An Inferential Graph Convolution Network for Explaining Traffic Congestion

Qing Zhai, Jiayi Chen, Yifan Yin, Zi'ang Yang, and He Li*

Xidian University, Xi'an 710000, China
heli@xidian.edu.cn

Abstract. Due to the growth of vehicles, traffic congestion is becoming increasingly serious. However, existing methods are used for predicting traffic congestion, which cannot been applied for evaluating traffic congestion. In this paper, we propose an Interpretable Graph Convolution Network called ShapGCN for explaining the reason of traffic congestion by considering its physical and semantic neighbors. Specifically, we first design the physical neighbor embedding and semantic neighbor embedding to collectively encode complex extern factors as well as the complex traffic cascade pattern. To interprete traffic congestion in a complex traffic cascade environment, we use the approximation of shapley value to comprehensively quantify the discovered regions and their importance score. We conduct extensive experiments on the real traffic dataset. The experiment results show our ShapGCN can well explain the reason of traffic congestion.

Keywords: Traffic Congestion Prediction · Graph Convolution Network(GCN) · Explainable Analysis.

1 Introduction

In the process of urbanization, the number of cars has in creased excessively with the continuous expansion of urban boundaries. Some of the previous road structure cannot meet the current large traffic flow, resulting in serious traffic congestion. For example, the roads near the schools will encounter traffic jams at the morning peak and evening peak. If we can explain the causes of traffic congestion, the government can optimize road structure and traffic strategy to alleviate traffic pressure according to these reasons. Thus, investigating how to interpret the reason of traffic congestion is of paramount importance for city construction and traffic control.

Potential research works can be used for solving the traffic prediction for the detector network. In early days, the task is simple viewed as the prediction of of a multivariate time series. Therefore, time series models[16], converts non-stationary sequences into stationary sequences by differential methods and then makes predictions. But it is more suitable for some stationary sequences. This is far from the actual traffic flow changes, so the effect is not good. In addition, it tends to view the historical traffic flow of each area in isolation, without considering the spatial-temporal correlation between areas, which is too simplified for the problem, so it is used less now. Some researchers [25, 3, 2, 27] use CNN to capture spatial features. However, the above methods cannot be applied to non-Euclidean structure data. In recent years, some methods based on GNN [28, 8] are applied for traffic prediction.

Explaining traffic congestion is very challenging due to two aspects: i) complex traffic cascade pattern, traffic congestion is caused by the convergence of traffic flow from many roads. ii) complex external factors, the POI information, weather condition will affect traffic status.

However, most methods can only predict the traffic flow of different regiond and can not explain the reason of traffic congestion. In recent years, many scholars focus on explainable network model. The traditional explainable methods generate a sensitivity map for the input data to calculate the importance of the underlying substructures [19]. Gradient-based saliency maps[1], Class Activation Mapping (CAM) [24], and Excitation Backpropagation (EB) [24]. After that, two variants: gradient-weighted CAM (Grad-CAM) [18] and contrastive EB. They are usually applied for two different applications: visual scene graphs and molecular graphs. STANE [14] introduces a spatial-temporal attention mechanism to learn the attention parameters to fulfill the interpretation requirements. KerGNN [6] integrates graph kernels into the message passing process of GNNs and visualize the graph filters to show the important features of input data.

However, the above methods are designed for interpretable network model and cannot explain the reason of traffic congestion. In order to tackle the aforementioned problem, we propose a model called ShapGCN to explain the reason of traffic congestion. But we are still facing the following challenges for in interpreting traffic congestion to increase traffic controller understanding and trust in traffic congestion: i) complex traffic cascade pattern, traffic congestion is caused by the convergence of traffic flow from many roads. ii) complex external factors, the POI information, weather condition will affect traffic status.

In order to address these problems, we propose the Inferential Graph Convolution Network called ShapGCN for Explaining Traffic Congestion. First, we employ spatial-temporal position embedding to encode spatial-temporal position information. In addition, we introduce the feature mask matrix to mask some features to reflct the importance of these features. Finally, the approximation of shapley value is used to comprehensively quantify the discovered regions and their importance score for interpreting traffic congestion. In summary, our contributions in this paper are as follows:

1.We propose a model called ShapGCN, which can explain the reason of traffic congestion well by the the feature mask matrix.

2.We design a spatial-temporal position embedding to encode spatial-temporal position information in the complex traffic cascade environment.

3.We conduct extensive experiments on real-world datasets, showing our ShapGCN can well explain the reason of traffic congestion.

The rest of the paper is organized as follows. Section 2 shows the related work. In Section 3, we describe the definitions and studied problem, and then we present the architecture of ShapGCN in detail in Section 4. Section 5 give extensive experiments to verify the explainable performance. Finally, our work is concluded at the end of this paper.

2 Related Work

2.1 Traffic prediction

Traffic flow forecastTraffic flow forecasting is critical to urban safety, so many researches have emerged in recent years. The method used is also highly relevant to the development of deep learning in recent years.

The classic solutions are mainly traditional time series models or machine learning methods. The most classic of the time series models is Auto-Regressive Integrated Moving Average (ARIMA)[16], which converts non-stationary sequences into stationary sequences by differential methods and then makes predictions, and many models are derived[12, 13]. The classic machine learning methods Vector Auto-Regressions (VAR)[12] and and Support Vector Regression (SVR)[20]. have also been used, and are more suitable for some stationary sequences. This is far from the actual traffic flow changes, so the effect is not good. These methods tend to view the historical traffic flow of each area in isolation, without considering the spatio-temporal correlation between areas, which is too simplified for the problem, so it is used less now.

Deep learning has developed rapidly in recent years and has greatly improved the accuracy of traffic prediction. At the earliest, STDNN[26] mapped trajectories to grid maps and used DNN to make predictions, and achieved good results. After that, many methods of extracting spatio-temporal correlation using convolution emerged, such as STResNet[25], DeepFTP[3], MGSTC[2], MDL[27], etc[9, 15]. On the other hand, the RNN[5], LSTM[10], GRU[4] models, which are widely used in natural language processing, have also been successful. They are very suitable for modeling the temporal dependence of spatio-temporal data. In order to achieve better results, many works combine spatial correlation and temporal dependence, such as DMVST-Net[23], DeepUrbanEvent[23] etc.

Graph neural networks have received the attention of researchers after being paid attention to from graph representation learning[1, 7, 17] and used in the field of transportation. Graphs in the transportation field often refer to spatio-temporal graphs, indicating that the relationship between nodes will be affected by other nodes and time. In common methods, graph convolution is used to replace CNNs, such as T-GCN[28], ASTGCN[8]. DCRNN combines GCN and RNN models for traffic prediction. In addition, after the birth of the transformer, the combination of transformer and GNN has further promoted the solution of problems[22], such as GMAN[29] and STGNN[21].

2.2 Interpretable model

In recent years, many scholars focus on explainable network model. The traditional explainable methods generate a sensitivity map for the input data to calculate the importance of the underlying substructures [19]. Gradient-based saliency maps[19], Class Activation Mapping (CAM) [24], and Excitation Backpropagation (EB) [24]. After that, two variants: gradient-weighted CAM (Grad-CAM) [18] and contrastive EB. They are usually applied for two different applications: visual scene graphs and molecular graphs. STANE [14] introduces a spatial-temporal attention mechanism to learn the attention parameters to fulfill the interpretation requirements. KerGNN [6] integrates graph kernels into the message passing process of GNNs and visualize the graph filters to show the important features of input data.

3 Problem Definition

In this section, in order to describe our approach clearly, we formulate the basic symbol definition and problem statement in the paper.

3.1 Definition 1 (Traffic Network)

The traffic network is regarded as a graph represented by $G = (V, E, A)$ where V is the set of nodes and E is the set of edges. The number of nodes denotes $|V| = N$. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ is used to represent the nodes' proximities. The adjacency matrix only contains 0 and 1. If node V_i and node V_j are adjacent, A_{ij} in A is equal to 1. Otherwise, it is equal to 0.

3.2 Definition 2 (Graph Signal Matrix)

At each time interval t, the traffic network has a graph signal matrix $X_t \in \mathbb{R}^{N \times C}$, where C represents the number of attribute features.

3.3 Problem (Explainable traffic congestion)

The aims of explaining traffic congestion is using a feature mask matrix to mask some features to show the importance of these features.

4 Methodology

The overview of our framework is shown in fig 1, it can be divided into three core parts. 1) spatial-temporal neighbor embedding module 2) featue mask modules. 3) a output layer. In order to interpret model knowledge for a target class, our method first extracts features automatically and then computes an importance score for features of every neighbor according to our proposed ShapGCN. Finally, the importance scores of every neighbor for the whole class are given for class-wise explanation.

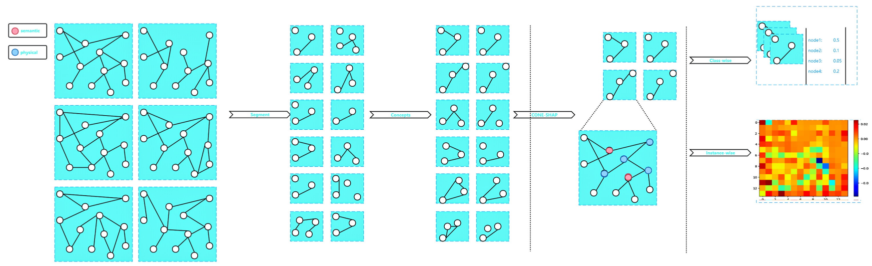


Fig. 1: The framework of our model

4.1 Fearues Discovery

Road feaures are defined as prototypes that are understandable for traffic congestion. Specifically, These feaures can be the position information, temporal information, etc. Since there are no user-defined features in real traffic scenarios, a method to discover traffic features automatically is needed. To extract such kind of traffic features, We denote the features as $C = \{C_1, C_2, ..., C_m\}$, where $C_i = \{c_{i,1}, c_{i,2}, .., C_{i,m}\}$, $c_{i,j}$ denotes the j^{th} feature in the i^{th} region, m is the number of the traffic features. To avoid missing the meaningful features information, we have considered both physical neighbors and semantic neighbors. We evaluate the importance of every feature of physical neighbors and semantic neighbors. We can obtain the physical neighbors by the road structure and semantic neighbors by DTW algorithm.

4.2 Features-based Neighbor Shapley

To measure the contribution of a features in an region for an explained model, we apply a counterfactual method which considers how the prediction of the model will change if this segment is absent. For prediction tasks, let g be the last layer before the softmax operation and g_k represents the logit values of feature k. Similar to CONE-SHAP[11], the value of a node of features k for the model is calculated as:

$$v_k(s) = g_k(x) - g_k(x \setminus \{s\}). \tag{1}$$

For convenience, we denotes $v_k(s)$ as $v(s)$.

Shapley Value. We consider all the N segments in a traffic image as a union, and each of them is a player. For a particular player i, let S be a subset that contains player i and $S \setminus \{i\}$ denotes the subset without the participation of i, then the contribution of i to the subset S is computed as:

$$\triangle v(i, S) = v(S) - v(S \setminus \{i\}). \tag{2}$$

Where $v(\cdot)$ is the utility function, then $\triangle v(\cdot)$ becomes the marginal contribution of the Shapley value. Thus,the Shapley Value of play i is the weighted average of marginal contribution in all of the subset:

$$\phi_v(i) = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{C_{j-1}^{N-1}}\sum_{S\in S_j(i)}\triangle v(i,S), \tag{3}$$

where $S_j(i)$ denotes the set with size j that contains the i_{th} segment. However, as the number of players increases, the computational complexity of the Shapley value grows exponentially. Since each traffic image contains more than a hundred segments, it is expensive for a computer to compute the true Shapley value. Therefore, recent studies have replaced the true Shapley values with approximations in different cases.

Approximation of Shapley Value. The regions can be treated as the nodes of a fully connected graph, where any two players are connected since they might have correlations during a game. In the application of image classification, the segments of an image can also be treated as nodes, but each node only connects with its neighbors. Here, we define the neighbors $N(i)$ of the i_{th} segment are those segments which are adjacent to it (physical neighbors) or belong to the same concept as it (semantic neighbors). Based on the assumption that participants which are not the neighbors of i hardly affect its contribution for a model's inference procedure, the Shapley Value of i in Equation above can be approximated as:

$$\phi_v^N(i) = \frac{1}{|N(i)|}\sum_{j=1}^{|N(i)|}\frac{1}{C_{j-1}^{|N(i)|-1}}\sum_{i\in S}\triangle v(i,S). \tag{4}$$

Considering that a segment may contain a large amount of neighbors in an instance, we adopt sample-based method to estimate $\phi_v^N(i)$ in order to further reduce the computation costs. Concretely, we first sample k nodes from $N(i)$ and denotes it as $N_k(i)$, and then compute the Shapley Value in the $N_k(i)$. This procedure will repeat M times, and we take the average of these results as the COncept-based NEighbor Shapley Value(CONE-SHAP) of i:

$$\tilde{\phi}_v^N(i) = \frac{1}{M|N_k(i)|}\sum_{t=1}^{M}\sum_{j=1}^{|N_k(i)|}\frac{1}{C_{j-1}^{|N_k(i)|-1}}\sum_{i\in S}\triangle v(i,S). \tag{5}$$

Next, we will introduce how to employ the approximation of Shapley Value from Equation 11 to interpret model knowledge from both instance-wise and class-wise.

4.3 Model Explaining

Instance-wise Explanation. In order to help users understand the basis for a model's reasoning procedure intuitively, we provide concept-based saliency maps to interpret model knowledge on the instance-level. The contribution of each segment of each instance is assigned according to its CONE-SHAP Value

$\phi_v^{\tilde{N}}(i)$. Compared to perturbation-based methods which explain a model in fine-grained features, our CONE-SHAP focuses on the concept-based explanation, which is more human-friendly.

Class-wise Explanation. To interpret model knowledge on the class level, our method distributes the concept scores to indicate which concept contributes more to the model's prediction on the explained class. A concept is considered important if all of its belongings own a high Shapley Value. Since we have gotten a group of possible concepts in the concept discovery procedure, for a concept C_i, we compute its score by averaging all of the approximate Shapley Values of its segments:

$$CS_i = \frac{1}{|SC_i|} \sum_{c_{i,j} \in C_i} \tilde{\phi}_v^N(i). \tag{6}$$

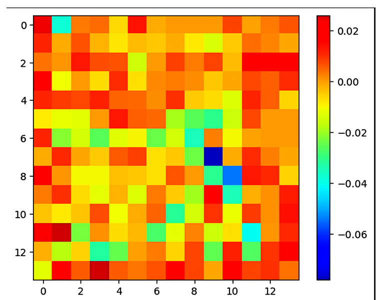


Fig. 2: The Heat Maps

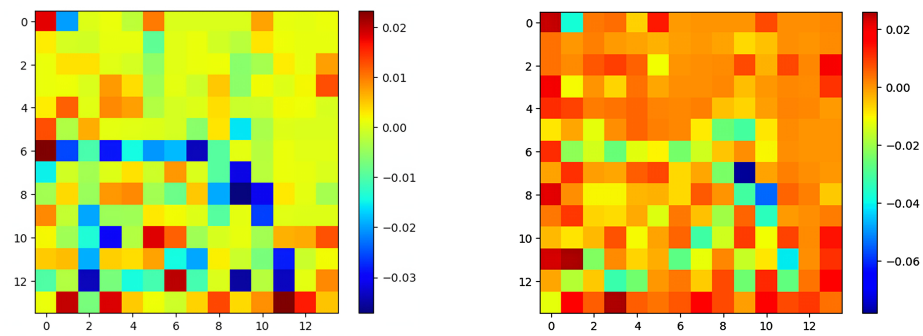
5 EXPERIMENTS

5.1 Experimental Settings

Table 1: Gradually increase the value of M

Metrics	M=1	M=2	M=3	M=4
Mean	0.00405483	0.00202741	0.00135161	0.00101370
Std	0.07392439	0.36962191	0.02464146	0.01848109

Our method can be applied to any task without any further training. To intuitively demonstrate the superiority of our method, we focus on the congestion-level classification of traffic flows in this paper.



(a) (b)

Fig. 3: Comparing two heatmaps at different times .

Table 2: Gradually increase the value of k

Metrics	k=2	k=3	k=4	k=5
Mean	0.00201455	-0.00235587	-0.00124802	-0.00149922
Std	0.06388019	0.01326836	0.00777133	0.00885320

Dataset. The data set we use comes from the Chengdu urban traffic flow data set in November 2016, which contains data of 196 nodes every half hour for 30 days in November. Then we use the average of node's concept scores for class-wise interpretation, and select the data of the first half hour of the third day for instant-wise interpretation.

Settings for neighbors. For each explained instance, we first calculate the inflow and outflow information of each node in different time periods, and then obtain the semantically adjacent neighbors of these nodes by calculating the flow information of these nodes. The meaningful semantic neighbors of nodes are different in different node positions, we use the *DTW* algorithm to find semantic neighbors similar to the node timing. Then, the physical neighbors physically adjacent to the node are calculated through the space-time matrix obtained from the data set. The union of the last two neighbor sets is the node's neighbor set.

5.2 Instance-wise Explanation

Explanation with Heat Maps. We provide the same fine-grained heat map as the input node to indicate which node of the instance is more important to the congestion situation of the traffic flow. Since we get the approximate Shapley value of each node at each timestamp, we treat these values as the score the node gets, and then for each node, we average the approximate Shapley values of the node at all moments to obtain the final scores and display them on a heat map.

Table 3: Physical neighbors and Semantic neighbor

Neighbors	physical	semantic	both
Mean	-0.00201529	0.00020384	0.00235587
Std	0.01425835	0.01326836	0.01326836

Figure 4 shows our CONE-SHAP heatmap for each node, where the importance score for each node is scaled between -1 and 1 by dividing by the absolute value of the largest number, for unit settings with scores below 0 is blue, over zero is set to red. On the graph you can see that our concept-based heatmap is easier for humans to understand.

Different Importance of Concepts on Different Instances. Intuitively, even the same node might have different importance to different instances. Based on the CONE-SHAP value at each timestamp, our method can estimate the importance of each node on each instance as follows: Firstly, we find out all the nodes in the instance. Then, for each node, we calculate the CONE-SHAPE value belonging to the node of Equation 11. Finally, we can estimate concept importance by summing the CONE-SHAPE values of its nodes. Figure 2 illustrates an example of our CONE-SHAPE interpretation of a traffic flow congestion class classification model. At the class level, our method shows that some nodes are very important for the recognition of the overall traffic jam, while in specific instances, the importance of these nodes varies. For example, in the first picture of Figure 3, the most important nodes are nodes 1, 2, 3, but in the second picture, the importance scores of these three points are not the most important.

5.3 Class-wise Explanation

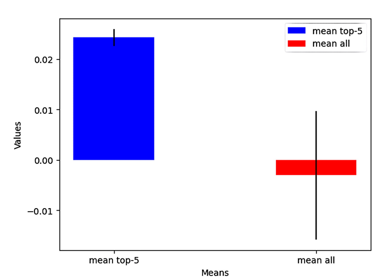


Fig. 4: The top-5 node's Mean

Validating the Performance of Concepts. To measure the most important concepts in the explanatory model, we compare the mean and variance of the five nodes with the highest importance scores in the explanatory model to the mean variance of the population.As shown in Figure 4:The average value of the five nodes with the highest importance score is much greater than the average value of all nodes.

Analysis of the Hyperparameters for Approximating Shapley Value. We approximate the true Shapley value by sampling from the neighbors, as shown in *Equation* 11. Too large M and k will bring pressure on the computational cost, while too small M and k will lead to inaccurate estimation. So we experimented to find the right M and k. In order to choose M, we first fix the value of k to 3, gradually increase the value of M to observe the change of the population mean, the results shown in Table 1 show that in our setting, setting M to 1 is enough to approximate the Shapley value . Similarly, we set M to 1 and gradually increase k. We found that when k is 2, a higher Shapely value is reached, as shown in Table 2. Therefore, we set M to 1 and k to 2 in our experiments.

Ablations experiment. In our experiments, the neighbor nodes of a node include semantic nodes and physical nodes, so we use ablation experiments to observe whether the two types of neighbor nodes have a great impact on the calculation of the final result. As shown in table 3, when considering two types of neighbors at the same time, the effect is better than only considering one type of neighbor nodes, which shows that the idea of considering two types of neighbor nodes in our method is reasonable.

6 Conclusions

In this paper, we propose a model named ShapGCN to achieve a better explanation of traffic congestion. We design the spatial-temporal position embedding and propose the spatial-temporal convolution module. By using the approximation of shapley value to comprehensively quantify the discovered regions and their importance score, ShapGCN has the ability to interprete traffic congestion in a complex traffic cascade environment. We evaluate our ShapGCN on the real dataset and the results show that our model achieve great interpretability.

References

1. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM international on conference on information and knowledge management. pp. 891–900 (2015)
2. Chen, C., Li, K., Teo, S.G., Zou, X., Li, K., Zeng, Z.: Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. ACM Transactions on Knowledge Discovery from Data (TKDD) **14**(4), 1–23 (2020)
3. Chen, Y., Chen, F., Ren, Y., Wu, T., Yao, Y.: Poster: Deeptfp: Mobile time series data analytics based traffic flow prediction. In: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking. pp. 537–539 (2017)
4. Cho, K.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)

5. Connor, J.T., Martin, R.D., Atlas, L.E.: Recurrent neural networks and robust time series prediction. IEEE transactions on neural networks **5**(2), 240–254 (1994)
6. Feng, A., You, C., Wang, S., Tassiulas, L.: Kergnns: Interpretable graph neural networks with graph kernels. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 6614–6622 (2022)
7. 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)
8. 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, pp. 922–929 (2019)
9. Guo, S., Lin, Y., Li, S., Chen, Z., Wan, H.: Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. IEEE Transactions on Intelligent Transportation Systems **20**(10), 3913–3926 (2019)
10. Hochreiter, S.: Long short-term memory. Neural Computation MIT-Press (1997)
11. Li, J., Kuang, K., Li, L., Chen, L., Zhang, S., Shao, J., Xiao, J.: Instance-wise or class-wise? a tale of neighbor shapley for concept-based explanation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3664–3672 (2021)
12. Lippi, M., Bertini, M., Frasconi, P.: Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. IEEE Transactions on Intelligent Transportation Systems **14**(2), 871–882 (2013)
13. Liu, H., Tian, H.q., Li, Y.f.: Comparison of two new arima-ann and arima-kalman hybrid methods for wind speed prediction. Applied Energy **98**, 415–424 (2012)
14. Liu, Z., Zhou, D., He, J.: Towards explainable representation of time-evolving graphs via spatial-temporal graph attention networks. In: Proceedings of the 28th ACM international conference on information and knowledge management. pp. 2137–2140 (2019)
15. Ma, X., Zhong, H., Li, Y., Ma, J., Cui, Z., Wang, Y.: Forecasting transportation network speed using deep capsule networks with nested lstm models. IEEE Transactions on Intelligent Transportation Systems **22**(8), 4813–4824 (2020)
16. Min, W., Wynter, L.: Real-time road traffic prediction with spatio-temporal correlations. Transportation Research Part C: Emerging Technologies **19**(4), 606–616 (2011)
17. 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)
18. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
19. Simonyan, K.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
20. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing **14**, 199–222 (2004)
21. Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., Yu, J.: Traffic flow prediction via spatial temporal graph neural network. In: Proceedings of the web conference 2020. pp. 1082–1092 (2020)
22. Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.J., Xiong, H.: Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908 (2020)

23. Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z.: Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
24. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. International Journal of Computer Vision **126**(10), 1084–1102 (2018)
25. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
26. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: Dnn-based prediction model for spatio-temporal data. In: Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems. pp. 1–4 (2016)
27. Zhang, J., Zheng, Y., Sun, J., Qi, D.: Flow prediction in spatio-temporal networks based on multitask deep learning. IEEE Transactions on Knowledge and Data Engineering **32**(3), 468–478 (2019)
28. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H.: T-gcn: A temporal graph convolutional network for traffic prediction. IEEE transactions on intelligent transportation systems **21**(9), 3848–3858 (2019)
29. Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attention network for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 1234–1241 (2020)