Predicting Traffic Demand Matrix by Considering Inter-flow Correlations

Kaihui Gao*, Dan Li*, Li Chen†, Jinkun Geng*, Fei Gui‡, Yang Cheng*, Yue Gu*
*Tsinghua University, †Tencent, ‡University of XiangTan
Email: gkh18@mails.tsinghua.edu.cn, tolidan@tsinghua.edu.cn

***Abstract*—Accurate traffic demand matrix (TM) prediction is essential for effective traffic engineering and network management. Based on our analysis of real traffic traces from Wide Area Network (WAN), the traffic flows in TM are both time-varying (i.e. with intra-flow dependencies) and correlated with each other (i.e. with inter-flow correlations). However, existing works in TM prediction ignore inter-flow correlations.**

In this work, we propose a new model for TM prediction by considering both inter-flow correlation and intra-flow dependencies, namely *CRNN model*. By observing that most strongly-correlated traffic flows have the same source or destination, we employ convolutional structures to capture the neighboring correlation in TMs and extract even higher-order correlations. To model the trends in time dimension, like previous works we use recurrent structures to capture the temporal variations of traffic flows. Our evaluation on two WAN datasets shows that, when predicting the next TM, CRNN model reduces the Mean Squared Error (MSE) by up to 44.8% compared to state-of-the-art method; and the gap is even larger when predicting the next multiple TMs.

***Index Terms*—Traffic Demand Matrix Prediction, Convolutional Neural Network, Recurrent Neural Network**

I. Introduction

Traffic demand matrix (TM) is a snapshot of traffic volumes between all pairs of source and destination node in a computer network at a given time. With the recent development of high-speed traffic measurement technologies [1], [2], complete TM could be collected from operational Wide Area Network (WAN). In this paper, we are concerned with the problem of TM prediction, which uses past and current network TMs to estimate future TMs. Accurate TM prediction plays an important role in the areas of traffic engineering (TE) [3], capacity planning [4], resource allocation [5], congestion mitigation [6], and failure/anomaly detection [7] in computer networks.

We model TM prediction as a multi-dimensional time-series prediction problem. In the past, autoregressive Integrated Moving Average (ARIMA) [8] and Support Vector Regression (SVR) [9] are two most-widely used classical methods. However, ARIMA cannot model the nonlinear patterns due to its linear structure. While SVR can model nonlinear patterns, laborious human effort is needed to tune its parameters for accurate predictions. In recent years, Recurrent Neural Network (RNN) [10] based approaches have achieved state-of-the-art results in many time-series prediction tasks [11]. RNN have cyclic connections over time, so it has been widely used to process sequence data in the modeling of nonlinear sequential patterns [12]. In particular, Long and Short Time Memory (LSTM) [13], and its variant, Gated Recurrent Unit (GRU) [14], are powerful RNN algorithms to solve the gradient exploding and vanishing problem in RNN [15]. Both LSTM and GRU have been used to model intra-flow dependencies for TM prediction in recent works [16], [17].

However, the accuracy of TM prediction in prior works is fundamentally limited, as they ignore the importance of correlations among network traffic flows. Our insight is that inter-flow correlations contain implicit knowledge of the traffic demands. For instance, traffic flows towards the same destination may have positive correlations due to a common popular service in the destination; traffic flows from the same source or towards the same destination may also have negative correlations due to bandwidth contention; etc. This insight is confirmed by our analysis on traces from real WAN: we observe high correlations among certain flows in the TMs, particularly flows with the same source or destination node. These observations show that the inter-flow correlations are not negligible in TM prediction.

Motivated by the observations, in this paper we propose a more comprehensive model for TM prediction, by considering both inter-flow correlations and intra-flow dependencies. For inter-flow correlation, since traffic flows with the highest correlations usually share the same source or destination and these flows are thus neighboring in TMs, we use CNN (Convolutional Neural Networks) to capture the local correlations in the TMs and extract the even higher-order correlations between adjacent and distant flows. For intra-flow dependencies, like previous works [16], [17] we use RNN to capture the temporal variations of traffic flows. We call the new approach *CRNN model*, and design a CRNN-based end-to-end algorithm for TM prediction.

We evaluate our approach on two real WAN traffic datasets (i.e. Abilene network [18] and GÉANT network [19]). Our results show that, when predicting the next single TM, CRNN reduces the MSE by 30.7% compared to state-of-the-art method on Abilene network, and the reduction of MSE is 44.8% on GÉANT network. When predicting the next multiple TMs, the gap will be even larger. Taking GÉANT network as the example, the MSE of CRNN model will be reduced by 65.9% compared to state-of-the-art method.

To summarize, the main contributions of this paper are:

1) We find high inter-flow correlations in the TMs, by analyzing public traffic traces of real networks. In particular, most strongly-correlated flows share the same source or

destination, i.e., located in the same row or column in the TMs.

2) We propose CRNN model for TM prediction, which uses CNN to capture the inter-flow correlations and RNN (LSTM) to capture the intra-flow dependencies.

3) We conduct extensive experiments to evaluate CRNN, based on two real-world WAN traffic datasets. Our evaluation results demonstrate the effectiveness of CRNN model in improving the TM prediction accuracy.

II. Related Work

A. TM Prediction

The task of TM prediction takes a series of historical TMs as the input and forecasts the next TM(s). In recent years, various time-series analytic methods have been proposed for this topic [20]. As the representative works, Azzouni et al. [16], Zhuo et al. [21] and Troia et al. [17] use LSTM or GRU to capture intra-flow dependencies for TM prediction, and achieve satisfactory prediction accuracy. However, inter-flow correlations among flows are not considered in these works to further reduce the prediction error.

B. Link Load Prediction

Link load refers to the aggregated traffic of flows crossing a certain link. It is worth noting that, we cannot directly infer the link loads from a TM, because link loads are also affected by the network topology and routing strategies. Besides, it is also difficult to infer the TM based on the link loads [22].

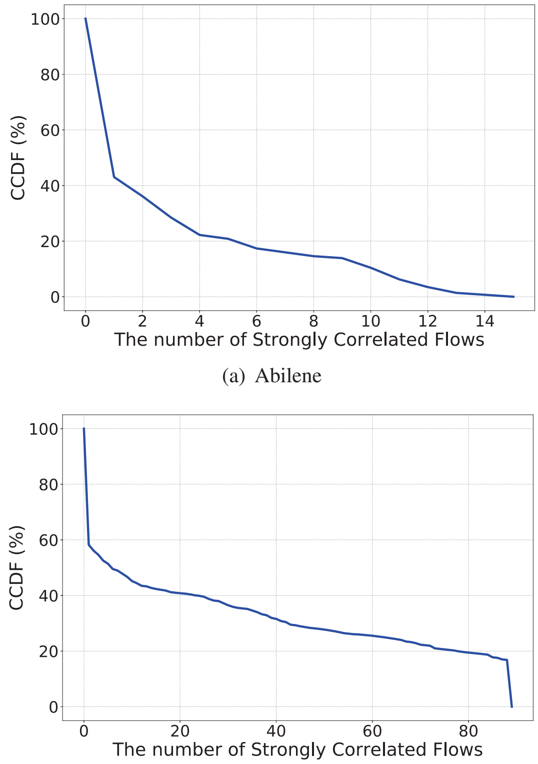
The works of link load prediction treats the traffic load on a link as time-series data, and predicts its future value. Barabas et al. [23] evaluate and compare traditional linear prediction models (ARMA, ARAR, HW) and NNs (neural networks)-based model for link load prediction. Their results show that NNs-based model outperforms traditional linear methods in terms of prediction accuracy.

Recently, there is growing interest in the literature to apply graph-based learning methods [24] to link load prediction. The key insight of these techniques relies on their capability to learn a representation of each node or link of the graph, considering both its properties (e.g., features) and the network structure (e.g., the topology). For instance, Andreoletti et al. [25] employ a graph-based Diffusion Convolutional Recurrent Neural Network (DCRNN) to forecast traffic loads of links in WAN.

C. Node Traffic Prediction

Node traffic refers to the aggregated traffic of flows crossing a node (e.g. a server or a router). Fro instance, in the well-known Kaggle contest [26] which focuses on forecasting the future traffic for a web server, the contestants use advanced Machine Learning algorithms, such as attention based seq2seq model, to improve the prediction accuracy.

Geo-spatial correlations are also exploited for node traffic prediction. For Cellular Networks, Wang et al. [27] present a hybrid deep learning model for spatiotemporal prediction. They apply auto-encoder to capture both local and global spatial dependencies of traffic between adjacent cell towers. Feng et al. [28] apply the seq2seq model with attention mechanism to build the sequential model, and capture the spatial correlations among base stations to model the influence of external information on traffic prediction. However, the geo-spatial correlations considered in this type of works are different from the inter-flow correlations we investigate in this work, and their approach cannot be directly employed for TM prediction.



(a) Abilene

(b) GÉANT

Fig. 1: CCDF of the number of strongly correlated flows ($\rho \geq 0.6$) for every flow.

III. Analysis of Inter-flow Correlations

In this section, we present our analytical results to demonstrate the critical correlations among flows in WAN, which have been overlooked by previous works. To this end, we first describe the terminology, datasets and our analytic method. Then, we perform the analysis and interpret the results by investigating the characteristics of strongly correlated flows. Finally, we derive guidelines for accurate TM prediction based on our observation.

A. Terminology

TM: A TM represents the traffic volumes collected over a certain time interval between all source-destination pairs in a network. To be specific, $\mathbf{TM}_t$ is a two-dimensional array representing the TM for the t-th time interval:

$$\mathbf{TM}_t = \begin{bmatrix} T^t_{1,1} & T^t_{1,2} & \cdots & T^t_{1,n} \\ T^t_{2,1} & T^t_{2,2} & \cdots & T^t_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ T^t_{n,1} & T^t_{n,2} & \cdots & T^t_{n,n} \end{bmatrix} \tag{1}$$

Flow: A flow (e.g. $\mathbf{T}_{s,d}$) is a one-dimensional array containing traffic volumes for multiple continuous time intervals from source s to destination d.

B. Datasets

Our analysis is based on two publicly available network traffic traces. The Abilene [18] topology has 12 nodes and 15 links, and the TMs of Abilene network are summarized every 5 minutes starting from Mar. 1st, 2004 for 24 weeks (48383 TMs in total). The GÉANT [19] topology has 23 nodes and 36 links, and the TMs of GÉANT network are summarized every 15 minutes starting from Jan. 8th 2005 for 16 weeks (10772 TMs in total).

C. Analysis & Observations

Pearson Correlation Coefficient (ρ) is a well-known indicator of measuring the correlation [29]. We reveal the correlation of flows by calculating ρ for each pair of $\mathbf{T}_{s,d}$ and $\mathbf{T}_{s',d'}$:

$$\rho = \frac{\operatorname{cov}\left(\mathbf{T}_{s,d}, \mathbf{T}_{s',d'}\right)}{\sigma_{\mathbf{T}_{s,d}} \sigma_{\mathbf{T}_{s',d'}}}, \tag{2}$$

where $\operatorname{cov}(\cdot)$ is the covariance operator, and $\sigma_{\mathbf{T}_{s,d}}$ is the standard deviation of $\mathbf{T}_{s,d}$. ρ varies between -1 and $+1$: the larger its absolute value is, the stronger the correlation between the two variables is. We only consider its absolute value in the analysis below.

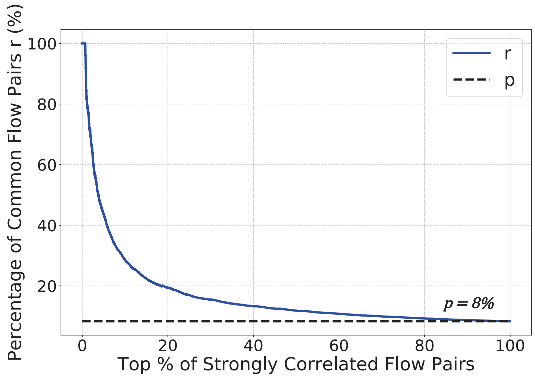
Observation 1: There exist abundant high correlations between flows in real WAN.

We calculate ρ between all pairs of flows. For each flow, we count the number of strongly correlated flows (i.e. $\rho \geq 0.6$). Then we depict the Complementary Cumulative Distribution Function (CCDF) of correlated flow number (for each flow) as Figure 1 for Abilene and GÉANT. We observe that, $\sim 58.2\%$ ($\sim 43.1\%$) flows in GÉANT (Abilene) have at least 1 strongly correlated flows. And $\sim 40.8\%$ ($\sim 20.8\%$) flows in GÉANT (Abilene) have at least 20 (5) strongly correlated flows. This shows that there exist abundant high correlations between flows in real WAN.

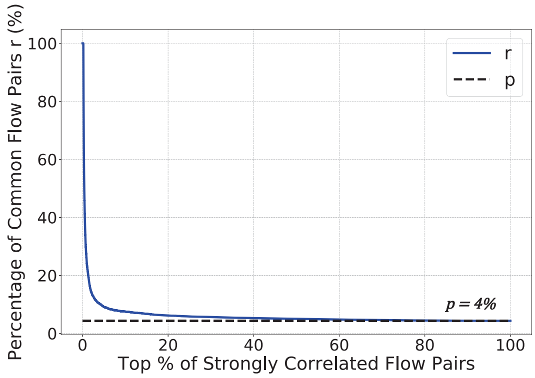
Observation 2: Most strongly-correlated flow pairs share the same source or destination.

For ease of expression, if a flow pair shares common source or destination, we call it a *common flow pair*. We calculate the correlation coefficient ρ of every flow pair, and sort all the flow pairs in descending order by the value of ρ. Then for the top $x\%$ strongly-correlated flow pairs, we count the percentage of common flow pairs out of them, denoted by r. Figure 2 shows the result.

We use p to represent the percentage of common flow pairs out of all the flow pairs, which is 8% in Abilene network and 4% in GÉANT network. It is shown as the baseline in Figure 2. It is worth noting that, if the common flow pairs do not have any correlation with ρ, the value of r should be always around the baseline value of p. However, from the figure we find that the curve of r is always above the baseline. Specifically, among the top 0.1% strongly-correlated flow pairs, all of them are common flow pairs in both Abilene and GÉANT networks.



(a) Abilene



(b) GÉANT

Fig. 2: The proportion of flow pairs with the same source or destination in top $x\%$ strongly-correlated flow pairs.

Among the top 1% strongly-correlated flow pairs, $\sim 80\%$ of them are common flow pairs in Abilene network, while $\sim 25\%$ of them are common flow paris in GÉANT network. When x is smaller, r is much larger, which indicates that common flow pairs have much higher probability to be strongly correlated, compared to other flow pairs.

D. Motivation of This Work

With the above analysis, we demonstrate that there exist abundant correlations among traffic flows, especially flows with the same source or destination. Such correlations are plausible in practice: traffic flow volume can be affected by multiple factors. For instance, traffic flows towards the same destination d may have positive correlations due to a common popular service in d. On the other hand, traffic flows from the same source s or towards the same destination d may also have negative correlations due to bandwidth contention.

Therefore, in this paper we take inter-flow correlations into account when predicting TMs, together with intra-flow dependencies considered by prior works.

IV. Design of CRNN Model

In order to capture both inter-flow correlations and intra-flow dependencies for TM prediction, in this section we design our CRNN algorithm, which combines both convolutional and recurrent structures in the model.

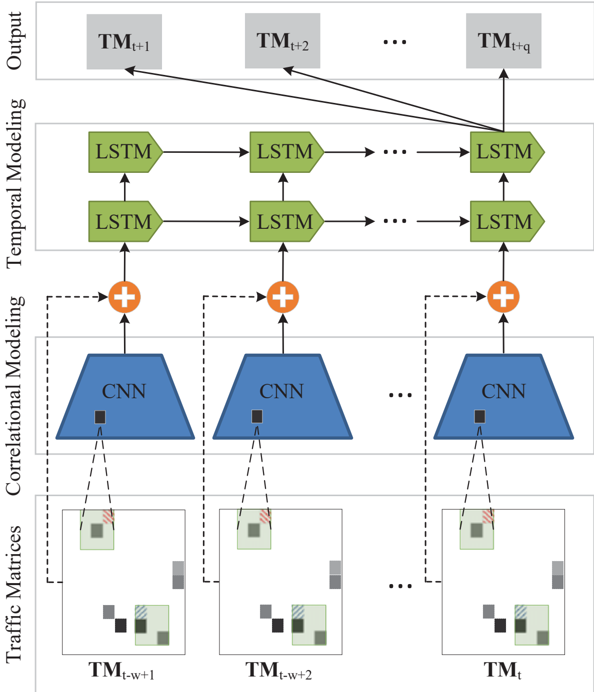


Fig. 3: The proposed Convolutional Recurrent Neural Network (CRNN) model.

A. *Overview*

Figure 3 presents the overall training framework of the CRNN model. It consists of two modules, namely, *Correlational Modeling* and *Temporal Modeling*. Firstly, TMs of the past w time intervals are taken as the input. In *Correlational Modeling* module, for each time interval t' $(t-w < t' \leq t)$, we use convolutional kernels and multiple convolutional layers to extract the inter-flow correlations from each $\mathbf{TM}_{t'}$. Then, we concatenate the raw $\mathbf{TM}_{t'}$ and its inter-flow correlational features into a single vector, and take it as the input of *Temporal Modeling* module. In *Temporal Modeling* module, we employ LSTM to learn intra-flow dependencies for each flow. Finally, based on the learned inter-flow correlations and intra-flow dependencies, CRNN predicts the next q TMs (i.e. $\mathbf{TM}_{t+i}, i = 1, 2, \cdots, q$). In the back-propagation phase, the prediction error used to update the model parameters.

B. *Learning Inter-flow Correlations*

Figure 4 presents the process of extracting correlational features in a TM with CNN. The CNN model first learns the local high correlations as the lower-order features, and abstracts these features together to the higher-order (as shown in subfigures a and b). With more convolutional layers, the global correlations will be extracted, and higher-order features can be learned (as shown in subfigures c and d).

In time interval t, the input of CNN is an one-channel matrix $\mathbf{TM}_t$. The features of inter-flow correlations are extracted by multiple convolutional filters and convolutional layers. We denote the k-th filter output of the l-th layer as $\mathbf{o}_l^k$, and the j-th filter output of the previous layer as $\mathbf{o}_{l-1}^j$. Then, $\mathbf{o}_l^k$ can be calculated as follows:

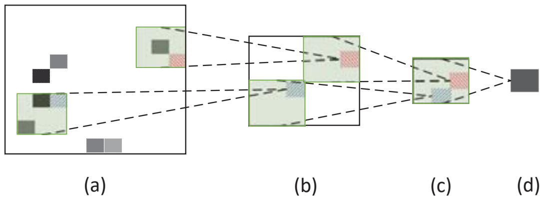


Fig. 4: The process of convolution for capturing higher-order inter-flow correlations.

$$\mathbf{o}_l^k = \sigma\left(\sum_{j=1}^{C_{l-1}}\left(\mathbf{W}_l^{jk} * \mathbf{o}_{l-1}^j + \mathbf{b}_l^k\right)\right) \tag{3}$$

where $\mathbf{W}_l^{jk}$ and $\mathbf{b}_l^k$ are the weight and the bias, respectively, $*$ is the convolutional operation, C_{l-1} is the number of convolutional filters in $(l-1)$-th layer, and σ represents the ReLU activation function.

After convolution, pooling layers are used to resample and aggregate the data in the neighboring region, so as to select the salient features and to reduce the scale of the network structure by merging groups of neurons. The function can be written as:

$$\mathbf{O}_l^k = \sigma\left(\beta_l^k \operatorname{down}\left(\mathbf{o}_{l-1}^k\right) + \mathbf{b}_l^k\right) \tag{4}$$

where β is the multiplicative bias. $down(\cdot)$ represents the down-sampling function. We use the Max-Pooling operation as the down-sampling function.

After being processed by the fully-connected layers, inter-flow correlational features extracted by CNN are fed to the input of the *Temporal Modeling* module, which is described as follows.

C. *Learning Intra-flow Dependencies*

The traffic volume in WAN have significant temporal dependency, and the previous traffic state may have a long-term impact on the current state. RNN is a widely used temporal modeling tool for time-series data, such as natural language processing and time-series prediction [11]. However, RNN is not suitable for bursty and long-term traffic prediction because of its gradient exploding and vanishing problem [15]. LSTM introduces memory units to learn whether to forget the previous hidden state and update the hidden state in order to avoid the two problems. Thus, we employ LSTM layers as our key module for predicting TM in WAN.

Each LSTM unit consists of a single memory cell c_t, cell input and output (g_t and h_t) and three gates(input i_t, output o_t and forget f_t). The memory cell combines the previous cell states, current input, and previous output to update the hidden states. The forget gate determines whether the information should be forgotten or remembered. The output gate learns how the memory cell should affect the hidden states. For the LSTM units in *LSTM layer 1*, the $\mathbf{X}_t$ consists of the raw $\mathbf{TM}_t$ and its inter-flow correlational features extracted by the

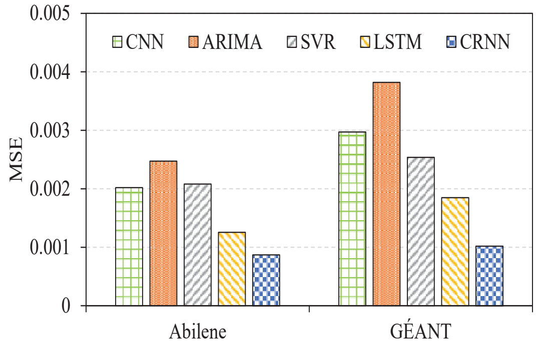
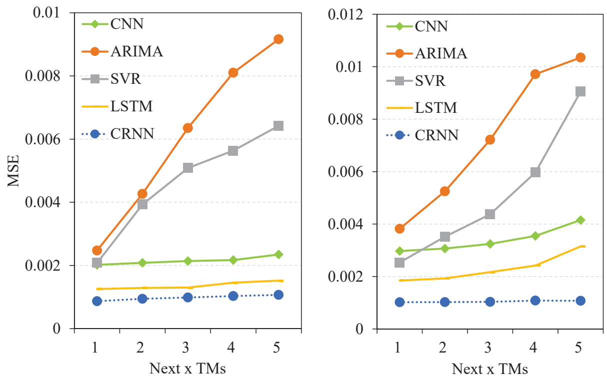


Fig. 5: Comparison of prediction errors for next single TM.



(a) Abilene

(b) GÉANT.

Fig. 6: Comparison of prediction errors for next multiple TMs.

Correlational Modeling module. Otherwise, $\mathbf{X}_t$ is the cell output of previous LSTM layer.

Finally, the output layer (i.e., *LSTM layer 2*) predicts the next q TMs. By comparing the predicted value and the true value, Back Propagation uses the error to update the parameters of CNN and LSTM structures.

V. Evaluation

A. Setting

The evaluation targets of this work include two parts: *1)* Accuracy of next single TM prediction with CRNN; *2)* Accuracy of next multiple TMs prediction with CRNN.

a) Datasets and data preparation: We consider two well-known WAN traces described in Section III-B: Abilene [18] and GÉANT [19]. We split the two datasets into training set and test set respectively with the ratio of $8:2$. Then, we normalize each traffic flow by their respective maximum value.

b) Baselines: We compare our model with the following widely used TM prediction methods:

- ARIMA [27]: Auto-Regressive Integrated Moving Average method is a classical approach for time-series forecasting problem. We use the implementation from the `statsmodels` [30] library.
- SVR [9]: Support Vector Regression is the regression version of Support Vector Machine and is also a well-known time-series analysis method for predicting the future values. We use the implementation from the `scikit-learn` [31] library.
- CNN [32]: We implement a CNN model to capture the inter-flow correlations features and predict TM directly.
- LSTM [13]: The major baseline is the LSTM, and it has been successfully applied to many time-series forecasting tasks including traffic prediction [16].

The CNN, LSTM and the proposed model CRNN are implemented in `TensorFlow` [33]. These four baselines are compared with the CRNN in terms of MSE.

c) Hyper-parameters: In the experiments, our CRNN model has three convolutional and pooling layers. For each iteration, we set 12 as the sequence length of per sample (i.e. the value of w). For CNN, LSTM and CRNN, the loss function is MSE, and the optimizer is *Adam* [34] with 0.0001 as learning rate and 32 as batch size. To avoid over-fitting, we select 0.5 as *Dropout* probability in CNN and LSTM. In CRNN, we set the length of the vector output by the CNN to be the same as the number of flows in the TM, it is 144 and 529 in Abilene and GÉANT, respectively.

B. Predicting Next Single TM

In this experiment, since the outputs of ARIMA and SVR must be one-dimensional, we employ them to predict the next traffic volume of each flow, and take the average of the prediction errors of each flow as the MSE of next single TM prediction.

Figure 5 shows the comparison among four baselines and the proposed model for predicting next single TM. We can see that the proposed model achieves the best performance in both two datasets. Specifically, the prediction results of the traditional time-series analysis methods are usually not ideal, demonstrating the limitations of these methods in modeling nonlinear and complex traffic volume. Furthermore, the prediction performance of CNN is worse than other NN-based methods, because its output has the same dimensions as the input. It shows that only capturing the inter-flow correlations cannot accurately predict future TM.

Since recurrent structures perform stronger nonlinear fitting, LSTM and CRNN perform better than ARIMA and SVR on both datasets. By considering the inter-flow correlation, compared with LSTM, CRNN achieves 30.7% and 44.8% reduction in terms of MSE on Abilene and GÉANT, respectively. These results well justify the effectiveness of emerging deep learning models on TM prediction in WAN, and more importantly, our CRNN model takes into account the inter-flow correlations within TMs.

C. Predicting Next Multiple TMs

By increasing q from 1 to 5, we evaluate the performance of these methods in long-term predictions. Figure 6 shows the change of prediction errors of various methods as the prediction interval increases. Overall, as the prediction interval becomes longer, the corresponding difficulty of prediction is getting greater, hence the prediction errors also increase. Specifically, the long-term prediction performance of ARIMA and SVR are much worse than the short-term prediction. The errors of deep learning methods increase slowly as prediction interval increases, and their overall performance is good. Although the errors of CNN have not changed much, the prediction accuracy of CNN is always low, because it cannot model the intra-flow dependencies.

Our CRNN model achieves the best prediction performance almost all the time. Especially in the long-term prediction, the differences between CRNN and other baselines are more significant. Take $q = 5$ as example, compared with LSTM, CRNN achieves 29.7% and 65.9% reduction in terms of MSE on Abilene and GÉANT, respectively. These results show that the strategy of combining CNN with RNN can better mine out the dynamic change patterns of TMs.

VI. Conclusion

In this paper, with the analysis of two real traces from WAN, we reveal that there exist abundant high inter-flow correlations in TMs and most strongly-correlated flows are in the same row or column in TM. These characteristics motivate us to take inter-flow correlation into account for TM prediction. Inspired by the ability of CNN to extract local correlations in matrices, we proposed CRNN model, which employs CNN to capture the inter-flow correlations and RNN (LSTM) to capture the intra-flow dependencies. Extensive experiments based on two real WAN datasets show that, CRNN can significantly improve the prediction accuracy for future TMs compared to state-of-the-art method. As the prediction interval becomes longer, the gap will be even larger.

References

[1] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 75–86, 2011.

[2] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4. ACM, 2009, pp. 267–278.

[3] V. Jalaparti, I. Bliznets, S. Kandula, B. Lucier, and I. Menache, "Dynamic pricing and traffic engineering for timely inter-datacenter transfers," in *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, 2016, pp. 73–86.

[4] Z. Cao, M. Kodialam, and T. Lakshman, "Traffic steering in software defined networks: planning and online routing," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 65–70.

[5] G. Lee, N. Tolia, P. Ranganathan, and R. H. Katz, "Topology-aware resource allocation for data-intensive workloads," in *Proceedings of the first ACM asia-pacific workshop on Workshop on systems*. ACM, 2010, pp. 1–6.

[6] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 15–26.

[7] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. ACM, 2008, pp. 151–156.

[8] M. S. Ahmed and A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*, 1979, no. 722.

[9] W.-C. Hong, "Application of seasonal svr with chaotic immune algorithm in traffic flow forecasting," *Neural Computing and Applications*, vol. 21, no. 3, pp. 583–593, 2012.

[10] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[11] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[15] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[16] 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.

[17] 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.

[18] "Abilene," http://www.cs.utexas.edu/~yzhang/research/AbileneTM/, 2004.

[19] 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.

[20] Y. Cheng, J. Geng, Y. Wang, J. Li, D. Li, and J. Wu, "Bridging machine learning and computer network research: a survey," *CCF Transactions on Networking*, vol. 1, no. 1-4, pp. 1–15, 2019.

[21] Q. Zhuo, Q. Li, H. Yan, and Y. Qi, "Long short-term memory neural network for network traffic prediction," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. IEEE, 2017, pp. 1–6.

[22] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 31, no. 1. ACM, 2003, pp. 206–217.

[23] 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 *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*. IEEE, 2011, pp. 95–102.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[25] D. Andreoletti, S. Troia, F. Musumeci, G. Silvia, G. A. Maier, and M. Tornatore, "Network traffic prediction based on diffusion convolutional recurrent neural networks," in *INFOCOM*, 2019, pp. 1–6.

[26] "Web traffic time series forecasting," https://www.kaggle.com/c/web-traffic-time-series-forecasting, 2017.

[27] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[28] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, "Deeptp: An end-to-end neural network for mobile cellular traffic prediction," *IEEE Network*, vol. 32, no. 6, pp. 108–115, 2018.

[29] R. Taylor, "Interpretation of the correlation coefficient: a basic review," *Journal of diagnostic medical sonography*, vol. 6, no. 1, pp. 35–39, 1990.

[30] "Statsmodels," http://www.statsmodels.org.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.