Federated Learning for Network Traffic Prediction

Sadananda Behera*, Saroj Kumar Panda†, Tania Panayiotou‡ and Georgios Ellinas‡
*Dept. of Electronics and Communication Eng., National Institute of Technology Rourkela, Sundargarh, Odisha, India
†Specialist Software Engineer, LTIMindtree, Bangalore, India
‡KIOS CoE and Dept. of Electrical and Computer Eng., University of Cyprus, Nicosia, Cyprus

***Abstract*—Network traffic prediction is imperative for effective network planning, decision-making, and optimization, leveraging the inherent predictability observed in traffic patterns. Although various machine learning models, such as deep neural networks (NNs) with recurrent units, including long-short-term memory (LSTM) NNs, have shown promise in this area, they are conventionally centralized trained, overlooking the potential benefits of distributed training techniques at the network edge. Exploiting advances in distributed training could provide certain advantages to the problem at hand, such as improved privacy preservation, reduced processing times, and lower bandwidth usage. Thus, this work proposes a federated learning (FL) framework for predicting network traffic, utilizing LSTMs for local model training. Simulation results validate the effectiveness of the proposed model, as this distributed implementation demonstrates high traffic prediction accuracy on real traffic traces, comparable to the traditional centralized approach, while safeguarding data privacy.**

***Index Terms*—Federated learning; Network Traffic Prediction; Machine Learning.**

I. Introduction

As the demand for network connectivity continues to soar with the emergence of new technologies like 6G networks, Internet of Things (IoT), and applications such as virtual reality, network service providers (NSPs) face the critical task of upgrading their infrastructures to cope with the ever-increasing traffic load, while at the same time minimizing capital and operational expenditures (OPEX/CAPEX). In this context, accurate prediction of traffic by the network planners plays a crucial role in decision making and network optimization.

In particular, accurate traffic prediction enables NSPs to strategically allocate resources, plan network expansions, and deploy traffic management solutions to meet evolving demands, while concurrently optimizing resource utilization and energy consumption, and minimizing costs [1]. Due to the unprecedented growth of the global Internet traffic, predicting network traffic poses persistent challenges, including ensuring accuracy in prediction, efficiently handling large volumes of data, and addressing concerns regarding data privacy/security, especially in regards to sensitive 6G emerging applications (e.g., Internet of drones, autonomous vehicles, augmented reality, etc.).

The advent of edge computing technologies allows overcoming such challenges, most notably when such technologies are associated with building intelligence at the edge [2], offering to the 6G connected applications ultra-low-latency, security, and reliability mechanisms. A key enabler of edge intelligence is distributed machine learning (ML), enabling effective data analysis at the edge.

One such paradigm is federated learning (FL), introduced by Google [3]–[5], enabling jointly training a model without sharing the data, as they remain distributed across many edge devices (i.e., through wireless or long-distance connections). Specifically, in FL several models are trained at the edge devices using distributed data without transferring sensitive information between the edge devices and a centralized location (i.e., cloud). Instead, these models are trained at the locations where the data reside [6], and after training inference can take place locally or centrally, depending on the application scenario. Hence, FL maintains data privacy and low latency, while utilizing data generated locally at the edge devices.

This work deals with the network traffic prediction problem, where decision making regarding network (re)optimization is based on the creation of (centralized) predictive traffic demand matrices. Even though decision making takes place centrally, learning can be achieved by means of distributed training to maintain, apart from data privacy, reduced processing times and low bandwidth requirements associated with the exchange of information between the edges and the central location. Specifically, this work advances the state-of-the-art by examining the efficiency of the FL model for network traffic prediction with the use of real-world traffic traces, and comparing it with traditional centralized training paradigms [1].

A. Related Work

The issue of network traffic prediction fundamentally concerns forecasting time-series data, where historical observations are utilized to anticipate future ones. Studies have demonstrated a degree of predictability in network traffic [7], indicating the feasibility of accurate predictions. In recent years, statistical linear models have played a significant role in tackling time-series prediction challenges, particularly in the domain of network traffic prediction. Among these models, the Auto Regressive Integrated Moving Average (ARIMA) and its various adaptations (such as Seasonal ARIMA, Fractional ARIMA, etc.) [8] have emerged as the predominant choices. However, due to their inherent assumption of linearity, such models frequently fail to capture the complexities of real time-series data [9], which typically exhibit non-linear characteristics [1].

As a result, machine learning has emerged as a powerful tool for accurately capturing network traffic behavior.

ISBN 978-3-903176-63-8 ©2024 IFIP
978-3-903176-63-8/24/$31.00 ©2024 IEEE

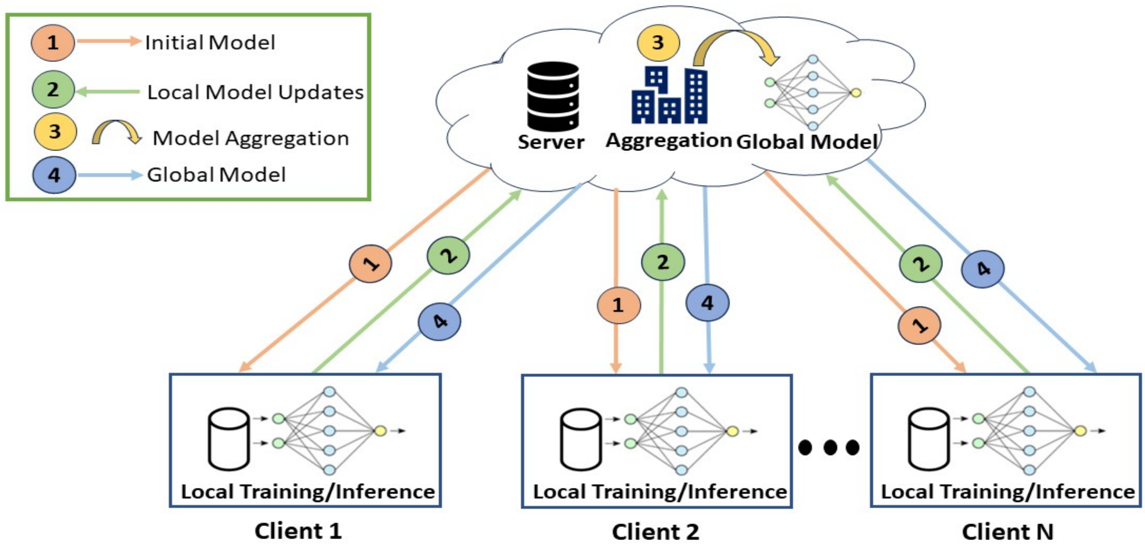

Fig. 1. Workflow of the proposed FL model for network traffic prediction.

Numerous studies comparing statistical linear models with non-linear ML methods have consistently demonstrated that the latter achieve higher accuracy in traffic prediction. Most notably, deep learning algorithms such as Recurrent Neural Networks (RNNs) [10], Long Short-Term Memory (LSTM) [11], and Gated Recurrent Unit (GRU) [12] have gained considerable attention for time-series problems, mainly due to their ability to capture the non-linear nature and long-term dependencies in network traffic.

However, the vast majority of existing studies is based on training a centralized model to subsequently create traffic prediction matrices for proactive network planning and/or real-time network operation [13], [14]. These works assume the exchange of data between the edges and the central controller, increasing communication overhead, without any provisioning regarding the privacy/security of data during transmission. Furthermore, such approaches increase computational overhead associated with centrally analyzing large volumes of data, while also increasing the requirements in storage capabilities at the central controller.

B. Contribution

To address these issues, in this work, network traffic analysis and inference is achieved by employing a FL framework. In this framework, local LSTM models are jointly trained by FL; that is, these models are locally trained to subsequently update the central LSTM model by periodically exchanging the updated model parameters between the central controller and the edge devices. The contributions of this work are as follows:

- A FL framework is proposed considering LSTM models tailored to predict network traffic and this framework is comprehensively evaluated utilizing a real-world dataset.
- A comparative analysis is presented between the FL framework and the traditional centralized training approach, particularly in terms of prediction accuracy.

The rest of the paper is organized as follows. The federated traffic modeling is presented in Section II, followed by the data processing and model training in Section III. Performance evaluation results are presented in Section IV, while Section V offers some concluding remarks and possible future research avenues.

II. Federated Traffic Modeling

Figure 1 illustrates the main workflow of the FL framework. In this framework a central controller (i.e., server/cloud) holds the global LSTM model which is trained indirectly by periodically receiving updated parameters from all the models that are locally trained (i.e., at the distributed clients). In a nutshell, the global model is randomly initialized, sending these initial parameters to all the distributed LSTM models (Step 1). Then, the global model is updated, by taking a weighted average of the resulting locally trained models (Steps 2 and 3). The newly formed global model is then transmitted back to the clients (Step 4), and the same process repeats (i.e., Steps 2,3,4) until either global model convergence is achieved or until a predetermined stopping criterion is met.

In particular, the FL objective is to:

$$minimize\ f(w) = \sum_{k=1}^{m} p_k F_k(w) \tag{1}$$

where F_k is the loss function to be locally optimized according to the stochastic gradient descent (SGD) algorithm at the kth client, for a total of m clients, and $p_k = \frac{n_k}{n}$, where n_k is the number of samples available locally for client k, and $n = \sum_k n_k$ is the total number of samples.

For the traffic prediction model proposed, it is assumed that each distributed client is used to store and analyze the traffic data of a single network node (i.e., the traffic load of a network node), with the global model eventually being

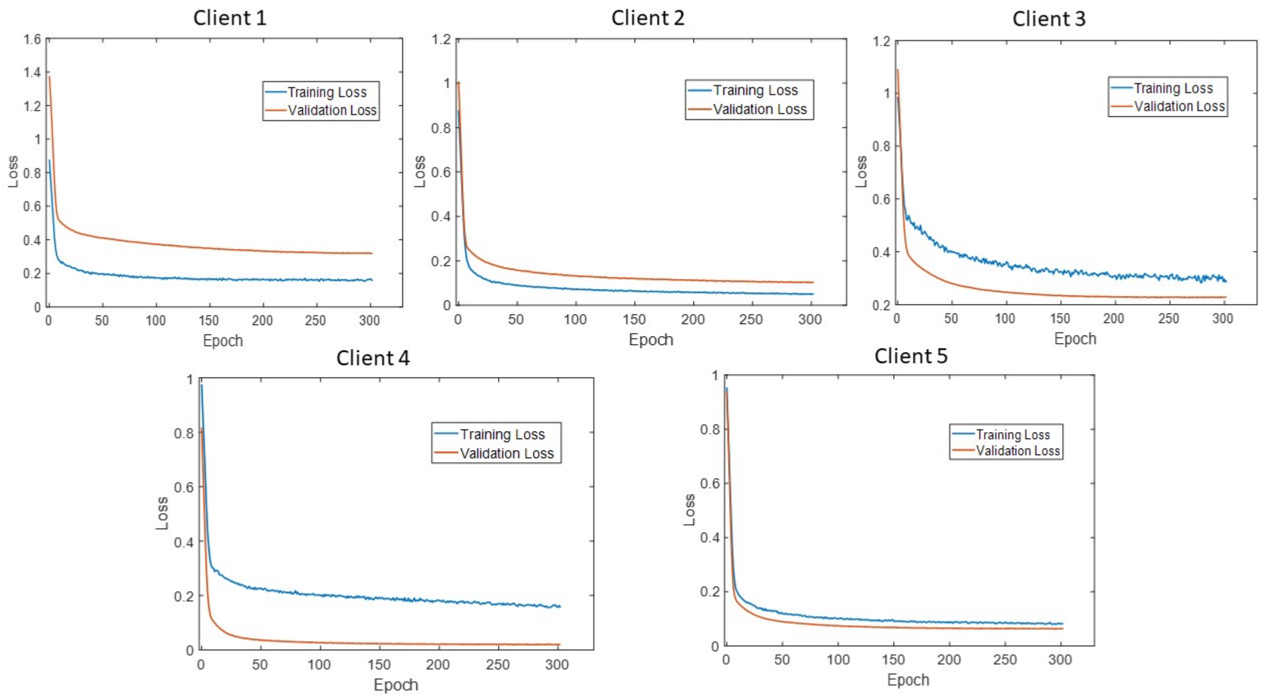

Fig. 2. MSE loss for all clients in the proposed FL setup.

capable of predicting the traffic values for all connected network nodes (i.e., to create a predictive traffic matrix).

As such, a local LSTM model for client/node k is trained to optimize $F_k(w)$, given the past and present traffic observations

$$x^{(k)'} = [x_{t-\kappa}^{(k)}, .., x_{t-1}^{(k)}, x_t^{(k)}],$$

to accurately predict the next traffic observation

$$x^{(k)} = x_{t+1}^{(k)},$$

where κ are the past traffic observations, and t is the present time instant. In this work, traffic observations $x^{(k)'}$ are collected every τ time units (e.g., every hour) for a total of κ past observation windows (e.g., hours), and the aim is to predict the next value of traffic at node k. Therefore, past and future traffic observations are τ time units apart, with $x^{(k)'}$ and $x^{(k)}$ being sequential in time.

Regarding the global LSTM model, this is capable of predicting the next traffic observation $x^{(k)}$, for all $k = 1, 2..., m$, with the central controller ultimately obtaining:

$$\chi = [x^{(1)}, x^{(2)}, ..., x^{(m)}],$$

by means of sequentially predicting the traffic load at node k, upon any time interval t. Assuming that the traffic load at a node k is destined for another node v, then χ can be used by the path computation element (PCE) of an orchestrator or controller to periodically (re)optimize and (re)configure the network according to future traffic information [15]. Note that more information about FL and LSTM models can be found in [4], [16].

III. Data Preparation and Model Training

To evaluate the performance of the traffic prediction FL framework, the publicly available real-world traffic traces of the 12-node, 30-link Abilene backbone network (http://sndlib.zib.de/home.action) are considered. This dataset provides bit-rate information (in Gbps) for every node pair in the form of traffic demand matrices, given every 5-minutes for a 6-month period. In this work, a dataset D_k is created for each node k in the network to represent the aggregated bit-rate to node k in 5-minute intervals. Given the aggregated bit rates, traffic patterns for each sequence in D_k (i.e., $[x_{t-\kappa}^{(k)}, \cdots, x_{t-1}^{(k)}, x_t^{(k)}, x_{t+1}^{(k)}]$) are created with $\kappa = 70$, following the sliding window approach.

To set up the FL framework, Python version 3.10.13 along with TensorFlow 2.10.0 is utilized on a computation system equipped with an Intel Core i5-10310U CPU running at 1.7 GHz and 16 GB of RAM. Further, for data preprocessing StandardScaler is employed to normalize the data.

For local model training and testing, each dataset D_k is partitioned in such a way that 70% of the patterns are used for training/validation and 30% for testing. In total, $k = 5$ clients are considered, with each client responsible for locally training a traffic prediction model for a predefined node, randomly selected among the nodes of the Abilene network. For each client, data sets D_k of different sizes are used to examine whether the FL framework is able to find an accurate global model under the conditions of data set imbalance. Specifically, each local model is trained considering $D_1 = 2799$, $D_2 = 9000$, $D_3 = 2000$, $D_4 = 7300$, and $D_5 = 5000$, with these patterns being sequential in time.

For local model training, the following parameters are considered: batch size=256, learning rate=10^{-4}, epochs=300, considering an LSTM model with 7 hidden layers trained to optimize the mean squared error (MSE) loss function (i.e., F_k is the MSE function).

Figure 2 illustrates FL-LSTM training/validation loss over the number of training epochs for all clients in the proposed

TABLE I
COMPARISON BETWEEN DISTRIBUTED AND CENTRALIZED SCHEMES.

Type	Model	MSE					Average MSE
		Client 1	Client 2	Client 3	Client 4	Client 5	
Centralized	LSTM	0.0658	0.0196	0.1716	0.0269	0.0518	0.0672
Distributed	FL-LSTM	0.0941	0.0216	0.1900	0.0277	0.0557	0.0778

FL setup. It is observed that both training and validation converge to an MSE loss that is close to zero, for all clients considered.

IV. COMPARATIVE ANALYSIS

To compare the efficiency of the FL-LSTM model against the conventional centralized training approach, the centralized approach considered an LSTM model consisting of 7 hidden layers, trained according to the hyperparameters previously mentioned for locally training the FL-LSTM models. This centralized LSTM model is trained considering a dataset D consisting of all the patterns in $D_1, ..., D_5$. As such, D consists of 25739 patterns, from which 70% are used for training/validation and 30% for testing. Figure 3 illustrates the centralized LSTM training/validation loss over the number of training epochs. This figure clearly illustrates that both training and validation loss converge to an MSE loss that approaches zero. It is noted that, compared to the FL scheme, convergence is now achieved in fewer epochs, as in the centralized scheme the LSTM model has direct access to all the traffic traces related to all clients.

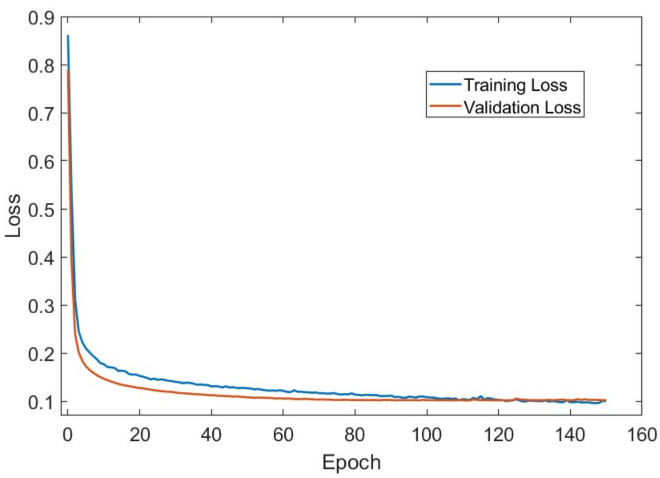

Fig. 3. MSE loss in centralized learning.

Furthermore, both distributed FL-LSTM and centralized LSTM models are compared according to prediction accuracy (i.e., per client and on the average). Specifically, the per client prediction accuracy is evaluated over the last 100 unseen sequences of each dataset D_k (i.e., sequences not used during training), and the average accuracy is evaluated over all clients. Comparative results are summarized in Table I, with the results demonstrating that the distributed FL-LSTM approach achieves accuracy levels that are close to the centralized LSTM model, both per client and for all clients (i.e., average) as well.

This outcome indicates the efficacy of the FL approach for network traffic prediction, in harnessing the collective knowledge from diverse client datasets without compromising predictive accuracy. The distributed nature of FL, although requiring constant synchronization and aggregation of models, allows each client to contribute its local knowledge, while aggregating insights from the entire network. This ultimately leads to a robust and accurate predictive model, preserving data privacy, and trained with lower bandwidth requirements as opposed to the alternative centralized approach.

V. CONCLUSIONS

This work proposes a distributed FL framework for network traffic prediction, that is shown to achieve sufficiently high accuracy over a real traffic dataset. This approach, unlike the centralized learning alternative, enhances data privacy protection, increases bandwidth efficiency, and offers greater scalability in terms of computing resources. As FL may face efficiency challenges when the data across the various clients are not independent and identically distributed (non-IID), the proposed framework will be extended to address this issue. Thus, an interesting future research direction is the consideration of a utility optimization function to ensure the fairness of the accuracies across all clients, without however causing degradation of the global model's accuracy.

ACKNOWLEDGEMENTS

This work has been supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 739551 (KIOS CoE - TEAMING) and from the Republic of Cyprus through the Deputy Ministry of Research, Innovation, and Digital Policy.

REFERENCES

[1] T. Panayiotou, M. Michalopoulou, and G. Ellinas, "Survey on machine learning for traffic-driven service provisioning in optical networks," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1412–1443, 2023.

[2] R. Gupta, D. Reebadiya, and S. Tanwar, "6G-enabled edge intelligence for ultra-reliable low latency applications: Vision and mission," *Computer Standards & Interfaces*, vol. 77, p. 103521, 2021.

[3] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv:1902.01046 [cs.LG]*, 2019.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[5] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv:1610.02527 [cs.LG]*, 2016.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[7] A. Sang and S.-q. Li, "A predictability analysis of network traffic," *Computer Networks*, vol. 39, no. 4, pp. 329–345, 2002.

[8] R. Agrawal and R. Adhikari, "An introductory study on time series modeling and forecasting," *arXiv:1302.6613 [cs.LG]*, 2013.

[9] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

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

[11] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[12] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. International Conference on Machine Learning (ICML)*, 2015, pp. 2342–2350.

[13] S. K. Singh, C.-Y. Liu, S. J. Ben Yoo, and R. Proietti, "Machine-learning-aided dynamic reconfiguration in optical DC/HPC networks," in *Proc. International Conference on Optical Network Design and Modeling (ONDM)*, 2022.

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

[15] T. Panayiotou and G. Ellinas, "Addressing traffic prediction uncertainty in multi-period planning optical networks," in *Proc. IEEE/OSA Optical Fiber Communications Conference (OFC)*, 2022, pp. 1–3.

[16] Y. Kim, Y. Jernite, D. Sontag, and A. Rush, "Character-aware neural language models," in *Proc. AAAI Conference on Artificial Intelligence*, 2016, pp. 2741–2749.