2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

Recurrent Attention Unit: A Simple and Effective Method for Traffic Prediction

Zebing Wei, Zhishuai Li, Chunxiang Wang, Yuanyuan Chen, Qinghai Miao†, Yisheng Lv† and Fei-Yue Wang

***Abstract*— Recurrent neural networks are widely used in sequential data modeling. For example, long-short term memory network (LSTM) and gated recurrent unit(GRU) are two typical traffic prediction methods. It is noted that the gate structure is the key component of LSTM and GRU while increasing the number of training parameters against traditional RNN. In this paper, inspired by the function of attention mechanism in regulating information flow, we propose a simple yet effective method for traffic prediction which embeds the attention mechanism within the recurrent module attempting to focus on the important information of inside features. The proposed model structure is named as RAU, which is short for the recurrent attention unit. We evaluated the proposed methods on five real-world datasets. Extensive experiments show that the proposed method can achieve comparable prediction performances against LSTM and GRU, while decrease more than 30% and 50% parameters, respectively.**

I. Introduction

As one of the key components of intelligent transportation systems [1], traffic prediction plays an important role in advanced traffic management and control systems, advanced traffic information systems [2]–[5], etc. It is known that traffic flow dynamics is a complex spatial and temporal process of vehicle interactions, thus it is a challenging problem for traffic prediction. Due to the rise of big data and deep learning, traffic prediction has emerged as a promising research field, recently.

Traffic prediction has a long research history, and many traffic prediction methods have been proposed [6]–[9]. Typical prediction methods are mainly based on statistical and traditional machine learning methods, $e.g.$ historical average (HA), Vector Auto-Regressions (VAR) [10], Auto-Regressive Integrated Moving Average (ARIMA) model [11], and Support Vector Regression (SVR) [12]. Usually, these methods are suitable for small-scale datasets like datasets from a few sensor stations. In the last decades, with the emergence of traffic big data and the success of deep learning methods, there is shifting from traffic prediction for a few sensor stations to large-network-wide traffic prediction. Typical deep learning based methods for traffic prediction include stacked autoencoders (SAE) [13], long-short term memory network (LSTM) [14], and graph-based neural networks [15]. Kang $et\ al.$ used the LSTM model while considering neighboring spatial traffic flow features for traffic flow prediction [16]. As a variant of LSTM, gated recurrent unit (GRU) [17] is also adapted to predict traffic flow. Considering the non-Euclidean spatio-temporal correlation in traffic data, Li $et\ al.$ [18] modeled traffic flow as a diffusion process on a directed graph, and proposed a diffusion convolution recurrent neural network (DCRNN) to predict traffic speed. It is popular to combine different modular structures within a single deep learning model to improve performance. Zheng $et\ al.$ combined the attention mechanism and LSTM to capture the important features in hidden layers [19]. Li $et\ al.$ designed a multi-stream feature fusion prediction model which is composed of graph convolutional neural network (GCN), GRU, feedforward neural network(FNN) and, the attention mechanism [20].

The attention mechanism is one recent advancement in deep learning which is widely used in natural language processing and computer vision tasks. It can flexibly select context information, which is a popular block for sequential data prediction [21], [22]. It is noticed that the attention mechanism is usually put at the ending parts of the deep network and outside the recurrent modules. What if we put the similar attention mechanism within the recurrent block for traffic prediction to dynamically and recurrently capture traffic features? This question motivates us to design a recurrent attention unit for traffic prediction.

In this paper, we design a novel traffic prediction model based on recurrent neural networks and the attention mechanism named the recurrent attention unit (RAU). We embed the attention mechanism within the recurrent neural network to make the model focus on critical features within each time step and expand the attention elements over the temporal dimension in a recurrent way. The main contributions of this paper are summarized as follows.

- We design a recurrent attention unit for traffic prediction. The proposed method embeds the attention mechanism within the recurrent neural module. In addition, we adopt weighted residual connection to restrain gradient vanishing/explosion problems when training the deep model.
- We performed extensive experiments on five real-world datasets. Experimental results show that the proposed method can achieve comparable prediction performances against LSTM and GRU, while decreases more than 30%

This work was partially supported by National Key R&D Program of China (2020YFB2104001), National Natural Science Foundation of China under Grants U1811463 and 61876011, Chinese Guangdong's S&T project (2019B1515120030, 2020B0909050001).

Z. Wei, Z. Li, C. Wang, Y. Lv, Y. Chen and F. Wang are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China. They are also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China.(e-mail: weizebing2019@ia.ac.cn; yisheng.lv@ia.ac.cn).

Q. Miao is with the University of Chinese Academy of Sciences, Beijing, 100049, China (e-mail: miaoqh@ucas.ac.cn).

† Corresponding author: Qinghai Miao, Yisheng Lv.

2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564984

and 50% parameters, respectively.

The rest of this paper is organized as follows. Section II describes the methodology, which states the traffic prediction problem and the proposed RAU in detail. Section III presents experimental results, Section IV concludes this paper.

II. Methodology

In this section, we first introduce the traffic prediction problem. Then we present the basics of recurrent neural networks. Finally, we describe in details the proposed model structure, namely RAU, which embeds the attention unit as the recurrent module.

A. Traffic prediction problem

The traffic prediction task aims to use the collected historical data (traffic flow, traffic speed, *etc.*) in the road sensor network for future traffic state prediction. Taking traffic flow prediction as an example, historical traffic flow data collected from detector stations can be expressed as time-series data. Suppose the traffic flow collected from n detector stations can be denoted as

$$\boldsymbol{\mathcal{X}} = [\mathbf{X}_1, \mathbf{X}_2, ...\mathbf{X}_n] = \begin{bmatrix} x_1^1 & x_2^1 & ... & x_t^1 \\ x_1^2 & x_2^2 & ... & x_t^2 \\ x_1^3 & x_2^3 & ... & x_t^3 \\ ... & ... & ... & ... \\ x_1^n & x_2^n & ... & x_t^n \end{bmatrix}, \quad (1)$$

where the historical traffic flow of the i-th detectior station is $\mathcal{X}^i = [\mathbf{X}_1^i, \mathbf{X}_2^i, ...\mathbf{X}_n^i]$, and each element x_m^i represents the cumulative sum of the i-th monitor's traffic flow over a period of time (e.g. 5min or 10min).

After getting the expression $\mathcal{X}$ which means the historical traffic flow of all station, the future traffic flow prediction can be described as

$$\hat{\boldsymbol{Y}} = \mathcal{G}(\boldsymbol{\mathcal{X}}, \boldsymbol{\theta}), \quad (2)$$

Where $\mathcal{G}$ means a defined regression function, and $\boldsymbol{\theta}$ represents its parameters. The objective of model training is to find the optimal parameters $\boldsymbol{\theta}^*$ by minimizing the loss function which fomulated as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{Y}, \mathcal{G}(\boldsymbol{\mathcal{X}}; \boldsymbol{\theta})\right), \quad (3)$$

where Y is the true value of future traffic flow and $\mathcal{L}$ represents the loss function which need be defined before model training.

B. Recurrent Neural Networks

The recurrent neural network has advantages over processing sequential data with contextual dependency [23]. The hidden state of RNN cell at time step t can be calculated with

$$\mathcal{H}_t = \Gamma(\boldsymbol{w}([\mathcal{H}_{t-1}, x_t] + \boldsymbol{b})), \quad (4)$$

where Γ is the activation function, x_t is the input, $\mathcal{H}_{t-1}$ is the hidden state at time step $t-1$, and w, b are the trainable weights.

Traditional RNN has the problems of gradient vanishing/explosion. To address the problems, its variants, such as LSTM [14], and GRU [17] are proposed, in which the gate structure is applied and increases the size of training parameters which needs mores computing resources.

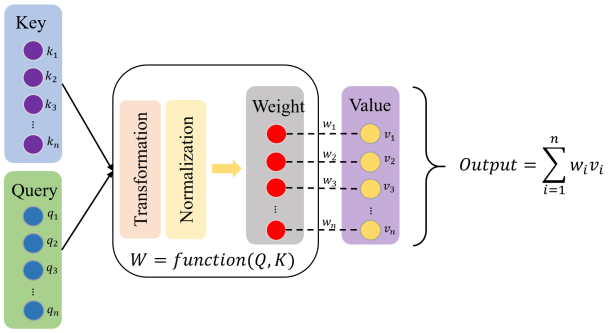


Fig. 1: The schematic explanation of attention mechanism.

C. Attention Mechanism

The attention mechanism arises from the human visual information processing, which gives emphasis on relevant parts of the task. Recently, it is used to design deep neural network models. Applying the attention mechanism to sequential data processing enables the model to focus on those elements which are beneficial to learning.

In general, the attention mechanism has the ability to model the dependency between values and target by adaptively assigning normalized weights under the mapping of queries and keys. Fig. 1 illustrates the holistic explanation of the attention mechanism. The main idea is to define the mapping relationship between the query and the key-value pairs, and obtains a weighted sum of values as the output. The weights are normalized by $softmax$ function which denote the relationship strength between queries and key-value pairs. The specific process can be formulated as

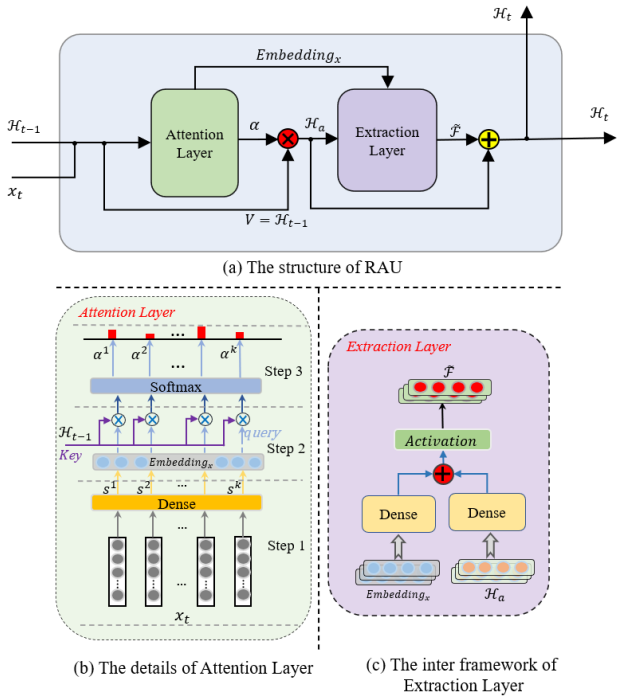
$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{\mathcal{D}_{dim}}})\mathbf{V}, \quad (5)$$

where $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V}$ define queries, keys and values respectively. $\mathcal{D}_{dim}$ represents the corresponding demension.

D. Recurrent Attention Unit

The original intention of designing a recurrent attention unit is to reduce parameters while improving RNN performance by utilizing the attention mechanism. In essence, attention mechanism and gate structure(implemented in LSTM/GRU) are similar in regulating the flow of information, while attention mechanism has the superiority of not requiring too many parameters.

In general, in the field of traffic prediction, the length of the input sequence is always short. Therefore, the task itself prevents the models from gradient vanishing/explosion to some extent while using recurrent neural networks. When dealing with traffic prediction problems, we embed the attention mechanism within the recurrent structure to focus on the information from input data that is more favorable to the learning prediction task. Fig. 2 shows the details of



(a) The structure of RAU

(b) The details of Attention Layer

(c) The inter framework of Extraction Layer

Fig. 2: The details of RAU structure. (a) The overall framework of RAU. (b) The details and the calculation procedure of internal Attention Layer. (c) The internal framework of Extraction Layer.

the structure, which is composed of two parts named the attention layer and extraction layer.

The process of the attention layer can be divided into three steps:

- Step 1 can be generalized as transformation operation which is designed to convert input data x_t and obtains useful information marked as $Embedding_x$.
- Step 2 is to calculate the score between key($\mathcal{H}_{t-1}$) and query gained form step 1.
- Step 3 is the normalization operation by which we can get the attention weights α.

The extraction layer is applied to extract features and fuse the useful information from input data with hidden state of last time step. Specifically, in the attention layer, we design a transformation operation to extract the input feature as the query and the calculation process can be formulated as

$$Embedding_x = \Gamma(\boldsymbol{w_a} x_t + \boldsymbol{b_a}), \tag{6}$$

$$\alpha^{(i)} = softmax(Embedding_x^{(i)} \cdot \mathcal{H}_{t-1}^{(i)}), \tag{7}$$

$$\mathcal{H}_a^{(i)} = \sum_{i=1}^{k} \alpha^{(i)} \mathcal{H}_{t-1}{}^{(i)}, \tag{8}$$

where $\mathcal{H}_{t-1}$ is the hidden state at time step $t-1$, x_t is the input, $\boldsymbol{w_a}$ and $\boldsymbol{b_a}$ are the weight matrix and bias, respectively. α is the attention weight which represents the degree of correlation between $\mathcal{H}_{t-1}$ and $Embedding_x$.

As showing in Eq.8, we retain useful historical information of $\mathcal{H}_{t-1}$ by weighting with α. Then fusing $\mathcal{H}_a$ with $Embedding_x$ by which we merge current input data with useful historical states. The specific process defined as

$$\tilde{\mathcal{F}} = Tanh(\boldsymbol{w_h}([\mathcal{H}_a, Embedding_x] + \boldsymbol{b_h})), \tag{9}$$

where $\tilde{\mathcal{F}}$ represents fusion feature, w_h and b_h are the trainable parameters.

We further skillfully design a weighted residual connection to restrain gradient vanishing and gradient explosion, and obtain the final output of each time step which is shown as

$$\mathcal{H}_t = \tilde{\mathcal{F}} + \boldsymbol{\lambda} \mathcal{H}_a, \tag{10}$$

During the back propagation, the loss of the current hidden state $\mathcal{H}_t$ is feedback to $\mathcal{H}_{t-1}$:

$$\frac{\partial \mathcal{L}}{\partial \mathcal{H}_{t-1}} = \frac{\partial \mathcal{L}}{\partial \tilde{\mathcal{F}}} \frac{\partial \tilde{\mathcal{F}}}{\partial \mathcal{H}_a} \frac{\partial \tilde{\mathcal{H}}_a}{\partial \mathcal{H}_{t-1}} + \frac{\partial \mathcal{L}}{\partial \alpha} \frac{\partial \alpha}{\partial \mathcal{H}_{t-1}} + \lambda \tag{11}$$

The second part in equation (11) indicates that attention mechanism inside gives an extra gradient, and the constant λ in the third part manifests that the weighted residual connection plays a positive role in avoiding gradient vanishing and exploding. They can help improve the model.

III. Experiments

In this section, we firstly introduce the datasets used for evaluating the proposed model. Then we present experimental results.

A. Dataset Description

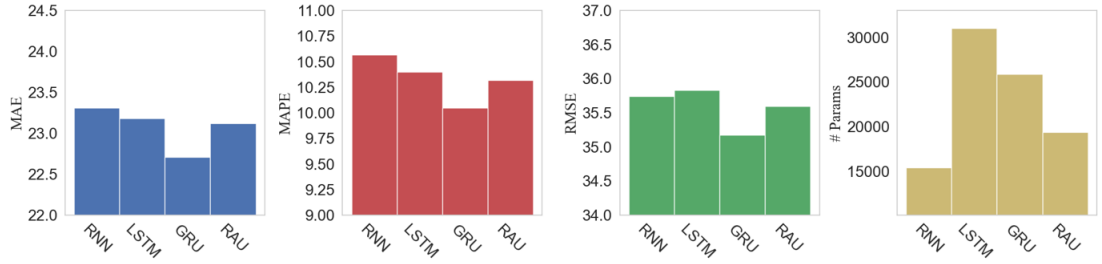
We perform experiments on five real-world traffic datasets detailed in Table I. PeMS-SJC contains 30 loop detectors' data in a section of San Jose City (SJC), we collected it in California Department of Transportation Performance Measurement System (PeMS) from July 1, 2019 to September 30, 2019. The 30 loop detectors located in the positions that have obvious upstream and downstream relationships. We aim to verify the multi-station(contains significant spatial correlations) feature extraction ability by implementing experiments on this dataset.

PeMS03, PeMS04, PeMS08 are three open-source datasets with larger scale which commonly used in traffic prediction. They are highway traffic flow data collected from PeMS in different regions. PeMSD7(L) is a traffic speed open-source dataset that contains 1026 monitors. These four datasets

TABLE I: DATASETS DESCRIPTION AND STATISTICS

# Datasets	# Stations num	# Time length
PeMS-SJC	30	07/01/2019-09/30/2017 (90 Days)
PeMS03	358	09/01/2018-11/30/2018 (91 Days)
PeMS04	307	01/01/2018-02/28/2018 (59 Days)
PeMS08	170	07/01/2016-08/31/2016 (62 Days)
PeMSD7(L)	1026	Weekday in May and June of 2012 (45 Days)

(a) 7 stations



(b) 15 stations

Fig. 3: The prediction results using 7 and 15 consecutive stations to prediction traffic flow in the next 5 minutes.

are used to identify the proposed method's performance on different traffic prediction tasks(traffic flow/speed prediction) and further verify its modeling capability on large road networks. All the samples of four datasets are collected with a time interval of 30 seconds and aggregated every 5-minute.

B. Index of Performance

In this paper, we choose three commonly used performance metrics: mean absolute error(MAE), root mean square error(RMSE) and mean absolute percentage error(MAPE) to evaluate the effectiveness of the proposed method. They are defined as

$$MAE = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} |y_i - \hat{y}_l| , \tag{12}$$

$$RMSE = \sqrt{\frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (y_i - \hat{y}_l)^2}, \tag{13}$$

$$MAPE = \frac{100\%}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left| \frac{y_i - \hat{y}_l}{y_i} \right| , \tag{14}$$

where $\hat{y}_i$ is the predicted value, y_i is the truth, and $\mathcal{N}$ is the total number of samples. For all of the three metrics, the smaller the value is, the better the model performs.

C. Comparable Experiments on Small-scale Dataset

We first test the proposed method on PeMS-SJC dataset and compare it with RNN, LSTM, and GRU to demonstrate its performance. Specifically, we perform two groups of experiments: 7 and 15 consecutive stations selected from PeMS-SJC. We process the dataset into sequential samples in a sliding window way, and use the historical data of 60 minutes to predict the traffic flow in the next 5 minutes. After obtaining the sampled data, we construct the training set, validation set, and test set with the ratio 6:2:2. All experiments are performed on a computer with an Intel Core i7-9700 CPU and an Nvidia GeForce GTX 1660Ti.

The experimental results are shown in Fig. 3. All the results are the average value of multiple-runs of repeated experiments. The impressive results illustrate the superiority of the proposed method: the proposed model achieves competitive performance, while significantly reducing model parameters against LSTM, and GRU. More specifically, the training parameters of our method reduce by more than 30% and 50% compared with GRU and LSTM, respectively.

D. Experiments on Large-scale Datasets

We increase the complexity of the input and further implement experiments on PeMS03, PeMS04, PeMS08 (traffic flow datasets), and PeMSD7(L) (traffic speed datasets) datasets. We conducted extensive experiments which can be divided into two parts: the contrast experiments on RNN and its variants, and the module replacement experiments on the state-of-the-art benchmark. In part two, we choose MFFB [20] as the benchmark, and replace the GRU module used in this framework with RAU. Multi-stream feature fusion block(MFFB) models the spatial-temporal correlation with a multi-stream mechanism that combines with GCN, GRU and FNN, and fusing them with the soft attention mechanism. It shows excellent performance in traffic prediction.

For the details of experiments implementation, we keep the training set, validation set and test set in the percentage of 70%, 10%, 20% respectively, and choose Adam as the optimizer. The training epochs are set as 200 to ensure the models are adequately trained on datasets. Table II shows the average results of multiple-runs experiments over 60 minutes(12 step horizons).

From Table II, we can see that RAU possesses impressive performance. Comparing with RNN and its two commonly

TABLE II: The performance comparison on four large-scale and two categories datasets (PeMS03, PeMS04, PeMS08(traffic flow datasets) and PeMSD7(L)(traffic speed dataset)). Predicting the next one hour traffic data with 12 steps inputs. (the best results are in bold and * is the second-best results, ‡ denotes the initial structure of the baseline method.)

MODEL NAME		RNN(+FCs)	LSTM(+FCs)	GRU(+FCs)	**RAU**(+FCs)	MFFB(RNN)	MFFB(LSTM)	MFFB(GRU)‡	**MFFB(RAU)**
Datasets	Metrics								
PeMS03	MAE	20.46	20.02	20.14	**19.99**	17.45	16.96	**16.67**	16.93*
	MAPE(%)	19.74	19.59	19.48	**18.83**	17.10	16.40	16.60	**16.10**
	RMSE	35.03	**34.47**	34.48	34.63*	28.17	27.23	27.41	**26.82**
	# **Params**	27.1K(+54.5K)	108.5K(+54.5K)	81.41K(+54.5K)	31.2*K(+54.5K)	– – –	– – –	– – –	– – –
PeMS04	MAE	23.43	23.07	23.13	**23.01**	22.57	22.18	**21.31**	21.72*
	MAPE(%)	15.86	15.64	15.62	**15.61**	15.50	15.10	14.70	**14.60**
	RMSE	37.52	**37.10**	37.19	37.20	35.32	34.72	**33.69**	34.49*
	#Params	23.9K(+47.9K)	95.5K(+47.9K)	71.6K(+47.9K)	28.0*K(+47.9K)	– – –	– – –	– – –	– – –
PeMS08	MAE	23.02	**22.56**	23.00	22.78*	18.38	17.22	17.68	**16.85**
	MAPE(%)	13.77	**13.59**	13.81	13.71*	11.20	10.70	10.60	**10.10**
	RMSE	37.44	36.95	**36.61**	36.80*	28.10	27.40	27.25	**26.21**
	#Params	15.2K(+30.3K)	60.4K(+30.3K)	45.3K(+30.3K)	19.2*K(+30.3K)	– – –	– – –	– – –	– – –
PeMSD7(L)	MAE	4.85	4.69	4.73	**4.49**	3.54	3.52	**3.34**	3.36*
	MAPE(%)	11.91	11.60	11.73	**11.53**	8.70	8.60	**8.40**	8.50*
	RMSE	8.63	8.34	8.44	**8.26**	6.48	6.39	6.36	**6.33**
	#Params	69.9(+140.7)	279.0K(+140.7K)	209.7K(+140.7K)	74.0*K(+140.7K)	– – –	– – –	– – –	– – –

used variants(LSTM, GRU), RAU gains obvious better prediction accuracy than RNN, and even outperforms RNN, LSTM, and GRU on some datasets. Fig. 4 shows this more intuitively with the box plot of 10-runs experimental results.

To further analyze the experimental results on PeMSD7(L), it is evident that RAU performs well on both traffic flow and speed prediction tasks. And it seems that its performance is improved more significantly on PeMSD7(L) whose scale is almost 3∼6 times compared with PeMS03, PeMS04, and PeMS08. What's more, concentrating on the scale of parameters, we need to highlight that the parameter size of RAU is slightly higher than that of RNN, while it is reduced by more than 50% and 30% respectively compared with LSTM and GRU(It practically depends on the shape of the input vector.). This conclusion can also be obtained from Equation (6) ∼ (10). For the replacement experiments on MFFB, the model combined with RAU also shows competitive performance. Therefore, These two parts experiments on large-scale datasets strongly show the proposed method's superiority.

To explore how the internal attention mechanism regulates the flow of information, we visualized the attention weights after model training which is shown in Fig. 5. In the developed model, each sample is mapped to a higher dimension as features. However, the feature in different dimensions is of different importance. As Fig. 5 illustrated, The attention mechanism in RAU gains the ability to assign appropriate weights adaptively to the features in different dimensions by which it realizes the similar function of the gate structures and controls information flow adaptively.

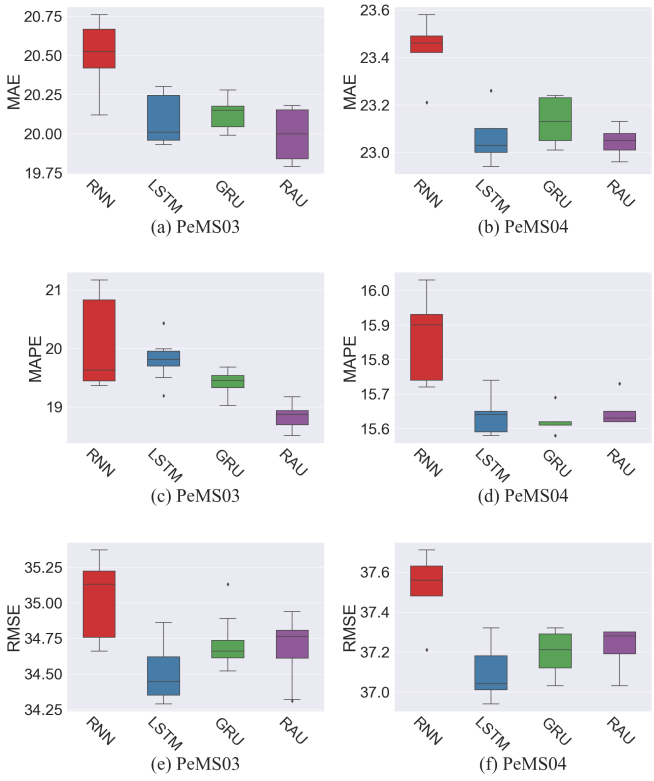


Fig. 4: Box-plot of experimental results on PeMS03 and PeMS04 datasets.

IV. Conclusion

In this paper, we propose a simple and effective model named RAU which embeds the attention mechanism within the recurrent neural network to conduct traffic prediction tasks.

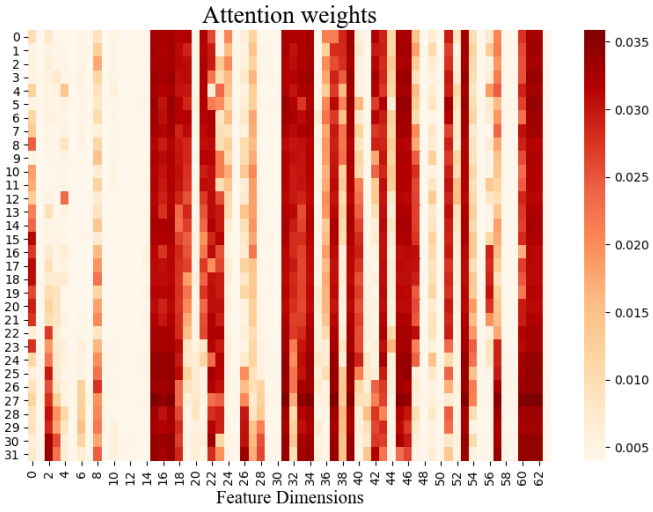


Fig. 5: The visualization of attention weights in RAU on the test set of PeMS04. The features in different dimensions are assigned different weights.

This method can capture the features which are beneficial to the training target (minimum loss) in the input vector at each time step, and focus on the selective important part of the features via attention weights. The attention mechanism embedded inside can both capture the crucial features effectively and expand the attention elements over temporal dimension in a recurrent way. We also add a weighted residual connection in the recurrent cell to restrain the gradient vanishing and gradient explosion problems. We carry out experiments on four real-world traffic datasets, and the results have shown the effectiveness of the proposed method. Compared with LSTM and GRU, the parameters of the proposed method are reduced more than 30% and 50%, respectively, and can get comparable and even better prediction performance. In the future, we will test the proposed method on other sequence prediction tasks.

REFERENCES

[1] Fei-Yue Wang. Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):630–638, 2010.

[2] Matthew Veres and Medhat Moussa. Deep learning for intelligent transportation systems: A survey of emerging trends. *IEEE Transactions on Intelligent transportation systems*, 21(8):3152–3168, 2019.

[3] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011.

[4] Runmei Li, Yinfeng Huang, and Jian Wang. Long-term traffic volume prediction based on k-means gaussian interval type-2 fuzzy sets. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1344–1351, 2019.

[6] Yanjie Duan, Yisheng Lv, and Fei-Yue Wang. Travel time prediction with lstm neural network. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pages 1053–1058. IEEE, 2016.

[5] Yuanyuan Chen, Hongyu Chen, Peijun Ye, Yisheng Lv, and Fei-Yue Wang. Acting as a decision maker: Traffic-condition-aware ensemble learning for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2020.

[7] Runmei Li, Chaoyang Jiang, Fenghua Zhu, and Xiaolong Chen. Traffic flow data forecasting based on interval type-2 fuzzy sets theory. *IEEE/CAA Journal of Automatica Sinica*, 3(2):141–148, 2016.

[8] Yadong Yu, Yong Zhang, Sean Qian, Shaofan Wang, Yongli Hu, and Baocai Yin. A low rank dynamic mode decomposition model for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[9] Chenyi Chen, Yin Wang, Li Li, Jianming Hu, and Zuo Zhang. The retrieval of intra-day trend and its influence on traffic prediction. *Transportation research part C: emerging technologies*, 22:103–118, 2012.

[10] Marco Lippi, Matteo Bertini, and Paolo Frasconi. 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.

[11] S Vasantha Kumar and Lelitha Vanajakshi. Short-term traffic flow prediction using seasonal arima model with limited input data. *European Transport Research Review*, 7(3):21, 2015.

[12] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[13] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2014.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 914–921, 2020.

[16] Danqing Kang, Yisheng Lv, and Yuan-yuan Chen. Short-term traffic flow prediction with lstm recurrent neural network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[17] Kyunghyun Cho, B van Merrienboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014.

[18] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

[19] Haifeng Zheng, Feng Lin, Xinxin Feng, and Youjia Chen. A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2020.

[20] Zhishuai Li, Gang Xiong, Yonglin Tian, Yisheng Lv, Yuanyuan Chen, Pan Hui, and Xiang Su. A multi-stream feature fusion approach for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2020.

[21] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1234–1241, 2020.

[22] Xuesong Li, Yating Liu, Kunfeng Wang, and Fei-Yue Wang. A recurrent attention and interaction model for pedestrian trajectory prediction. *IEEE/CAA Journal of Automatica Sinica*, 7(5):1361–1370, 2020.

[23] Yuan-yuan Chen, Yisheng Lv, Zhenjiang Li, and Fei-Yue Wang. Long short-term memory model for traffic congestion prediction with online open data. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 132–137. IEEE, 2016.