One for All: Traffic Prediction at Heterogeneous 5G Edge with Data-Efficient Transfer Learning

Xi Chen*, Ju Wang*, Hang Li*, Yi Tian Xu*, Di Wu*, Xue Liu*, Gregory Dudek*, Taeseop Lee†, Intaik Park†
*Samsung Electronics, Canada
†Samsung Electronics, Korea (South)

***Abstract*—By placing the computing, storage and networking resources close to the end users, distributed edge computing greatly benefits the performance of 5G communication systems. However, as a tradeoff, resources on the edge are usually limited and imbalanced among the heterogeneous edge nodes. To overcome this drawback, this paper proposes a Transfer Learning based Prediction (TLP) framework that allows the edge nodes to share their resources and data in an efficient manner. In particular, the TLP framework focuses on the prediction of the future traffic load, which is a key reference for many automated network functions. To enhance the efficiency of data and bandwidth, TLP first learns a base model on a data-abundant edge node (the source), and then transfers this model (instead of data) to other data-limited nodes (the targets). To achieve a delicate balance between maintaining common features and learning target-specific features, we develop a new transfer learning technique named Similarity-based Elastic Weight Consolidation (SEWC), and integrate it into TLP. Experiments on real-world data illustrate that, compared to the state-of-the-art methods, TLP-SEWC reduces the Mean Absolute Error (MAE) of traffic prediction by up to 57.9%.**

I. Introduction

Traffic volume (or traffic for short), being one of the most fundamental measures of 5G networks, is a key reference variable for many 5G edge operations, such as predictive load balancing [1], and automated network slicing [2]. The accurate and timely prediction of traffic is expected to improve the efficacy and efficiency of 5G edge systems [3].

To achieve this, it is desirable to deploy the prediction function on the edge nodes. On one hand, a centralized framework could be inefficient. The data migration from edge nodes to centralized cloud servers would consume a lot of bandwidth and scheduling resources. These resources are usually quite occupied by core 5G functions, and may not have adequate quota left for the transmission of big data [4]. In addition, service agreements and data compliance may prohibit data consolidation at the cloud or back-end. On the other hand, it is beneficial to push the prediction to the edge, since computing resources have been already deployed at the edge for other tasks [3]. With prediction at the edge, the predicted results can be seamlessly and promptly integrated into other functions.

Existing prediction methods [5], [6] can not support such an edge-oriented framework, due to their dependency on centralized big data. At the 5G edge, data and resources are actually **limited** and **imbalanced**. First of all, it could be very expensive and unworthy to maintain a big data set at every edge node, mainly due to 1) the Capital Expenditures (CapEx) of storage at the edge [7], and 2) the degradation of resource utilization caused by frequent disk writes [8]. Consequently, most edge nodes can only keep their fine-grained logs over a short period, and compress the other obsolete logs into coarse-grained statistics [9]. Moreover, the storage capabilities of the heterogeneous edge nodes differ significantly from each other [10]. Hence, some powerful nodes could afford to store bigger data sets than others. Such limited and imbalanced data could bring serious overfitting to the existing methods.

To deal with this issue, in this paper, we propose TLP, a **T**ransfer **L**earning based **P**rediction framework, which transfers a base model trained on a data-abundant edge node (the source) to other data-limited nodes (the targets). The heart of our design is data and bandwidth efficiency: we make full use of all available data from both the data-abundant source node and the data-limited target nodes without any data migration. A major **challenge** in achieving this goal is that *it is hard to maintain the general features while updating the node-specific features with a small amount of data at the edge.* The state-of-the-art transfer learning based prediction methods [11] can not deal with this challenge, since they spare no effort to preserve the learned feature during the model transfer. Federated learning based methods are not suitable as well. They focus heavily on training a general model by aggregating the edge gradient updates, and yet pay little attention to customizing models for the edge nodes [12].

To overcome this challenge, TLP embraces our newly developed transfer learning technique, namely Similarity-based Elastic Weight Consolidation (SEWC). Compared to the existing techniques, the proposed SEWC is more fine-grained and node-customized. It uses weight-level importance scores to balance between 1) keeping an NN weight learned for generality and 2) updating this weight to serve the specific target. SEWC also customizes the model transfer for each individual edge node by adjusting the importance scores according to the similarity between the source and the target.

We conduct experiments on a 50-node data set collected over an anonymous city by a mobile carrier partner over a year. Compared to the state of the art, TLP reduces the prediction Mean Absolute Errors (MAEs) by up to 57.9%.

II. Design of TLP

A. Definitions and Problem Statement

Suppose there exist $N+1$ edge nodes in a communication network. Everyday, starting from $0:00$, each node records

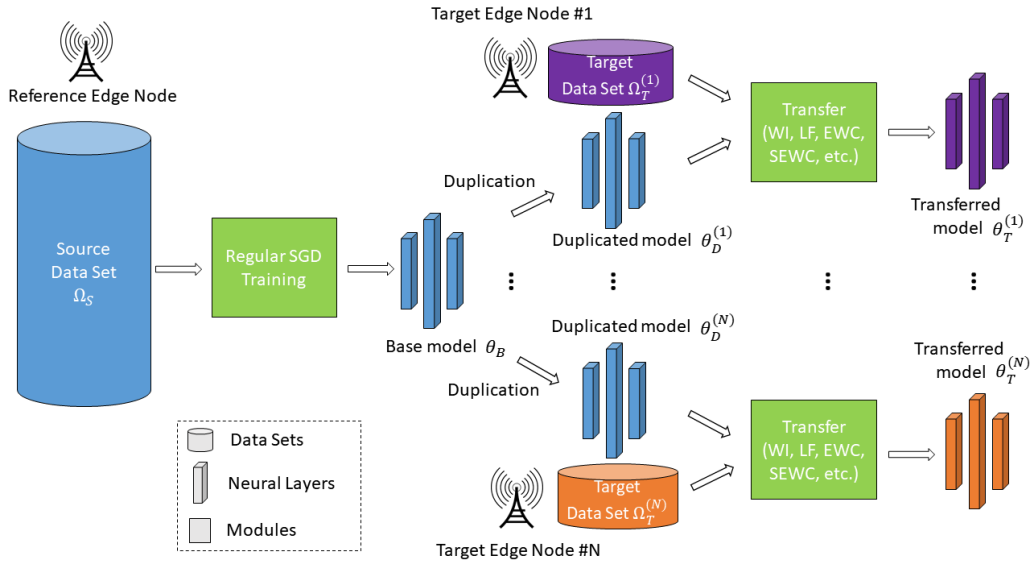


Fig. 1: The overview of TLP.

one traffic data sample every $24/M$ hours. This give us M traffic data samples per day. Every traffic data sample $s[t]$ is a tuple of the traffic amount $a[t]$ and the time stamp t, i.e., $s[t] = [a[t]; t]$.

As discussed in the Section I, due to the real-world constraints on the heterogeneous 5G edge, most edge nodes store their fine-grained data (that are useful for training) only over a short period. Their obsolete logs are compressed into coarse-grained formats (e.g., daily means of traffic), which is not so useful for real-time accurate traffic prediction. Only a few (powerful) reference edge nodes can afford to store their fine-grained traffic history over a long period. Hence, the data are limited and imbalanced across different nodes.

For clarity, we study a scenario where there exists only one reference node. This reference node keeps a fine-grained traffic data set S_S over the past m days (e.g., 365 days), which serves as the source for the transfer learning. The rest N edge nodes can only store their fine-grained traffic data of a much shorter period, i.e., the past n days with $n \ll m$ (e.g., 7 days). These nodes are the targets of the transfer learning. The data set of the kth target node in the past n days is denotes as $S_T^{(k)}$.

We further transform the traffic data samples in a way that is more suitable for time sequence prediction. We take a window of $c+1$ traffic data samples, use the first c samples for the input vector $x[t]$, and take the last traffic data sample as the prediction output $y[t]$. Concretely, we define

$$x[t] = [a[t], a[t+1], \cdots, a[t+c-1]; t, t+1, \cdots, t+c-1],$$

and $y[t] = a[t+c]$.

A transformed data sample $u[t]$ is now a tuple as $u[t] = \{x[t]; y[t]\}$. We then move the window forward by one sample to generate $u[t+1]$, and so on until the transformation is applied to the whole source traffic set S_S. In this way, we produce a transformed source data set $\Omega_S = \{X_S, Y_S\}$, where $X_S = \{x[t]\}$ and $Y_S = \{y[t]\}$ denote all the input and output vectors transformed from S_S, respectively. Similarly, we produce the transformed target data sets $\{\Omega_T^{(k)}\}$.

Given the source data set Ω_S and the target data sets $\{\Omega_T^{(k)}\}$, we aim to develop one base model θ_B for the source node and N transferred models $\{\theta_T^{(k)}\}$ for the target nodes. When a new data sample $x[t]$ comes at the kth target, the model $\theta_T^{(k)}$ needs to predict $y[t]$ as accurate as possible. Let us further define the prediction loss function over the target data set $\Omega_T^{(k)}$ as

$$L^{(k)} = \frac{1}{\left|\Omega_T^{(k)}\right|} \sum_{y(t) \in \Omega_T^{(k)}} d(y[t], \hat{y}[t]), \tag{1}$$

where $\left|\Omega_T^{(k)}\right|$ is the size of the data set, $\hat{y}[t]$ is the prediction of the ground truth $y[t]$, and $d(\cdot, \cdot)$ represents an error metric. Available error metrics include, but are not limited to, absolute error, square error, and root square error.

Problem Statment: Given a big source data set Ω_S and a small target data set $\Omega_T^{(k)}$, develop a model $\theta_T^{(k)}$ for the kth edge node, so as to minimize $L^{(k)}$ defined in Eq. (1).

B. Overview of TLP

Fig. 1 presents the overall flow of TLP. At the beginning, TLP utilizes the large amount of data in the source set Ω_S to train a base model θ_B, using the regular Stochastic Gradient Descent (SGD) technique [13]. Then, we replicate this base model into N duplicated models $\{\theta_D^{(k)}\}$, i.e., $\forall k, \theta_D^{(k)} = \theta_B$, and assign each of them to the corresponding target edge node. Note that exchanging a prediction model is much more bandwidth-efficiency for the edge than migrating data.

Next, we take the kth target edge node as an example. For the sake of clarity, we ignore the superscript $\cdot^{(k)}$ in the rest of this section. So, given the duplicated model θ_D, TLP will transfer this model to support the target node using the target data set Ω_T, which has much less traffic history than Ω_S. The output of our TLP framework is the transferred model θ_T.

The transfer module could apply traditional techniques such as Weight Initiation (WI) and Layer Freezing (LF) [14], [15]. However, these techniques are not fine-grained enough to deal with the aforementioned challenge.

C. Transfer via Elastic Weight Consolidation (EWC)

The EWC approach is a weight-level transferring/fine-tuning method, which helps us better utilize the spatial-temporal similarity among edge nodes. Essentially, EWC analyzes the importance of different weights after training a NN. Based on this analysis, EWC adds penalties (regularization terms) to the loss function during future retraining. A more important weight is given a larger penalty, so that it becomes harder to update during the retraining process.

The importance of each weight is estimated by the diagonal of the Fisher Information Matrix (FIM) [16]. This diagonal of FIM is considered to be equivalent to the second-order derivative of the loss, which reflects the curvature of the loss function. In other words, it implies how fast the loss function varies with respect to a certain weight. Let $F_{i,j}$ be the diagonal value of $w_{i,j}$'s FIM. An advantage of $F_{i,j}$ lies in its small computational cost. Although $F_{i,j}$ represents the second-order derivative, it can be computed using the first-order derivative of the loss function [17] as

$$F_{i,j} = \mathbb{E}_{(x,y)\sim\Omega_S}[(\frac{\partial L}{\partial w_{i,j}[0]})^2]. \quad (2)$$

Note that $F_{i,j}$ is now a constant score that measures the importance of $w_{i,j}$. Given this importance score, the loss function to transfer/fine-tune the NN is now updated as

$$L_{EWC} = L + \lambda L_R = L + \lambda \sum_{i,j} \frac{1}{2} F_{i,j}(w_{i,j} - w_{i,j}[0])^2, \quad (3)$$

where $w_{i,j}|_{\tau=0}$ is the initial value of $w_{i,j}$ before transferring, i.e., the initial weight in the duplicated model θ_D, and λ controls the balance between the prediction loss and the regularization term. In this way, the EWC updates the weights of transferred model θ_T as follows:

$$w_{i,j}[\tau+1] = w_{i,j}[\tau] - \eta \cdot \nabla L_{EWC}, \quad (4)$$

where $\nabla L_{EWC} \doteq \partial L_{EWC}/\partial w_{i,j}[\tau]$, and η is a learning rate.

To sum up, given a base/duplicated model, we first calculate the importance score $F_{i,j}$ of each $w_{i,j}$.and then apply Eq. (4) in transferring the model to a target edge node.

D. Transfer via Similarity-based EWC (SEWC)

Ideally, if a target is quite similar to the source, we prefer to keep the importance scores high to maintain the common features learned from the source. Otherwise, we lower the importance scores, so that the transfer model would not be overly constrained by the base model.

To this end, we develop the Similarity-based EWC (SEWC) technique. Concretely, SEWC first calculates the source traffic distribution $\vec{H}_S$ and the target traffic distribution $\vec{H}_T$ as

$$\vec{H}_S = histo(\Omega_S), \text{ and } \vec{H}_T = histo(\Omega_T), \quad (5)$$

where $histo(\cdot)$ calculate the normalized histogram of the traffic within a data set. It then uses the cosine similarity between $\vec{H}_S$ and $\vec{H}_T$ as the similarity score μ between the source and the target, i.e.,

$$\mu = (\vec{H}_S \cdot \vec{H}_T)/(||\vec{H}_S|| \times ||\vec{H}_T||). \quad (6)$$

This similarity score is then multiplied to EWC importance scores to generate similarity-aware importance scores $F^{sim}_{i,j}$ as

$$F^{sim}_{i,j} = \mu F_{i,j}. \quad (7)$$

Accordingly, the loss function to transfer the base model is

$$L_{SEWC} = L + \lambda \sum_{i,j} \frac{1}{2} \mu F_{i,j}(w_{i,j} - w_{i,j}[0])^2. \quad (8)$$

The weights are updated at every epoch as

$$w_{i,j}[\tau+1] = w_{i,j}[\tau] - \eta \cdot \nabla L_{SEWC}. \quad (9)$$

In this way, SEWC customizes the model transfer for each individual target node.

III. Evaluation

A. Data Set Description

Our data set records the traffic history of 50 edge nodes over one year from January 1st to December 31th. The edge nodes are edge BSs geographically randomly sampled from a mobile network covering an anonymous city. Every 15 minutes, each node records one sample data of its total traffic and the time stamp (together with other information not used in this paper). Thus, the total number of traffic data samples is $\frac{60}{15}\ samples \times 24\ hours \times 365\ days \times 50\ nodes = 1,752,000$[1]. We use the first edge node as the source and the rest as the targets, and define the following data sets.

- **Source training set** is the first 10-month traffic history (i.e., January 1st to October 31th) of the source node (i.e., the first edge node). It is used to train the base model.
- **Target training sets** are the November traffic records of the target nodes. Each of these data set is used to transfer the base model to the corresponding node. They are also used for the training of some baseline methods. Note that the size of each targeting training set is less than $3,000$.
- **Target testing sets** are the December traffic records of the target nodes, and are used to test all methods.

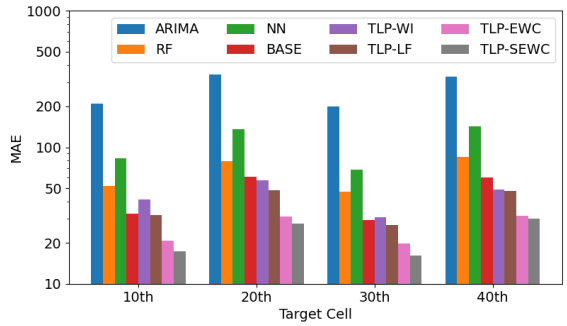
B. Metrics

Metrics on the prediction accuracy are defined as follows.

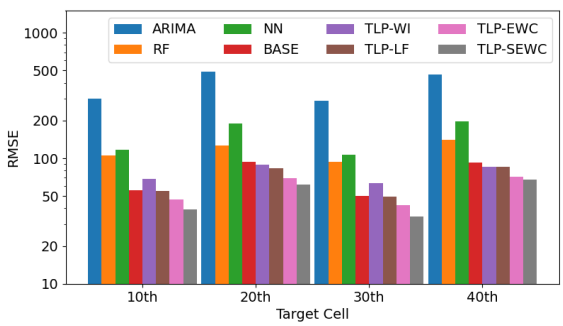
Mean Absolute Error (MAE) measures the average of the absolute differences between the predicted traffic and the ground truth. The MAE of kth edge node is:

$$MAE^{(k)} = \frac{1}{\left|\Omega^{(k)}_T\right|} \sum_{y(t)\in\Omega^{(k)}_T} |y[t] - \hat{y}[t]|, \quad (10)$$

[1]More detailed information of the data set cannot be released to the public, due to the agreement between the authors and the partner.



(a) The MAEs of the example edge nodes.



(b) The RMSEs of the example edge nodes.

Fig. 2: The performance of different methods at the example edge nodes.

where $\left|\Omega_T^{(k)}\right|$ is the size of the data set of kth edge node, $|y[t] - \hat{y}[t]|$ calculates the absolute prediction error.

Root Mean Square Error (RMSE) represents the square root of the mean of the differences between the predicted traffic and the ground truth. The RMSE of kth node is:

$$RMSE^{(k)} = \sqrt{\frac{1}{\left|\Omega_T^{(k)}\right|} \sum_{y(t)\in\Omega_T^{(k)}} (y[t] - \hat{y}[t])^2}. \quad (11)$$

C. Methods Evaluated

We first implement the following baseline methods[2].

- **ARIMA**, or Auto-Regressive Integrated Moving Average, is a well know time series prediction method.
- **RF**, or Random Forest, based regression is an ensemble learning method.
- **NN**, or Neural Network, establishes a collection of connected neurons for the traffic prediction task. In this paper, we employ an encoder-decoder NN structure.
- **BASE** reuse the same structure of NN. Yet, it is trained on the source training set and applied to the target testing sets. This is the most "straightforward" model transfer.

For every target edge node, ARIMA, RF, and NN methods are trained on the target training set, while BASE is trained on the source training set.

We further establish our TLP framework using BASE as the base model to be transfer by four transfer learning techniques, and implement the following four methods.

- **TLP-WI** adopts the Weight Initiation (WI) technique in our TLP framework. It serves as a representative of the state-of-the-art deep/transfer learning based traffic prediction methods, e.g., those in [11], [18].
- **TLP-LF** uses the Layer Freezing (LF) technique in our TLP framework. We freeze the first 30% layers[3].
- **TLP-EWC** employs the Elastic Weight Consolidation (EWC) technique in our TLP framework.
- **TLP-SEWC** integrates the Similarity-based EWC (SEWC) technique in our TLP framework.

[2]Due to space limit, we do not present the results of LSTM NNs, of which the prediction errors were surprisingly high (e.g., 5 times higher than ARIMA). The reason is that a LSTM NN by its nature contains numerous weights, of which the number is much larger than the number of data points in the small target training sets. Thus, the LSTM NNs are significantly overfitted.

[3]This 30% is the result of hyper-parameter tuning on freezing percentage.

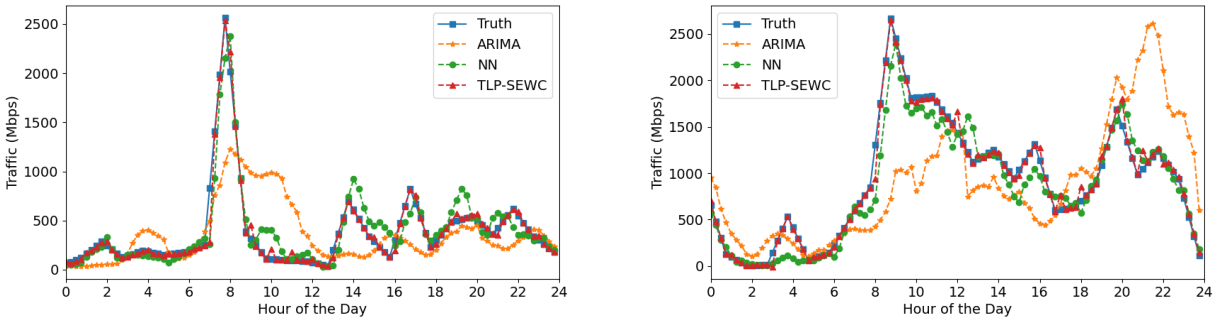
D. Evaluation Results on Example Edge Nodes

We first take the 10th, 20th, 30th, and 40th edge nodes as examples, and present the prediction results on them.

Fig. 2(a) and Fig. 2(b) compare different methods on the example nodes in terms of MAE and RMSE, respectively. As expected, the non-transferring methods, i.e., ARIMA, RF and NN, achieve much worse performance than the transfer learning based methods, i.e., BASE and TLP methods. This can be explained by the spatial-temporal similarity between source and target nodes. Since the source training set has an adequate amount of data and can better represent general traffic pattern in the testing data, the distribution of the testing set is more similar to the source training set than to the target training set. Therefore, it is natural that methods utilizing the source data achieve better performance.

Among the non-transferring methods, there is **an interesting phenomenon**: although the NN method is more sophisticated than the RF method, the performance of RF is better than that of NN. This is the consequence of small training data sets. In our implementation, the number of parameters in the NN method ($\sim 150k$) is much larger than the number of data points in a target data set ($\sim 3k$). In this case, the NN method is overfitted to the small amount of data, and thus achieves worse performance than a simple method with less parameters, e.g., the RF method. This could easily happen in many real-world scenarios, where data are limited, and illustrates why transfer learning is needed for deep models in these scenarios.

Among the transfer learning based methods, BASE and TLP-WI result in higher errors than TLP-LF, TLP-EWC and TLP-SEWC. Taking TLP-WI as the state-of-the-art baseline, TLP-LF, TLP-EWC, TLP-SEWC reduce the MAEs by up to 22.7%, 49.8% and 57.9%, respectively. Similarly, the reductions of RMSEs by TLP-LF, TLP-EWC and TLP-SEWC are up to 22.6%, 33.5%, and 42.8%, respectively. This illustrates that, by utilizing fine-grained transfer learning methods, our proposed TLP framework further advances the state of the art. In addition, by adapting the importance scores to the source-



(a) The 10th edge node.

(b) The 20th edge node.

Fig. 3: Traffic prediction results on December 1st at the example edge nodes.

target similarity, TLP-SEWC outperforms TLP-EWC by up to 8.1% and 9.3% in terms of MAE and RMSE, respectively.

We next visualize the prediction results. Due to the space limit, we take the results on December 1st as an example, and plot ARIMA, NN and TLP-SEWC results in Fig. 3. We can see that ARIMA experiences a hard time in following the changes in the load, especially around the peak hours (e.g., 7:30 - 8:30 in Fig. 3(a) and 8:00 - 10:00 in Fig. 3(b)). NN improves the prediction accuracy over ARIMA, and yet still exhibits some large errors from time to time. Its peak prediction usually looks like one-slot delayed version of the ground truth. With the help of transfer learning, TLP-SEWC reduces these errors, and achieves the best performance.

E. Evaluation Results on All Edge Nodes

We next illustrate that TLP's improvements on all edge nodes, and plot the Cumulative Distribution Functions (CDFs) of MAE and RMSE over all target nodes in Fig. 4(a) and Fig. 4(b), respectively. Note that the more to the left a CDF curve is, the better the performance is. We again observe that the performance increases, as the transfer operation goes more and more fine-grained from BASE to TLP-SEWC. Specifically, there is a notable gap between TLP-SEWC and TLP-WI in either MAE or RMSE. This suggests that the weight-level transfer learning techniques can significantly improve the prediction accuracy of the transferred models. Moreover, we can find a clear improvement by TLP-SEWC over TLP-EWC. This confirms that utilizing the source-target similarity to guide the model transfer is beneficial.

IV. Related Work

The related work falls into roughly the three categories.

A. Statistical model based traffic prediction

In this category, the traffic data is modeled and predicted based on their statistics or probabilistic distributions. Many studies show that traffic data over a different period or among different sites have certain statistical distributions. Thus, statistical models, such as mobility model [19], network traffic model [20], and α-stable model [21], have been applied into the early traffic prediction work.

In real world, the accuracy of the statistical model based traffic prediction is limited by the data distribution variations.

B. Temporal information based traffic prediction

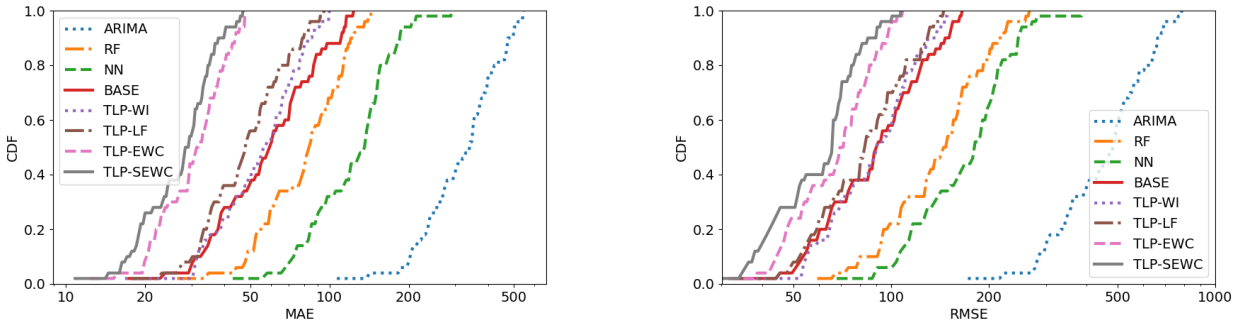
In this category, the temporal correlation in traffic history is leveraged for traffic prediction. On one hand, various of time-series analysis based methods are applied for the traffic prediction, e.g., ARIMA [22], Holt-Winters method [2] and Markov processes [23]. On the other hand, ML-based nonlinear prediction methods, such as support vector regression [24], and Long Short-Term Memory (LSTM) [18] are employed for traffic prediction. With their strong ability to learn from big data, ML-based methods achieve satisfactory prediction accuracy.

Overall, these temporal information based methods achieve a better performance than the statistical model based traffic prediction, since they are not sensitive to the statistical distribution variations. However, their accuracy is still limited since they only consider the temporal traffic correlation within an edge node, neglecting the potential benefits of jointly considering spatial correlations across different nodes.

C. Spatial-temporal information based traffic prediction

To further improve the prediction accuracy, some studies try to capture both the spatial and temporal characteristics of traffic with advanced ML methods [11], [5], [25]. For example, Wang *et al.* [26] propose a hybrid deep learning based traffic prediction method, where autoencoder is used for the spatial modeling, and LSTM is used for the temporal modeling. Zhang *et al.* [11] design a spatial-temporal cross-domain neural network framework, which exploits similarities of different traffics to improve the prediction accuracy. To fully use the spatial-temporal correlation, Liu *et al.* [6] propose a graph neural network based method to better capture topology and relation of cellular nodes.

Although the above advanced ML-based methods achieve better accuracy than past studies, their accuracy relies on an adequate amount of big data. Unfortunately, the available data for traffic prediction in real scenarios is usually limited and imbalanced, due to the high cost of gathering and maintaining



(a) The CDF of MAE over all edge nodes.

(b) The CDF of RMSE over all edge nodes.

Fig. 4: The performance of different methods over all edge nodes.

big data at the edge. As a result, the ML-based methods may suffer from significant prediction errors due to overfitting.

Unlike all the existing studies, this paper develops a data-efficient framework TLP, which integrates the newly developed transfer learning technique - Similarity-based Elastic Weight Consolidation (SEWC). TLP-SEWC is a more practical and accurate solution for traffic prediction in 5G edge networks, where only a limited amount of imbalanced data are available.

V. Conclusion

This paper introduces TLP, a transfer learning based traffic prediction framework that can achieve high prediction accuracy with limited and imbalanced data at the 5G edge. TLP embraces a newly developed transfer learning techniques, namely SEWC, which transfers a well-trained prediction model to a target edge node with only a small amount of data locally. Experiments on a real-world data set illustrates that TLP integrated with SEWC technique can improve the prediction accuracy by up to 57.9% over the state of the art.

References

[1] J. Kang, X. Chen, D. Wu, Y. T. Xu, X. Liu, G. Dudek, T. Lee, and I. Park, "Hierarchical policy learning for hybrid communication load balancing," in *ICC*, pp. 1–6, IEEE, 2021.

[2] V. Sciancalepore, K. Samdanis, X. P. Costa, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5g network slicing resource utilization," in *INFOCOM*, pp. 1–9, IEEE, 2017.

[3] Y. Huang, L. Qian, A. Feng, N. Yu, and Y. Wu, "Short-term traffic prediction by two-level data driven model in 5g-enabled edge computing networks," *IEEE Access*, vol. 7, pp. 123981–123991, 2019.

[4] N. Zhang, P. Yang, J. Ren, D. Chen, L. Yu, and X. Shen, "Synergy of big data and 5g wireless networks: Opportunities, approaches, and challenges," *IEEE Wirel. Commun.*, vol. 25, no. 1, pp. 12–18, 2018.

[5] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1656–1659, 2018.

[6] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Trans. Mob. Comput.*, vol. 18, no. 9, pp. 2190–2202, 2019.

[7] F. Grijpink, A. Ménard, H. Sigurdsson, and N. Vucevic, "The road to 5g: The inevitable growth of infrastructure cost," *Article*, 2018. https://tinyurl.com/zhcsw5h4.

[8] X. Chen, Y. Li, and T. Zhang, "Reducing flash memory write traffic by exploiting a few mbs of capacitor-powered write buffer inside solid-state drives (ssds)," *IEEE Trans. Computers*, vol. 68, no. 3, pp. 426–439, 2019.

[9] Huawei, "Carrier data and storage architecture in the 5G era," *White Paper*, 2020. https://tinyurl.com/y5br4uf8.

[10] 5G Americas, "5G at the edge," *White Paper*, 2020. https://tinyurl.com/y6fspycd.

[11] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, 2019.

[12] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[13] C. M. Bishop, *Pattern recognition and machine learning, 5th Edition*. Information science and statistics, Springer, 2007.

[14] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?," *IEEE Trans. Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[15] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1790–1802, 2016.

[16] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *CoRR*, vol. abs/1612.00796, 2016.

[17] R. Pascanu and Y. Bengio, "Revisiting natural gradient for deep networks," in *ICLR*, 2014.

[18] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *MobiHoc*, pp. 231–240, ACM, 2018.

[19] F. Ashtiani, J. A. Salehi, and M. R. Aref, "Mobility modeling and analytical solution for spatial traffic distribution in wireless multimedia networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 10, pp. 1699–1709, 2003.

[20] K. Tutschku and P. Tran-Gia, "Spatial traffic estimation and characterization for mobile communication network design," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 5, pp. 804–811, 1998.

[21] R. Li, Z. Zhao, J. Zheng, C. Mei, Y. Cai, and H. Zhang, "The learning and prediction of application-level traffic data in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3899–3912, 2017.

[22] H. Zare Moayedi and M. A. Masnadi-Shirazi, "ARIMA model for network traffic prediction and anomaly detection," in *2008 ITCC*, pp. 1–6, 2008.

[23] S. Chinchali, P. Hu, T. Chu, M. Sharma, M. Bansal, R. Misra, M. Pavone, and S. Katti, "Cellular network traffic scheduling with deep reinforcement learning," in *AAAI*, pp. 766–774, AAAI Press, 2018.

[24] A. Chaudhuri, "Hierarchical support vector regression for qos prediction of network traffic data," in *IML*, pp. 15:1–15:6, ACM, 2017.

[25] X. Chen, Y. Jin, S. Qiang, W. Hu, and K. Jiang, "Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale," in *ICC*, pp. 3585–3591, IEEE, 2015.

[26] 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 *INFOCOM*, pp. 1–9, IEEE, 2017.