After the data processing is completed, we use the *Z-score* data standardization method to standardize the data. The *Z-score* method is mainly used to process data with too messy distribution and too many singularities. The standard deviation of the data after *Z-score* processing is 1, and the mean is 0. The calculation of the *Z-score* is shown in Eq. (4).

$$z = \frac{x - \mu}{\sigma} \tag{4}$$

where x is the original data, μ and σ are respectively the average and the standard deviation of the overall sample space.

3.2 Model Design

This paper solves the spatial-temporal problem of taxi demand forecast. We use two temporal convolution layers to obtain the long-term dependence of time and a spatial convolution layer to extract the spatial correlation. When the data is input to the first temporal convolutional layer, it is also input to the attention layer to highlight its characteristics, and the output of the attention layer is fed into the first temporal convolutional layer for extracting the low-level feature. The adjacency matrix is mapped from the distance between the lattices. After normalization, the obtained data can be comparable and maintain the relative relationship between the data. The results of the first temporal convolution layer and the adjacency matrix are input to the spatial convolution layer to enhance the spatial relative relationship of features. Enter these results into the second temporal convolution layer to obtain the temporal dependence and high-level feature of the fused data. TCN is composed of dilated convolution, causal convolution, and residual structure. To solve the gradient explosion or disappearance caused by complete convolution networks, we add *BatchNorm2D* to the data post-processing layer. Finally, to map the learned distributed feature representation to the sample label space, we add a fully connected layer, which retains the complexity of the model. The model structure is shown in Fig. 1.

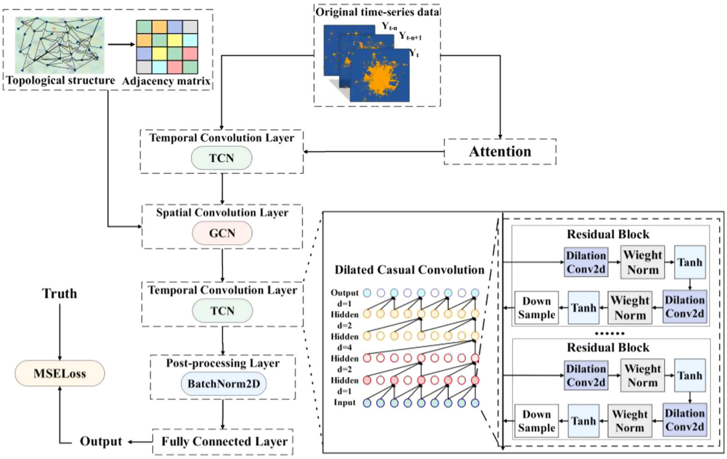


Fig. 1. The model structure. TAGCN is composed of temporal attention mechanism, temporal convolutional layer, spatial convolutional layer, and post-processing layer.

Spatial Convolution Layer. We use GCN to extract spatial correlation based on the distance between lattices. The core of GCN is based on the spectral decomposition of the Laplacian matrix, i.e., the eigenvector of the Laplacian matrix corresponding to the graph is obtained from the eigenfunction e^{-iwt} of the Laplacian operator. Laplacian matrix can make the transfer intensity of the data features in GCN proportional to their state differences. To increase the influence of the original node in the calculation process, we use the modified version of the Laplacian matrix, and the formula is defined as shown in Eq. (5),

$$L = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \tag{5}$$

where $\tilde{A} = A + I$, I is the identity matrix, $\tilde{D}$ is the degree matrix of $\tilde{A}$, and its formula is $\tilde{D}_{ii=} \sum j\tilde{A}_{ij}$.

However, because the graph convolution kernel is global, the amount of parameters is large, and the calculation process involves high computational complexity feature decomposition, the complexity of the graph convolution operation is extremely high. Using Chebyshev [35] polynomials to fit the convolution kernel can achieve the purpose of reducing computational complexity. The *k-th* order truncated expansion calculation of Chebyshev polynomial $T_k(x)$ is shown in Eq. (6),

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta_k' T_k(\widetilde{\Lambda}) \tag{6}$$

where $g_{\theta'}(\Lambda)$ is a function expressed by the eigenvalues of the Laplacian matrix, k represents the highest order of the polynomial, $\widetilde{\Lambda}$ is the scaled eigenvector matrix, $\widetilde{\Lambda} = \frac{2\Lambda}{\lambda_{max}} - I_n$, λ_{max} is the spectral radius of L, and the Chebyshev polynomial is recursively defined as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, $T_1(x) = x$, $T_0(x) = 1$.

Temporal Convolution Layer. We use Temporal Convolutional Network (TCN) to obtain the temporal dependence, which is composed of dilated convolution with the same input and output length, causal convolution, and residual structure. Causality means that in the output sequence, the elements at time t can only depend on the elements at time t and before in the input sequence. To ensure that the output tensor has the same length as the input tensor, zeros are padded to the left of the input tensor. For causal convolution, the modeling length of its time is limited by the size of the convolution kernel. To obtain long-term dependence, it is necessary to linearly stack many layers. Therefore, the researchers proposed dilated convolution. When the number of layers of the convolutional network is small, we can obtain a large receptive field after using dilated convolution. However, even if dilated convolution is used, the network structure may still be deep, which will cause the gradient to disappear. The addition of the residual structure can solve it. The reason is that there is a cross-layer connection structure in the residual structure, which transmits messages in a cross-layer manner. A residual block includes two layers of convolution and nonlinear mapping, and *WeightNorm* [27] is added to each layer to regularize the network. To make TCN not just an overly complex linear regression model, we add an activation function *Tanh* to the residual block to introduce nonlinearity.

Attention Mechanism. In the time dimension, there is a correlation between traffic status in different time periods, and the correlation is also different in various actual situation. The addition of the attention mechanism can highlight the correlation between input data and current output data, and obtain the temporal-dependent intensity of time i and time j to improve the accuracy of prediction. By dynamically calculating the current input and learning parameters, we can get the time attention matrix T, as shown in Eqs. (7) and (8). The activation function *sigmoid* is used to normalize the output data, and the range is compressed between [0, 1].

$$T = V_e((X * U_1)U_2(X * U_3) + b_e \tag{7}$$

$$T^{'}_{i,j} = \frac{e^{T_{i,j}}}{\sum_{i=1}^{N} e^{T_{i,j}}} \tag{8}$$

Post-processing Layer. Since the model is composed of complete convolutions, it is easy to cause gradient disappearance and gradient explosion. To solve these problems, we introduce *BatchNorm2D*. *BatchNorm2D* standardizes the output of each layer to make the mean and variance consistent, so as to improve the stability of model training and accelerate the speed of network convergence. We can generally understand it as *BatchNorm2D* pushing the output from the saturated zone to the unsaturated zone. The calculation of *BatchNorm2D* is shown in Eq. (9),

$$y = \frac{x - E(x)}{\sqrt{Var(x)} + \in} \times \gamma + \beta \tag{9}$$

where x is the input data, $E(x)$ and $\sqrt{Var(x)}$ are the mean and variance of x, $\in$ is a variable added to prevent zero from appearing in the denominator, and γ and β are learning parameters of 1 and 0, respectively. In the Adam optimizer, the step size is updated by the mean of the gradient and the square of the gradient. Different adaptive learning rates are calculated by the first-order and the second-order moment estimation of the gradient for different parameters. The role of the fully connected layer is to fuse the learned distributed feature representation together as an output value. Finally, the difference between the predicted value y_i and the actual value x_i is calculated by the mean square loss function MSELoss.

4 Experiment

4.1 Dataset Analysis

Three real-world datasets are used in our experiment, i.e., the taxi order dataset of Chengdu City in August 2014, Chengdu City in November 2016, and Haikou City from May to October 2017 after desensitization [26]. The above datasets all contain the time, longitude, and latitude of the trajectory. We select the start time of the order from 7:00 to 21:00 in all datasets and the latitude and longitude of the pick-up location. For Chengdu City, the longitude range is [103.95, 104.15] and the latitude range is [30.65, 30.75], which is shown in Fig. 2.

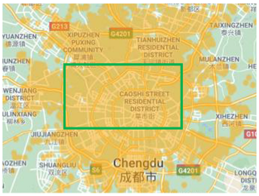


Fig. 2. Parts of Chengdu City, Sichuan Province, China.

4.2 Baselines

In order to specifically judge the performance of our model, we choose seven state-of-the-art models to compare with TAGCN. The baselines are introduced as follows:

- Long Short Term Memory Network (LSTM) [29]: The addition and removal of information can be achieved through the gated structure.
- Gated Recurrent Unit (GRU) [30]: It can realize forgetting and selective memory using one gate.
- Spatial-Temporal Dynamic Network (STDN) [17]: It utilizes CNN, LSTM and attention to obtain spatial-temporal features.
- Graph Convolution Networks (GCN) [22]: It is a generalization of CNN for learning non-grid data in the field of graphs.
- Temporal Graph Convolutional Network (T-GCN) [31]: It combines GCN and GRU to capture the road network topology and the temporal dependence.

- Spatio-Temporal Graph Convolutional Networks (STGCN) [32]: It consists of multiple spatio-temporal convolution modules to capture spatio-temporal correlation.
- Attention Based Spatial-Temporal Graph Convolutional Networks (ASTGCN) [33]: It uses three modules to process three different fragments and the data is processed by two layers of ST blocks.

4.3 Evaluation Index

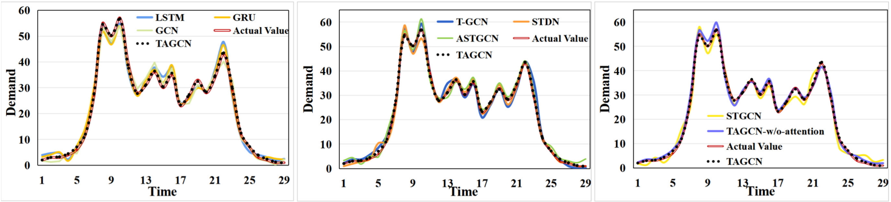
We use Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error(MAPE) to evaluate baselines and our model. RMSE is sensitive to the extra-large and extra-small error in the predicted value, so it can well reflect the precision of the model. MAPE is more robust than RMSE. MAE can reflect the real situation of the predicted value error. It can be seen from Table 1 that all the indicators of TAGCN are the smallest.

Table 1. MAE, RMSE and MAPE of each model.

Model	MAE	RMSE	MAPE
LSTM [29]	4.95	7.87	12.43
GRU [30]	4.76	7.61	12.11
GCN [22]	4.39	7.04	11.05
T-GCN [31]	4.21	6.26	9.24
ASTGCN [33]	4.04	5.53	8.01
STDN [17]	3.63	4.79	6.96
STGCN [32]	3.51	4.55	6.37
TAGCN-w/o-attention	3.25	4.01	5.13
TAGCN	**2.96**	**3.75**	**4.56**

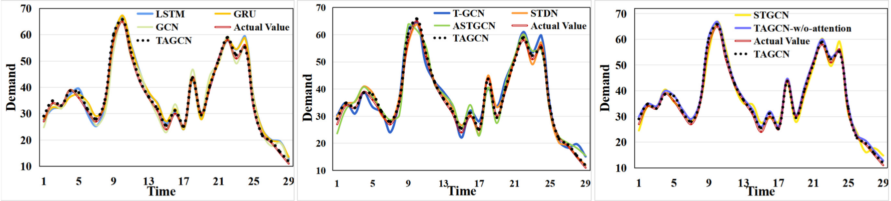
4.4 Spatial-Temporal Analysis

Figures 3–6 show the predictions of the taxi's demand in the recreation area, school district, residential area, and business district by eight models. The X-axis represents time, and one unit represents half an hour. The Y-axis represents the demand for taxis. We can see that the prediction results of TAGCN are more accurate than baselines. As can be seen from Fig. 3, 11:00–13:00 and 17:00–20:00 are the peaks for taxi's demand in the recreation area. The reason is that people prepare to leave the area after taking a rest and starting entertainment. It can be seen from Fig. 4 that 11:30–12:30 and 17:00–18:00 are the peaks for taxi's demand in the school district because students leave school during these periods. In Fig. 5, 7:00–8:00 and 12:00–13:00 are the peaks for taxi's demand in the residential area, which are the rush hours before the work time . It can be seen from



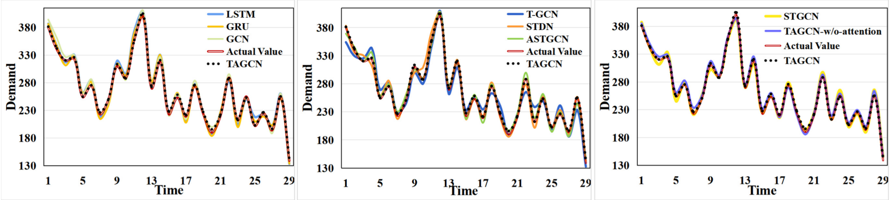
(a) (b) (c)

Fig. 3. Demand in recreation area.



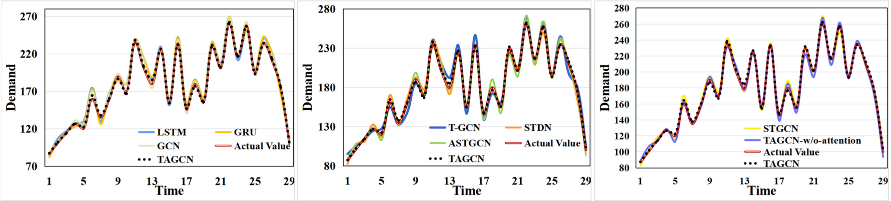
(a) (b) (c)

Fig. 4. Demand in school district.



(a) (b) (c)

Fig. 5. Demand in residential area.



(a) (b) (c)

Fig. 6. Demand in business district.

Fig. 6 that 11:30–12:00 and 18:00–20:00 are the peaks for taxi's demand in the business district because people knock off work.

The comparisons between TAGCN and LSTM, GRU, and GCN are shown in Figures (a). LSTM and GRU are mainly used to predict sequence information, and they cannot obtain spatial information. GCN can process irregular graph data to obtain spatial characteristics, however, it lacks the capture of temporal characteristics. The comparisons between TAGCN, T-GCN, STDN and ASTGCN are shown in Figures (b). The comparisons of TAGCN with and STGCN and TAGCN-w/o-attention are shown in Figures (c). T-GCN, STDN, ASTGCN, STGCN and TAGCN-w/o-attention can simultaneously obtain spatial and temporal dependence, but it is found that the prediction results of TAGCN for local peaks and edge values are better than baselines. The reason is that traditional neural networks are more difficult to capture long-term dependent information due to the limitation of the size of the convolution kernel, while TCN composed of dilated convolution and casual convolution can extract features across time steps and the temporal attention can highlight the features of the time-series data. Compared with the attention-based ASTGCN model, the reason for the advantage of our model is the TCN mentioned above, while ASTGCN uses standard convolution to extract the time features. Therefore, adding TCN and the temporal attention based on GCN makes the prediction of the model more stable.

5 Conclusion

We proposed a temporal attention-based graph convolution network model, which is used to predict passengers' demand for taxis in each functional area of the city. The model focuses on extracting temporal dependence, spatial information, and highlighting the characteristics of the time-series data. After comparing the indicators and predicted values with some state-of-the-art models, we concluded that TAGCN is superior to the baselines in predicting local peaks and edge values. The role of TAGCN is to assist in the scheduling of taxis to avoid the waste of road resources and passenger time. In the future, we will consider the destination of passengers, and introduce external factors to further improve the accuracy of our model.

Acknowledgments. This research was supported in part by Shandong Province colleges and universities youth innovation technology plan innovation team project under Grant No. 2020KJN011, Shandong Provincial Natural Science Foundation under Grant No. ZR2020MF060, Program for Innovative Postdoctoral Talents in Shandong Province under Grant No. 40618030001, National Natural Science Foundation of China under Grant No. 61802216, and Postdoctoral Science Foundation of China under Grant No.2018M642613.

References

1. Puschmann, T., Alt, R.: Sharing economy. Bus. Inf. Syst. Eng. **58**(1), 93–99 (2016)
2. Hurvich, C.M., Tsai, C.L.: Regression and time series model selection in small samples. Biometrika **76**(2), 297–307 (1989)
3. Saboia, J.L.M.: Autoregressive integrated moving average (ARIMA) models for birth forecasting. J. Am. Stat. Assoc. **72**(358), 264–270 (1977)

4. Maydeu-Olivares, A., Shi, D., Fairchild, A.J.: Estimating causal effects in linear regression models with observational data: the instrumental variables regression model. Psychol. Methods **25**(2), 243 (2020)
5. Patra, A.K.: Adaptive kalman filtering model predictive controller design for stabilizing and trajectory tracking of inverted pendulum. J. Inst. Eng. (India) Ser. B **101**(6), 677–688 (2020)
6. Saadatfar, H., Khosravi, S., Joloudari, J.H.: A new K-nearest neighbors classifier for big data based on efficient data pruning. Mathematics **8**(2), 286 (2020)
7. Yariyan, P., Janizadeh, S., Van Phong, T.: Improvement of best first decision trees using bagging and dagging ensembles for flood probability mapping. Water Resour. Manage. **34**(9), 3037–3053 (2020)
8. Jiang, L., Zhang, L., Yu, L.: Class-specific attribute weighted naive Bayes. Pattern Recogn. **88**, 321–330 (2019)
9. Huang, Y., Zhao, L.: Review on landslide susceptibility mapping using support vector machines. CATENA **165**, 520–529 (2018)
10. Van Gerven, M., Bohte, S.: Artificial neural networks as models of neural information processing. Front. Comput. Neurosci. **11**, 114 (2017)
11. He, Z., Shao, H., Wang, P.: Deep transfer multi-wavelet auto-encoder for intelligent fault diagnosis of gearbox with few target training samples. Knowl. Based Syst. **191**, 105313 (2020)
12. Goodfellow, I.J., et al.: Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural. Inf. Process. Syst. **25**, 1097–1105 (2012)
14. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)
15. Liu, L., Qiu, Z., Li, G., Wang, Q., Ouyang, W., Lin, L.: Contextualized spatial–temporal network for taxi origin-destination demand prediction. IEEE Trans. Intell. Transp. Syst. **20**(10), 3875–3887 (2019)
16. Yao, H., et al.: Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, (2018)
17. Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z.: Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5668–5675 (2019)
18. Van Hoai, D.P., Duong, H.-T., Hoang, V.T.: Text recognition for Vietnamese identity card based on deep features network. Int. J. Doc. Anal. Recogn. (IJDAR) **24**(1–2), 123–131 (2021). https://doi.org/10.1007/s10032-021-00363-7
19. Liu, Y., Gu, J., Goyal, N.: Multilingual denoising pre-training for neural machine translation. Trans. Assoc. Comput. Linguist. **8**, 726–742 (2020)
20. Lv, Z., Li, J., Dong, C., Zhao, W.: A deep spatial-temporal network for vehicle trajectory prediction. In: Yu, D., Dressler, F., Yu, J. (eds.) WASA 2020. LNCS, vol. 12384, pp. 359–369. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59016-1_30
21. Cai, Z., Zheng, X., Yu, J.: A differential-private framework for urban traffic flows estimation via taxi companies. IEEE Trans. Industr. Inf. **15**(12), 6492–6499 (2019)
22. Michaël, D., Xavier, B., Pierre, V.: Convolutional neural networks on graphs with fast localized spectral filtering. Neural Inf. Process. Syst. **3**(1), 1–9 (2016)
23. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2008)
24. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)

25. Grover, A., Leskovec, J.: Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
26. Cai, Z., Zheng, X.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. **7**(2), 766–775 (2018)
27. David Cruz-Uribe, S.F.O., Moen, K.: One and two weight norm inequalities for Riesz potentials. Ill. J. Math. **57**(1), 295–323 (2013)
28. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
29. Sepp, H., Jürgen, S.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
30. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
31. Ling, Z., et al.: T-GCN: a temporal graph convolutional network for traffic prediction. IEEE Trans. Intell. Transp. Syst. **21**(9), 3848–3858 (2019)
32. Bing, Y., Haoteng, Y., Zhanxing, Z.: Spatiotemporal graph convolutional networks: a deep learning framework for traffic prediction. Int. Jt. Conf. Artif. Intell. Organ. **4**(1), 3634–3640 (2017)
33. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 922–929 (2019)
34. Qing, G., Zhu, S., Jie, Z., Yinleng, T.: An attentional recurrent neural network for personalized next location recommendation. In: AAAI Conference on Artificial Intelligence (AAAI 34), pp. 83–90 New York (2020)
35. Mason, J.C., Handscomb, D.C.: Chebyshev Polynomials. CRC Press (2002)