Traffic Matrix Prediction Based on Deep Learning for Dynamic Traffic Engineering

Zhifeng Liu*‡, Zhiliang Wang†‡, Xia Yin*‡, Xingang Shi†‡, Yingya Guo*‡ and Ying Tian*‡
*Department of Computer Science and Technology, Tsinghua University
†Institute for Network Sciences and Cyberspace, Tsinghua University
‡Beijing National Research Center for Information Science and Technology
Beijing, P.R. China
Email:{liuzf17, guo-yy13, y-tian18}@mails.tsinghua.edu.cn, {wzl, shixg}@cernet.edu.cn, {yxia}@tsinghua.edu.cn

***Abstract*—Traffic matrix (TM) is a critical information for network operation and management, especially for traffic engineering (TE). Due to the technical and mercantile problems, real time measurement for TM is difficult in large scale networks. In this paper, we focus on predicting TM for dynamic traffic engineering. We propose several TM prediction methods based on Neural Networks (NN) and predict TM from three perspectives: predict the overall TM directly, predict each origin-destination (OD) flow separately and predict the overall TM combined with key element correction. In addition to the prediction accuracy, we evaluate different prediction methods through the performance of TE, as well as the prediction time. We test the proposed methods by real world datasets from Abilene, CERNET and GÉANT. The experiment results show that prediction methods based on Recurrent Neural Networks (RNN) can achieve better prediction accuracy than methods leveraging Convolutional Neural Networks (CNN) and Deep Belief Networks (DBN). Predicting each OD flow through RNN models can further improve the prediction accuracy, as well as the performance of TE under the OSPF network scenario, the SDN/OSPF hybrid network scenario and the multi-commodity flow problem scenario. However, it takes longer prediction time for predicting each OD flow sequence. In contrast, predicting the overall TM combined with key element correction can provide a trade-off between the TE results and the prediction overhead, which is more appropriate for dynamic TE in the above three scenarios.**

***Index Terms*—traffic matrix prediction, traffic engineering, neural networks**

I. Introduction

Network traffic matrix (TM) represents the volume of traffic flows between all possible pairs of origin and destination (OD) routers in the network, which plays an important role in the areas of network management and planning [1]. For example, TM is a crucial input for most traffic engineering (TE) algorithms in different network scenarios [2]–[4]. However, despite of its importance, direct measurement on TM has been proved to be difficult due to the technical and economic issues [5]. Therefore, based on the self-similarity characteristic of network traffic [6], researchers try to estimate or predict TM instead of measuring it directly.

Several methods have been proposed for TM prediction. [5] and [7] use statistical and linear models such as Gaussian distribution and Autoregressive Integrated Moving Average (ARIMA) to model network traffic, but those traditional models cannot well extract the nonlinear nature of network traffic [6]. [8]–[14] establish time series models for TM sequences and OD flow sequences based on Neural Networks (NN). They evaluate different prediction methods by comparing the prediction error between predicted TMs and actual values, and prove that NN based methods can achieve higher prediction accuracy than traditional linear models. However, prediction methods with small prediction error are not always suited to TE [15], and those works have not investigated the influence of different prediction methods on TE performance. [15] predicts the long-term traffic demands through ARIMA and Seasonal Autoregressive Integrated Moving Average (SARIMA), and uses the standard deviation to estimate the range of short-term fluctuation. It evaluates different prediction methods by the results of TE instead of the prediction accuracy. But those linear models are insufficient for network traffic prediction due to the nonlinear nature of network traffic. Moreover, the TE scenario considered in [15] is a simple multi-commodity flow (MCF) problem, which is not applicable in actual networks.

In this paper, we study TM prediction methods for dynamic traffic engineering. In the dynamic TE scenario, the routing strategy of each time interval will be calculated in the beginning based on the predicted TM so as to avoid congestion. We consider both the traditional OSPF network scenario and the Software Defined Networking (SDN) scenario, and select two representative TE algorithms: optimizing OSPF weights [2] and SDN/OSPF hybrid network traffic engineering (SOTE) [4]. [2] first introduces a local search heuristic algorithm for optimizing OSPF weights in OSPF networks. [4] is the first paper to optimize both the OSPF weight of each link and the flow splitting ratio of SDN nodes. In addition, we regard the result of MCF problem as the theoretical optimal solution of TE. We propose several TM prediction methods based on NN. In addition to predicting TM directly, we model each OD flow sequence separately by Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), hoping to further reduce the prediction error. Besides, considering that modeling each OD flow sequence requires more prediction overhead, we explore a compromise prediction approach which predicts the overall TM as well as key OD flows.

We evaluate the performance of different prediction methods by the similarity between predicted and actual values, as well as the performance of different TE scenarios. Moreover,

978-1-7281-2999-0/19/$31.00 ©2019 IEEE

we also consider the prediction overhead of each prediction method. The evaluation results on Abilene, CERNET and GÉANT datasets demonstrate that prediction methods based on Recurrent Neural Networks (RNN) can achieve higher prediction accuracy than methods leveraging Convolutional Neural Networks (CNN) and Deep Belief Networks (DBN). Compared to predicting the overall traffic matrix directly, methods of predicting individual OD flows with RNN can significantly reduce the prediction loss and improve the performance of TE. However, modeling each OD flow sequence takes longer prediction time. In contrast, predicting the overall TM combined with large OD flows correction can achieve a trade-off between the TE results and the prediction overhead, which is a better choice for dynamic TE in the above three scenarios.

In summary, the main contributions of our work fall into three categories.

- We propose several traffic matrix prediction methods based on deep neural networks. Those proposed methods predict traffic matrix from three different perspectives: predict the overall traffic matrix directly, predict each OD flow separately and predict the overall traffic matrix combined with key element correction.
- We consider three different TE scenarios and introduce the effects of TE as another important evaluation indicator for different traffic matrix prediction methods.
- We evaluate different prediction methods based on three metrics: the prediction accuracy, the effects of different dynamic TE scenarios and the prediction time of each method, and test them with real world TMs from the Abilene, CERNET and GÉANT datasets.

The rest of this paper is organized as follows. Section II formalizes the TM prediction problem and presents the proposed methods. Section III describes the dynamic TE scenario based on TM prediction and introduces different TE algorithms used in this paper. Evaluation and results are presented in section IV and we survey related works of TM prediction in section V. Finally, we conclude our work in section VI.

II. TRAFFIC MATRIX PREDICTION METHODS

In this section, we first formalize the traffic matrix prediction problem. Then, we describe the proposed TM prediction methods.

A. Problem Statement

Let N be the number of router nodes in the network. The N-by-N traffic matrix is denoted by X and each entry of X is defined as x_{ij}, which represents the traffic demand from node i to node j. The measured traffic matrix sequence is denoted by $X_t(t \in [1,T])$, where T is the total time slots of measurement. Then, the TM prediction problem can be defined as predicting $\hat{X}_t$ via a series of historical data $(X_{t-k}, X_{t-k+1}, \ldots, X_{t-1})(k \in [1,t])$.

[9] and [13] have demonstrated that NN based TM prediction approaches perform better than traditional methods like Autoregressive Moving Average (ARMA), Holt-Winters (HW) and Principal Component Analysis (PCA). Therefore, in this paper, we mainly study TM prediction methods based on NN and predict TM from three different perspectives.

B. Predict the Overall TM Directly

The most straightforward way to predict TM is to build time series models based on the TM sequences. [9]–[12] have validated the performance of RNN on TM prediction. Since the normalized TM is a two-dimensional matrix, which can be regarded as a gray image, the TM prediction problem can be transformed into the time series prediction problem for gray image sequences. Here, we build two novel CNN based models for image sequence prediction.

a) LRCN: Long-term Recurrent Convolutional Networks (LRCN) is a novel image sequence prediction model used for the task of generating textual descriptions of images and videos [16]. LRCN uses CNN layers for image feature extraction and performs sequence prediction based on LSTM. In this method, we do min-max normalization for the TM sequences and treat the normalized TMs as gray images. Then, we use LRCN to model the image sequences. We build LRCN models containing three CNN layers and one LSTM layer with 100 hidden nodes.

b) TCN: Temporal Convolutional Networks (TCN) is a new innovation of the CNN architecture. It uses dilated causal convolution and residual connections to exhibit substantially longer memory and can outperform canonical recurrent architectures such as LSTMs and GRUs in several sequence modeling tasks [17]. In this method, we also regard the normalized TMs as gray images and use TCN to model the image sequences.

In addition, we reconstruct the LSTM framework proposed in [9] and GRU framework proposed in [11] as baselines.

C. Predict Each OD Flow Separately

Another possible way to predict TM is to predict each OD flow separately. This approach assumes each OD flow is independent and may get better prediction accuracy on large OD flows. However, modeling each OD flow sequence separately may ignore the inherent correlations between OD flows and requires longer training and predicting time. [13] and [14] have proposed a DBN based method for OD flow prediction, but DBN cannot well track the long-term dependencies of TM sequences. In this paper, we try to further improve the prediction accuracy by predicting each OD flow sequence based on RNN, since the RNN architectures are more appropriate for time series prediction tasks. We propose two methods for OD-flow based prediction.

a) LSTM-OD: In this method, we build an LSTM network for each OD flow sequence. Each LSTM network contains 1 hidden layer with 50 hidden nodes.

b) GRU-OD: Similar to LSTM-OD, we establish a GRU model for each OD flow and predict based on the historical data. The GRU network also contains 1 hidden layer with 50 hidden nodes.

Besides, we also build the DBN networks proposed in [13] as a comparative experiment.

D. Predict the Overall TM Combined with Key Element Correction

The large OD flows can affect the network performance more significantly than small ones [18]. Therefore, it is necessary to reduce the prediction error of large OD flows. Predicting each OD flow separately can get higher prediction accuracy on each OD flow, but it requires longer prediction time. In order to make a trade-off between the prediction accuracy and the prediction overhead, we propose two methods focusing on predicting the overall traffic matrix as well as the large elements.

a) LSTM-KEC: In this method, we first build an LSTM network based on the TM sequences and predict the overall TM directly. Then, we select the top M OD flows with maximum average traffic volume as key elements and predict each one separately based on LSTM models. Finally, we use the prediction results of key elements to correct the predicted TM.

b) GRU-KEC: Similar to LSTM-KEC, we establish a GRU network for direct traffic matrix prediction. Then, we use GRU models to predict the selected M key elements and correct the predicted TM.

Simply, we set M as 20 for the above two methods. The impact of different M values is left as our future work.

III. DYNAMIC TE BASED ON TM PREDICTION

In this section, we will first introduce the dynamic TE scenario based on TM prediction, and then describe the TE algorithms used in this paper.

A. Problem Statement

The purpose of TM prediction in TE is to set up routing strategy for the future traffic demands in advance to avoid congestion. We follow the assumptions of [8] in the TE scenario. Time is divided into consecutive intervals. TM is assumed to be fixed throughout a single interval and it can be measured at the end of the interval. At the beginning of each interval t, the measured TM of interval $t-1$ will be added to the historical TM sequence. Then, the TM prediction method will predict the TM of interval t based on the measured TM sequence. The TE algorithm will use the predicted TM as input and calculate the routing strategy of interval t in advance. Afterwards, the actual traffic of interval t will be routed based on the calculated strategy and the maximum link utilization (MLU) will be obtained as the result of TE. In order to obtain the routing strategy at the beginning of each interval, the prediction of TM should be fast enough. Note that we do not address on the problem of TM measurement in this paper.

B. Traffic Engineering Algorithms

We consider three different TE algorithms in this paper.

a) Optimizing OSPF weights: [2] proves that optimizing the setting of link weights in OSPF networks is NP-hard and first proposes a local search heuristic algorithm to get the approximate optimal solution. The main idea of this algorithm is to optimize the OSPF weights so that the out traffic of a router node can be equally split on multiple next hops as much as possible.

b) SDN/OSPF hybrid network traffic engineering: [4] proposes a novel heuristic algorithm called SOTE to achieve traffic engineering objectives in an incrementally deployed SDN network. SOTE will first optimize the OSPF weight of each link by local search and then adjust the flow-splitting ratio of the SDN nodes to get lower MLU. We set the deployment rate of SDN nodes as 30%, which is an appropriate ratio as shown in [4].

c) The multi-commodity flow problem: We take the solution of the MCF problem as the theoretical optimal solution of TE. Assume that we are given a directed network graph $G = (V, E)$ (V is the vertices set, E is the arcs set). $c(e)(e \in E)$ denotes the capacity of link e. There are L traffic demands in the network and each traffic demand $D_i(i \in L)$ has a source node denoted by $s(i)$ and a target node denoted by $t(i)$. f_e^i represents the flow that demand Di is split on link e. $In(v)(v \in V)$ denotes the link sets on which there are traffic flowing into node v, and $Out(v)$ denotes the link sets on which there are traffic flowing out from node v. U represents the MLU of the network. We can formulate the MCF problem as following.

$$\textbf{Minimize} \quad U \tag{1}$$

$$\sum_{e \in Out(v)} f_e^i - \sum_{e \in In(v)} f_e^i = \begin{cases} D_i & \text{if } v = s(i) \\ -D_i & \text{if } v = t(i) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \{1, \dots, L\}, v \in V \tag{2}$$

$$\sum_{i=1}^{L} f_e^i \le U \cdot c(e) \quad \forall e \in E \tag{3}$$

$$0 \le f_e^i \le D_i \quad \forall i \in \{1, \dots, L\}, e \in E \tag{4}$$

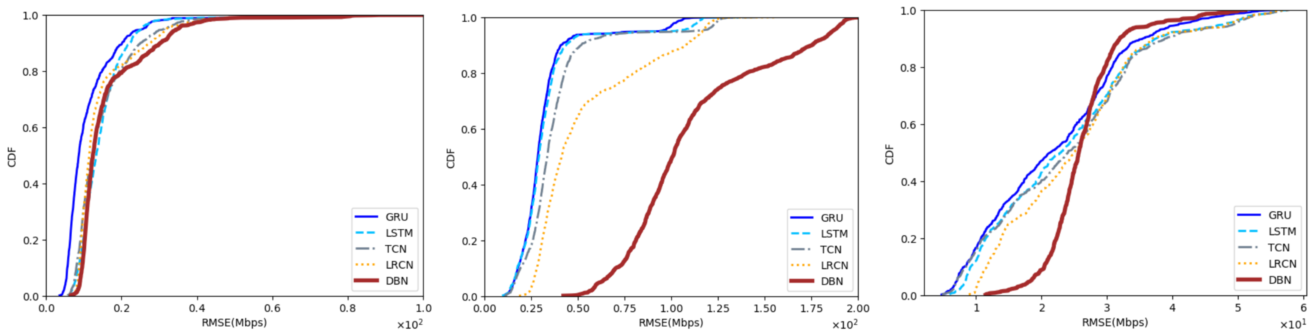
Eq. (2) denotes the flow conservation on transit nodes. Eq. (3) denotes that the total flows on one link cannot exceed its capacity. Eq. (4) denotes that flows on one link must be non-negative and cannot exceed the demands.

IV. EXPERIMENTS AND EVALUTION

In this section, we first introduce the datasets used in this paper. Then, we describe the performance metrics and conduct the simulation experiments.

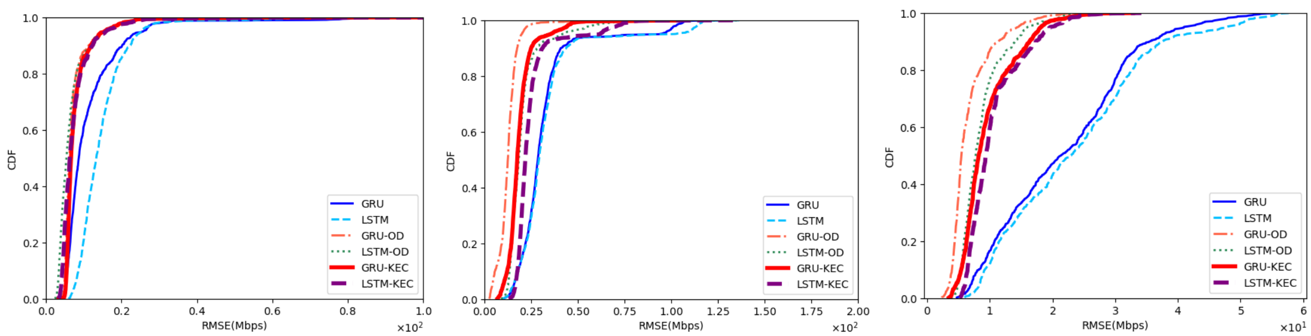
A. Datasets

We evaluate our prediction methods by real world TMs from the American Research and Education Network (Abilene) [19], the China Education and Research Network (CERNET) and the Europe Research and Education Network (GÉANT) [20]. The Abilene network is made up of 12 peer nodes and 30 undirected links (as of 2004). The CERNET network consists of 14 router nodes and 32 undirected links (as of 2013), and the GÉANT network contains 23 router nodes and 74 undirected links (as of 2005). We record 5 weeks of consecutive data from each dataset. Then, we build slide windows on the data sequences for continuous feeding and learning. For every W consecutive data, the first $W-1$ will be used as the input and the last one will be regard as the output, as proposed in [13].



(a) Abilene (b) CERNET (c) GÉANT

Fig. 1: The RMSE result of different NN models.



(a) Abilene (b) CERNET (c) GÉANT

Fig. 2: The RMSE result of different prediction perspectives.

B. Performance metrics

In order to estimate the prediction accuracy of different prediction methods, we adopt the Root Mean Squared Error (RMSE) as a performance metric. The RMSE between predicted traffic matrix $X^{'}$ and actual traffic matrix X can be represented as:

$$RMSE = \sqrt{\sum_{i=0}^{N}\sum_{j=0}^{N}(x'_{ij} - x_{ij})^2} \tag{5}$$

where N denotes the node number of the network.

In addition, we take the performance of different dynamic TE scenarios as another important evaluation indicator. Under different TE scenarios, each TM prediction method will predict the TM at the beginning of each time interval. Then, the TE algorithm will calculate the routing strategy of this interval based on the predicted TM in advance. The actual traffic of this time interval will be routed through the calculated strategy and the MLU will be recorded as U'. We also use the actual TM of each interval as the input of TE and record the MLU result as U. Afterwards, we compute the bias between U' and U, which can reflect the proximity of dynamic TE performance to actual TE performance. The bias can be defined as:

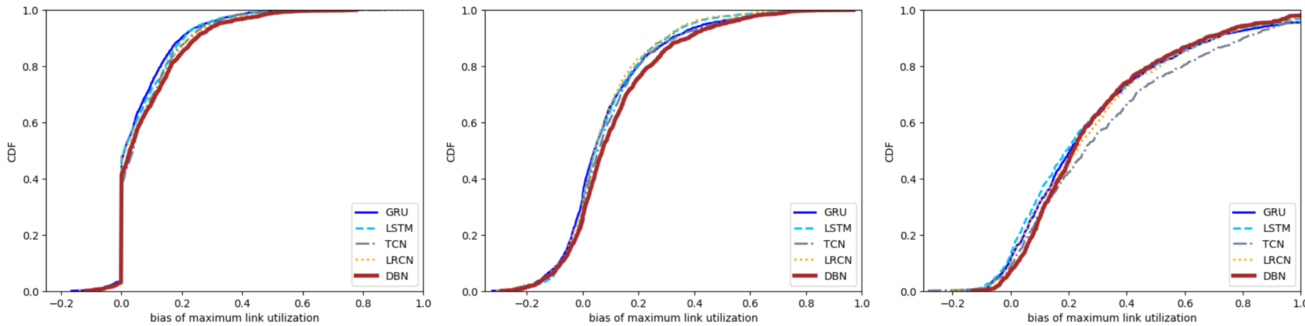
$$bias = \frac{U' - U}{U} \tag{6}$$

Moreover, in order to obtain the routing strategy at the beginning of each time interval, TM prediction should be completed in a short time. Therefore, we also consider the prediction time of each TM prediction method. The experiments are done on a computer with an Intel i7-6700 CPU, an Nvidia GeForce GTX 1060 GPU and 16GB memory.

C. Results and Analysis

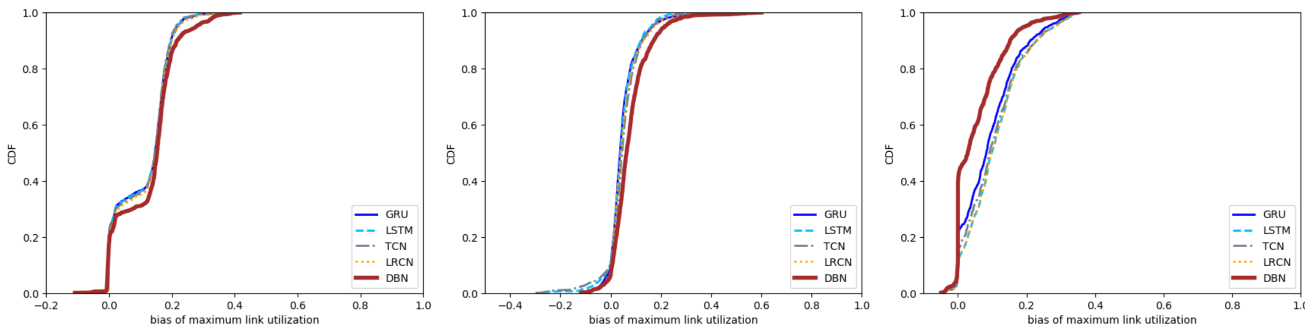
We first plot the Cumulative Distribution Function (CDF) curves for the RMSE result of different NN models. As shown in Fig. 1, RNN based methods can obtain better prediction accuracy than LRCN and TCN in all of the three datasets. Although DBN based method predicts each OD flow separately, it cannot achieve higher accuracy, which indicates that DBN cannot well explore the inherent correlations among the TM sequences. We also compare the accuracy of different TM prediction perspectives, as illustrated in Fig. 2. Compared to GRU and LSTM, GRU-OD and LSTM-OD can significantly improve the prediction accuracy. For example, in the GÉANT topology, the RMSE result based on GRU-OD is always less than 25Mbps. However, when predicting through GRU, only 60% of the RMSE result is less than 25Mbps. GRU-KEC and LSTM-KEC can achieve similar prediction accuracy as GRU-OD and LSTM-OD, which means that predicting the overall traffic matrix directly performs poorly on large OD flows.

Fig. 3 - Fig. 5 illustrate the MLU bias of different prediction methods under OSPF, SOTE and MCF scenarios. As we can see, the differences between GRU, LSTM, LRCN and TCN are not very significant, although their prediction accuracy can vary greatly. The result of DBN is worse than the above four models in the Abilene and CERNET datasets. However, in the GÉANT topology, DBN can well reduce the MLU bias and provide better TE performance, despite of its poor prediction accuracy. Under the OSPF scenario and the SOTE scenario,



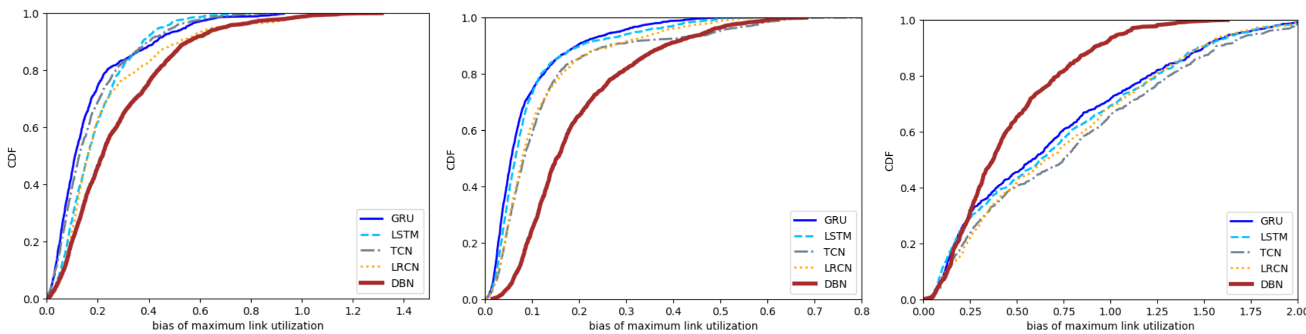
(a) Abilene (b) CERNET (c) GÉANT

Fig. 3: The MLU bias of different NN models in OSPF scenario.



(a) Abilene (b) CERNET (c) GÉANT

Fig. 4: The MLU bias of different NN models in SOTE scenario.



(a) Abilene (b) CERNET (c) GÉANT

Fig. 5: The MLU bias of different NN models in MCF scenario.

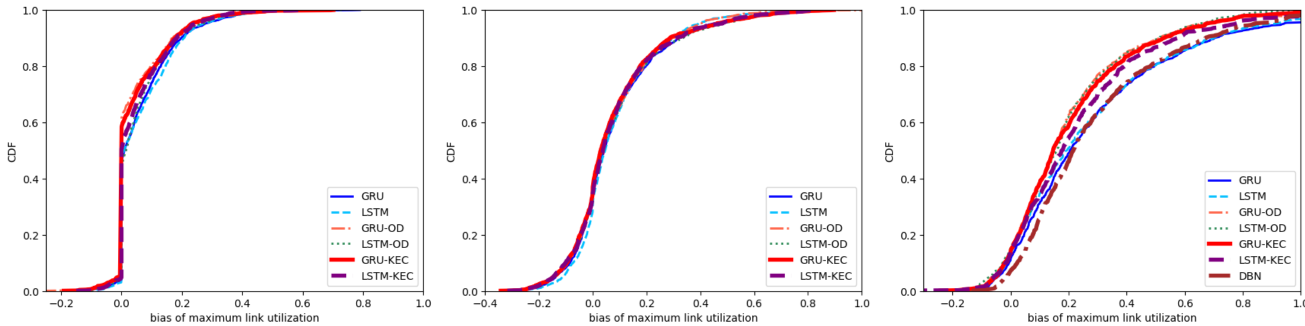
the bias of different prediction methods can even be negative. This is because those two algorithms are heuristic and cannot guarantee that the TE result based on the actual TM is optimal.

Fig. 6 - Fig. 8 compare the MLU bias of different prediction perspectives under the three TE scenarios. Since DBN can provide better result than GRU and LSTM in the GÉANT topology, we also plot the MLU bias of DBN in the GÉANT experiments. As shown in the three figures, methods of predicting individual OD flows with RNN can further improve the effects of TE, especially in the MCF scenario. For example, in Fig. 8c, 90% of the bias based on GRU-OD is less than 0.3, while only 50% of the bias leveraging DBN is less than 0.3. GRU-KEC and LSTM-KEC can provide better performance than GRU and LSTM, and have little difference compared with GRU-OD and LSTM-OD. This indicates that the large OD flows in TM have greater impacts on network link utilization.

In addition, we also evaluate the prediction overhead of different prediction methods in the three network topologies. As shown in Table II, methods which predict each OD flow separately require longer prediction time. This problem can be alleviated by parallelization, but parallelization requires more calculation overhead. GRU and LSTM can predict TM quickly, while methods based on CNN architecture take more time. GRU-KEC and LSTM-KEC need to predict the overall traffic matrix as well as the selected OD flows, hence their prediction time is a compromise.

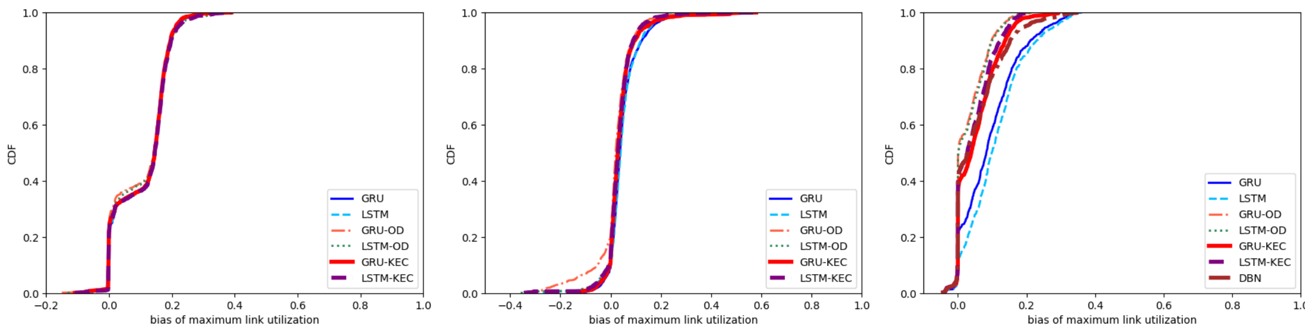
V. RELATED WORK

Various methods have been proposed for TM prediction. [5] adopts statistical models such as Gaussian distribution and Poisson distribution for TM prediction. [7] proposes three TM prediction methods: Independent Node Prediction (INP), Total Matrix Prediction with Key Element Correction (TMP-KEC) and Principle Component Prediction with Fluctuation Com-



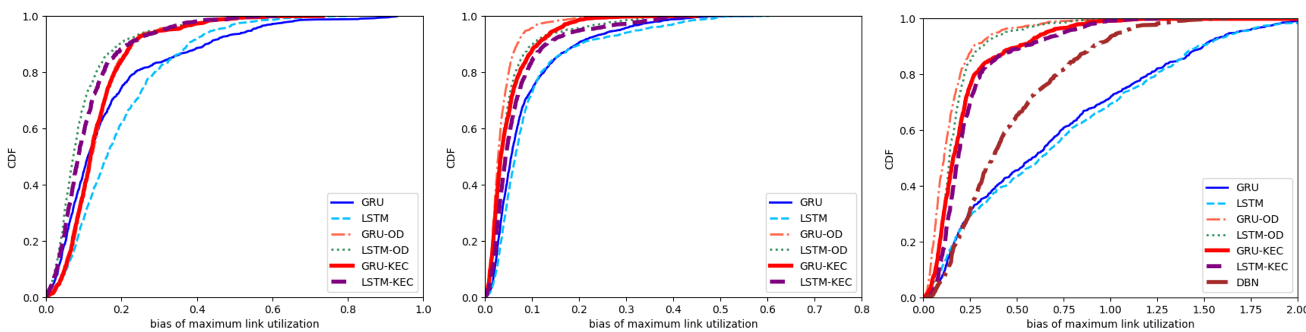
(a) Abilene (b) CERNET (c) GÉANT

Fig. 6: The MLU bias of different prediction perspectives in OSPF scenario.



(a) Abilene (b) CERNET (c) GÉANT

Fig. 7: The MLU bias of different prediction perspectives in SOTE scenario.



(a) Abilene (b) CERNET (c) GÉANT

Fig. 8: The MLU bias of different prediction perspectives in MCF scenario.

TABLE I: THE PREDICTION TIME OF DIFFERENT PREDICTION METHODS

Topologies	GRU	LSTM	LRCN	TCN	GRU-KEC	LSTM-KEC	DBN	GRU-OD	LSTM-OD
Abilene	1.8590ms	2.0731ms	3.3602ms	25.308ms	28.307ms	32.620ms	99.032ms	175.68ms	236.89ms
CERNET	2.0004ms	2.1793ms	4.0112ms	26.537ms	33.237ms	36.943ms	141.31ms	305.72ms	320.41ms
GÉANT	1.8948ms	2.0020ms	3.9130ms	25.192ms	37.430ms	40.696ms	456.43ms	867.51ms	952.48ms

ponent Correction (PCP-FCC). INP uses exponential fitting to predict node traffic and then apportions it to different OD flows. TMP-KEC predicts the total network traffic as well as the large node traffic, and obtains TM through the proportional coefficient of each OD flow. PCP-FCC predicts the long term and short term components of network traffic separately through PCA and ARIMA to improve the prediction accuracy. [15] filters network traffic into long-term and short-term variations. It predicts the long-term variation through ARIMA and SARIMA and estimates the short-term component instead of predicting it. In addition, [15] evaluates different prediction methods through the result of MCF problem rather than the prediction accuracy. However, due to the nonlinear nature of network traffic [6], those statistical and linear models are insufficient for TM prediction.

Deep learning can well solve nonlinear problems and has been widely used for TM prediction. [21] compares NN based models with traditional time series models and proves that NN can better extract the nonlinear nature of network traffic volume than ARMA and HW. [8] uses fully-connected networks, CNN and nonlinear auto-regressive model to predict TM and finds that the prediction loss is acceptable only when there are temporal correlations between TMs in the sequence.

[9]–[12] establish time series models for TM sequences via RNN and predict TM directly, which may perform poorly on large OD flows. [9] presents a LSTM based framework for TM prediction. The experiment results demonstrate that LSTM models converge quickly and can give state of the art TM prediction performance compared with traditional models such as ARMA and HW. [10] also predicts TM based on LSTM architecture and proves that LSTM can outperform PCA [5], TomoGravity [22] and back-propagation neural networks. [11] investigates another type of RNN, the GRU networks, for TM prediction and then dynamically allocates the network resources based on the prediction results. [12] validates the strong performance of standard RNN, LSTM and GRU on TM prediction. [13] and [14] use DBN to predict each OD flow sequence in large-scale IP backbone networks and data center networks. Compared to PCA and TomoGravity, DBN based method can better reduce the prediction error of each OD flow. However, DBN cannot well track the long-term dependencies of network traffic.

VI. CONCLUSION

Traffic matrix is critical for many network management tasks, especially for traffic engineering. In this paper, we focus on predicting TM for different dynamic TE scenarios. We propose several TM prediction methods based on deep neural networks, and predict TM from three different perspectives. Then, we evaluate the proposed methods through the prediction accuracy, the performance of different dynamic TE scenarios and the prediction overhead. Evaluation results on Abilene, CERNET and GÉANT datasets illustrate that RNN based models can outperform LRCN, TCN and DBN on the prediction accuracy. Predicting each OD flow separately through RNN can further improve both the prediction accuracy and the performance of TE under the OSPF network scenario, the SDN/OSPF hybrid network scenario and the MCF problem scenario, but it requires longer prediction time. In contrast, predicting the overall traffic matrix combined with key element correction can achieve a trade-off between the TE performance and the prediction time, which is more appropriate for dynamic TE in the above three scenarios.

In future work, we will study the impact of different large OD flows in TM prediction. We will also refer to oblivious routing [23] and try to guarantee the effects of TE when the prediction of TM is inaccurate.

ACKNOWLEDGMENT

This work is partially supported by the National Key Research and Development Program of China under Grant No. 2017YFB0803004.

REFERENCES

[1] A. Svigelj, R. Sernec, and K. Alic, "Network traffic modeling for load prediction: a user-centric approach," *IEEE Network*, vol. 29, no. 4, pp. 88–96, 2015.

[2] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, vol. 2. IEEE, 2000, pp. 519–528.

[3] S. Agarwal, M. Kodialam, and T. Lakshman, "Traffic engineering in software defined networks," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2211–2219.

[4] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in sdn/ospf hybrid network," in *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, 2014, pp. 563–568.

[5] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, "Traffic matrices: balancing measurements, inference and modeling," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1. ACM, 2005, pp. 362–373.

[6] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," in *ACM SIGCOMM computer communication review*, vol. 23, no. 4. ACM, 1993, pp. 183–193.

[7] W. Liu, A. Hong, L. Ou, W. Ding, and G. Zhang, "Prediction and correction of traffic matrix in an ip backbone network," in *2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2014, pp. 1–9.

[8] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, "Learning to route," in *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*. ACM, 2017, pp. 185–191.

[9] A. Azzouni and G. Pujolle, "Neutm: A neural network-based framework for traffic matrix prediction in sdn," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–5.

[10] J. Zhao, H. Qu, J. Zhao, and D. Jiang, "Towards traffic matrix prediction with lstm recurrent neural networks," *Electronics Letters*, vol. 54, no. 9, pp. 566–568, 2018.

[11] S. Troia, R. Alvizu, Y. Zhou, G. Maier, and A. Pattavina, "Deep learning-based traffic prediction for network optimization," in *2018 20th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2018, pp. 1–4.

[12] N. Ramakrishnan and T. Soni, "Network traffic prediction using recurrent neural networks," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 187–193.

[13] L. Nie, D. Jiang, L. Guo, and S. Yu, "Traffic matrix prediction and estimation based on deep learning in large-scale ip backbone networks," *Journal of Network and Computer Applications*, vol. 76, pp. 16–22, 2016.

[14] L. Nie, D. Jiang, L. Guo, S. Yu, and H. Song, "Traffic matrix prediction and estimation based on deep learning for data center networks," in *2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2016, pp. 1–6.

[15] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, and K. Shiomoto, "Traffic prediction for dynamic traffic engineering," *Computer Networks*, vol. 85, pp. 36–50, 2015.

[16] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[17] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[18] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft, "How to identify and estimate the largest traffic matrix elements in a dynamic environment," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1. ACM, 2004, pp. 73–84.

[19] Y. Zhang, "Abilene dataset," *URL http://www. cs. utexas. edu/yzhang/research/AbileneTM*, 2011.

[20] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.

[21] M. Barabas, G. Boanea, A. B. Rus, V. Dobrota, and J. Domingo-Pascual, "Evaluation of network traffic prediction based on neural networks with multi-task learning and multiresolution decomposition," in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 95–102.

[22] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," *Proceedings of the Acm Sigmetrics*, vol. 31, no. 1, pp. 206–217, 2003.

[23] N. Bansal, A. Blum, S. Chawla, and A. Meyerson, "Online oblivious routing," in *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*. ACM, 2003, pp. 44–49.