Transfer Learning Based Efficient Traffic Prediction with Limited Training Data

Sajal Saha, Anwar Haque, and *Greg Sidebottom
Department of Computer Science
University of Western Ontario, London, ON, Canada
*Juniper Networks, Kanata, ON, Canada
Email: {ssaha59, ahaque32} @uwo.ca, *gsidebot@juniper.net

Abstract—Efficient prediction of internet traffic is an essential part of Self Organizing Network (SON) for ensuring proactive management. There are many existing solutions for internet traffic prediction using machine and deep learning techniques. But designing individual predictive models for each service provider in the network is challenging due to data heterogeneity, scarcity, and abnormality. Moreover, the performance of the deep sequence model in network traffic prediction with limited training data has not been studied extensively in the current works. In this paper, we investigated and evaluated the performance of the deep transfer learning technique in traffic prediction with inadequate historical data leveraging the knowledge of our pre-trained model. First, we used a larger real-world traffic dataset for historical data, and then built a model for the target data. This technique can help build a model for the target data. This technique can help in two ways: re-use knowledge for quick learning and model parameter transfer technique uses the learned parameters from the source data and uses this knowledge to transfer learning helps to reduce the execution time for most cases, while the model’s accuracy is improved in transfer learning with a larger training session.

Index Terms—deep sequence model, internet traffic, IP traffic prediction, ISP, transfer learning

I. INTRODUCTION

Over the past decade, internet traffic growth has been accelerated by the rapid and widespread development of network technologies such as the Internet of Things (IoT), Industrial IoT (IIoT), 5G, and cloud computing. Also, according to the Cisco Annual Internet Report [1], there is an indication of a massive number of internet users by 2023 (approximately 5.3 billion). With the advancement of network and communication technologies and the increment of potential users, it is inevitable to design a more intelligent and self-organizing network (SON) capable of adapting the dynamic behavior and taking preemptive actions. The ISP (Internet Service Provider) network traffic is a critical metric for assessing network load and performance, and achieving a precise traffic prediction can be a suitable technique for proactive network management [2]. Most network management operations, such as resource allocation, short-term traffic scheduling or re-routing, long-term capacity planning, network architecture, and network anomaly detection, need an accurate traffic prediction tool. However, real-world internet traffic prediction is a challenging area.

Predicting internet traffic is commonly thought of as a time series forecasting problem that can be tackled by using traditional statistical techniques such as ARIMA [3], SARIMA, Holt-Winter [4], etc. But these models cannot predict the non-linear component of the actual internet traffic. With the advancement of artificial intelligence, machine learning [5] and deep learning [6] based traffic prediction models achieve superior performance improvement over the classical approaches. However, modern machine learning and deep learning algorithms require a massive amount of historical data to learn the general pattern in the traffic. Unfortunately, it is quite difficult to ensure enough data for building a new model separately for different geographical or network sectors. Also, it is unrealistic to design and deploy an individual model for each network segment for a larger ISP as the process is expensive and time-consuming.

Transfer learning [7] comes into play to solve the problem stated earlier by using the knowledge from the existing prediction model to devise a new model for another dataset. It is the crucial technology for the industrialization of the deep learning solution. The transfer learning approach consists of parameter transfer, domain adaption, and multi-task learning methods. The parameter transfer technique uses the learned parameters from the source data and uses this knowledge to build a model for the target data. This technique can help in two ways: re-use knowledge for quick learning and model smaller datasets. The main contributions of this work are:

- Analyzing different deep sequence models in the source domain to identify the best performing model for the target domain.
- Investigating transfer learning technique on the target domain consisting of multiple datasets based on the best two models from the source domain.
- Comparing the performance of standard and transfer learning in terms of average accuracy and execution time in the target domain for comparatively smaller datasets.

This paper is organized as follows. Section II describes
the literature review of current traffic prediction using deep learning models. Section III presents the methodology, including dataset preprocessing, deep transfer learning, deep learning models explanation, and experiment details. Section IV summarizes different deep learning methods’ performance with smaller datasets by applying both standard learning and transfer learning and draws a comparative picture among them. Finally, section V concludes our paper and sheds light on future research directions.

II. LITERATURE REVIEW

Wu, Qiong, et al. [8] proposed a novel mobile traffic prediction framework that combines the parameter-transfer [9] and domain adaption [10] approaches from deep transfer learning to enhance the model performance with a smaller dataset. The framework functionality is divided into two main parts: build the target prediction model with a massive dataset and then use the pre-trained model knowledge from the source domain, which faces the data-scarcity problem. Furthermore, they applied a GAN-based approach to solving the domain shift problem due to different data distribution between source and target domain. According to their experiment, the GAN-based domain adaption helps their model leverage the knowledge from the source domain to the target domain, giving a better prediction for a smaller dataset. However, the quality of the generated data using GAN is not presented. Moreover, since GAN is suffering to reach a stable training point [11], it is unclear how their GAN performs in generating samples.

Li, Ning, et al. [12] proposed a satellite traffic prediction model based on Gated Recurrent Unit (GRU) architecture that uses the transfer learning and particle filter algorithm for better prediction with a smaller dataset and lower training time. According to their experimental environment, they used similar distribution for the source and target domain in transfer learning. It is unclear how their prediction model will perform if the source and target distribution are asymmetric. The distribution is unlikely to be similar for the source and target domain in the real world, and that’s why it is crucial to validate the model performance with data coming from a different distribution than that source domain. A wireless cellular traffic prediction model has been proposed by Zeng, Qingtian, et al. [13], which is trained based on a cross-domain dataset. They also used the already trained model’s parameters for the target domain by adjusting the parameter’s values or transferring the learned features to improve the model accuracy. The experimental results showed the outperformance of the model with the transfer learning capability than the model having no transfer learning. In [14], Dridi, Aicha, et al. proposed a transfer-learning based deep learning model for time series classification and prediction. The transfer-learning technique is adapted in their model mainly for two reasons: better prediction with a smaller dataset and re-adaption of the already trained model for another domain. Their experimental results showed an outperformance of transfer learning in time-series prediction. However, their source and target domain data are drawn from the same distribution, which is unlikely to happen in the real world.

The current works show the usefulness of the transfer learning method in traffic prediction. But we found a lack of investigation in the performance comparison of the deep sequence model such as RNN and its varieties in real-world traffic prediction based on deep transfer learning. In this work, a comprehensive analysis of different deep sequence models has been performed for networks with limited training data based on transfer learning. In addition, we evaluate the performance using five different real-world internet traffic datasets.

III. OUR METHODOLOGY

A. Dataset and Preprocessing Steps

Real internet traffic telemetry on several high-speed interfaces has been used for this experiment. A total of five different datasets are used in our experiment. The source domain dataset, Dataset A, consists of 8563 data samples and it is used to build the predictive model for the source task. The other four datasets, Dataset B, C, D, and E are comparatively smaller in size, having 363, 369, 358, and 365 data instances, respectively. We used these four datasets for the predictive task in target domain. The smaller datasets for target domain are collected from four specific pair of source and destination node. All of our experimental datasets has similar kind of attribute such as timestamp and corresponding amount of internet traffic on that particular time-point. But their data distribution is different.

B. Deep Transfer Learning (DTL)

In this subsection, firstly we discuss fundamental of transfer learning. Then we present our motivation of using transfer learning in traffic prediction and details explanation on how to use transfer learning to solve existing research gap.

1) Our Approach: Firstly, we explore five different deep sequence model for single-step prediction task, $T_S$ in source domain. $D_S$. As a Recurrent Neural Network (RNN) is the fundamental deep sequence model capable of handling sequence data such as time series, we first began our experiment with RNN. The time-series data has been converted into a tabular format with feature and target variables for supervised learning. The window width for data windowing techniques has been determined using ACF and PACF plot information. Next, we extend our experiment with two additional model architecture: Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) as RNN has a problem of vanishing gradient in case of handling longer sequence. The GRU model has special type of gate which control the information flow between two cells and can retain context information from longer sequence than RNN. LSTM has also similar concept but the number of gates in more in LSTM than GRU. Finally, we incorporate two sequence-to-sequence model architecture that can predict a output sequence given an input sequence. It is made up of two recurrent neural networks, one of which is an encoder and the other a decoder. The encoder turns the
input sequence into a fixed-length context vector, which the decoder receives. The decoder takes the context vector and the encoder’s end state as input and produces a series of outputs. We consider LSTM model as both encoder and decoder in our sequence model. However, the encoder-decoder model cannot identify significant contextual relationships from extended sequence data, lowering model performance and accuracy. The encoder-decoder model’s extra attention layer, on the other hand, can detect relevance in the long sequence data. So that, we consider LSTM encoder decoder with attention layer for prediction task in the source domain. In the source domain, we explored different deep sequence model to identify the best-performing for the target domain, \( D_T \).

For the target domain prediction task \( T_T \), we consider two best-performing models from the source domain. And four different datasets with fewer samples have been considered to compare performances between the two best-performing models. We considered all the hidden layers from the source domain prediction model for our target domain task as both \( T_S \) and \( T_T \) are very similar tasks. The last layer of the pre-trained model from the source domain has been removed for the target domain prediction. First, we freeze all the reused layers from the pre-trained model for the first few epochs, i.e., make the weights for the corresponding layer non-trainable. Then, we unfreeze all the hidden layers so that the back-propagation can tweak the corresponding weights to gain better results. We consider two different learning rates during the experiment. A lower learning rate has been considered when we unfreeze the layers so that models can avoid tweaking their fine-tune weights. Finally, the experiment has been conducted for different epoch sizes to show a comparative analysis between standard learning and transfer learning training time.

C. Evaluation Metrics

We used Mean Absolute Percentage Error (MAPE) to estimate the performance of our traffic forecasting models. The performance metric identifies the deviation of the predicted result from the original data. For example, MAPE error represents the average percentage of fluctuation between the actual value and predicted value. Therefore, we can define our performance metric mathematically as follow:

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{p_i - o_i}{o_i} \right| \times 100% \tag{1}
\]

\[
Accuracy = (100 - MAPE)\%	ag{2}
\]

Here, \( p_i \) and \( o_i \) are predicted and original value respectively and \( n \) is the total number of test instance.

D. Software and Hardware Preliminaries

We used Python and deep learning library TensorFlow-Keras [15] to conduct the experiments. Our computer has the configuration of Intel (R) i5-9500T CPU@2.20GHz, 8GB memory, and a 64-bit Windows operating system.

### IV. RESULTS AND DISCUSSION

A. Result analysis

1) Source domain prediction model: We applied several deep sequence models such as RNN and their variants in this phase, including LSTM, LSTM En De, LSTM En De Attn, and GRU, to identify the best performing models for our source domain. All our model training continued for 100 epochs with a batch size of 16. The prediction accuracy for all the prediction models on dataset \( A \) is summarized in Table I. According to our experimental results, RNN is the worst performing model with an accuracy of 92.49%, which is improved by more than 2% and 1% after applying LSTM and GRU, respectively. Since LSTM and GRU can retain the information from the longer sequence, it is expected to have a better performance than RNN for final prediction. Finally, we achieved our best prediction result using LSTM En De with an accuracy of 96.06%, which is around a 4% improvement compared to the RNN model. The LSTM En De Attn also gave us around 4% higher accuracy (96.05%) when compared with the RNN model, which is very close to the LSTM En De model performance. According to our experiment, the encoder-decoder architecture-based models performed better, and we considered our two best-performing models in the second step of our investigation. We used two source domain model-based LSTM En De and LSTM En De Attn to compare and validate the performance of deep transfer learning.

2) Target domain prediction model: In this phase, we design a predictive model for datasets \( B, C, D, \) and \( E \) using both standard learning and transfer learning approaches. For both strategies, 70% of the data has been used for the training, while the remaining 30% has been applied to test the model. Two best-performing models, such as LSTM En De and LSTM En De Attn from the source domain, are used to design the target domain’s predictive models. Also, we trained the target domain models for different epoch lengths to identify the best training settings and make a comparative analysis between standard learning and transfer learning in terms of total training time and accuracy. In the case of transfer learning, we freeze the reused layers during the first ten epochs with a more significant learning rate of 0.001, allowing the new layers to learn reasonable weights. Then we unfreeze the reused layers and decrease the learning rate to 0.0001 to avoid damaging the reused weights. Finally, to calculate the average training time and average accuracy of the prediction in the target domain, we executed the experiment five times and took the corresponding metric’s average.
| Dataset B | Dataset C | Dataset D | Dataset E |
|----------|----------|----------|----------|
| **Epoch** | **Avg. Acc. (%)** | **Time (S)** | **Avg. Acc. (%)** | **Time (S)** | **Avg. Acc. (%)** | **Time (S)** | **Avg. Acc. (%)** | **Time (S)** |
| 250      | 83.50     | 33.15     | 84.08     | 22.03     | 70.94     | 33.44     | 76.77     | 27.44     | 73.54     | 37.99     | 76.24     | 26.94     | 82.68     | 42.81     | 84.43     | 27.75     |
| 200      | 82.72     | 26.48     | 84.26     | 23.11     | 72.88     | 27.38     | 76.97     | 23.39     | 74.32     | 31.40     | 78.43     | 23.41     | 82.95     | 35.76     | 84.13     | 23.65     |
| 150      | 84.19     | 21.28     | 84.36     | 19.27     | 77.24     | 22.10     | 77.00     | 19.48     | 74.59     | 24.59     | 80.25     | 19.13     | 84.17     | 27.53     | 84.39     | 19.46     |
| 100      | 83.37     | 15.24     | 82.41     | 14.97     | 77.25     | 15.60     | 75.60     | 15.09     | 68.89     | 17.74     | 82.91     | 14.84     | 84.38     | 18.94     | 82.13     | 14.97     |
| 50       | 86.42     | 9.52      | 81.06     | 10.68     | 77.47     | 10.01     | 73.17     | 10.87     | 79.42     | 10.90     | 79.39     | 10.71     | 84.58     | 11.60     | 80.91     | 10.72     |

The details results for four different data sets are presented in Table II and Table III, respectively, for the LSTM En De and LSTM En De Attn model. The average accuracy (Avg. Acc.) and average total training time in second (Time (S)) indicates the mean accuracy and mean training time for five different runs. For LSTM En De model, the model training time is much lesser in transfer learning than standard learning for all target domain’s datasets. The training time gap between the two learning strategies is increased with the epoch size, which indicates that transfer learning has the chance of quick learning in case of longer training. Also, the graphs indicate better accuracy with transfer learning for the first three epochs settings, such as 250, 200, and 150, for all four datasets. Our results indicate a similar pattern in the LSTM En De Attn model, although the gap between average time is smaller than expected. The average time for all target domain’s datasets. The training time gap between the two learning strategies is increased with the epoch size, which indicates that transfer learning has the chance of quick learning in case of longer training. Also, the graphs indicate better accuracy with transfer learning for the first three epochs settings, such as 250, 200, and 150, for all four datasets. Our results indicate a similar pattern in the LSTM En De Attn model, although the gap between average time is smaller than in the LSTM En De model. Transfer learning gave us better training time or better prediction accuracy for all settings and performed better for larger epochs in both metrics.

V. CONCLUSION

In this work, we evaluate the effectiveness of the deep transfer learning technique in real-world internet traffic prediction for the network with smaller training data. It is practically challenging to arrange a large dataset for efficient model training. Furthermore, designing an individual model for each service provider in a large network is infeasible due to time and resource complexity. Therefore, a deep transfer learning based traffic prediction methodology is proposed that is expected to provide better performance with a comparatively smaller dataset. Our experiment used one large dataset for the source domain and four smaller datasets for the target domain.

REFERENCES

[1] “Cisco annual internet report a white paper 2020,” Mar 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report-white-paper-c11-714190.html
[2] Y. Han, Y. Jing, and K. Li, “Multi-step prediction for the network traffic based on echo state network optimized by quantum-behaved fruit fly optimization algorithm,” in 2017 29th Chinese Control And Decision Conference (CCDC). IEEE, 2017, pp. 2270–2274.
[3] S. Wheelwright, S. Makridakis, and R. J. Hyndman, Forecasting: methods and applications. John Wiley & Sons, 1998.
[4] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. John Wiley & Sons, 2015.
[5] S. Saha, A. Haque, and G. Sidebottom, “Towards an ensemble regressor model for isp traffic prediction with anomaly detection and mitigation,” in 2022 International Symposium on Networks, Computers and Communications (ISNCC), 2022, pp. 1–6.
[6] ——, “Deep sequence modeling for anomalous isp traffic prediction,” in ICC 2022 - IEEE International Conference on Communications, 2022, pp. 5439–5444.
[7] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” Journal of Big data, vol. 3, no. 1, pp. 1–40, 2016.
[8] Q. Wu, K. He, X. Chen, S. Yu, and J. Zhang, “Deep transfer learning across cities for mobile traffic prediction,” IEEE/ACM Transactions on Networking, 2021.
[9] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2009.
[10] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” Neurocomputing, vol. 312, pp. 135–153, 2018.
[11] L. Weng, “From gan to wgan,” arXiv preprint arXiv:1904.08994, 2019.
[12] N. Li, L. Hu, Z.-L. Deng, T. Su, and J.-W. Liu, “Research on gnu network satellite traffic prediction based on transfer learning,” Wireless Personal Communications, vol. 118, no. 1, pp. 815–827, 2021.
[13] Q. Zeng, Q. Sun, G. Chen, H. Duan, C. Li, and G. Song, “Traffic prediction of wireless cellular networks based on deep transfer learning and cross-domain data,” IEEE access, vol. 8, pp. 172 387–172 397, 2020.
[14] A. Dridi, H. Afifi, H. Moungla, and C. Boucetta, “Transfer learning for classification and prediction of time series for next generation networks,” in ICC 2021-IEEE International Conference on Communications. IEEE, 2021, pp. 1–6.
[15] F. Chollet et al., “Keras,” https://keras.io, 2015.