A Federated Learning Approach to Traffic Matrix Estimation using Super-resolution Techniques

Roberto Amoroso   Lorenzo Pappone   Flavio Esposito
Department of Computer Science
Saint Louis University, USA

***Abstract*— Network measurement and telemetry techniques are central to the management of modern computer networks. Traffic matrices estimation is a popular technique that supports several applications. Existing approaches use statistical methods which often make invalid assumptions about the structure of the traffic matrix. Data-driven methods, instead, leverage detailed information about the network topology that may be unavailable or impractical to collect. In this work, we propose a super-resolution technique for traffic matrix estimation that can infer fine-grained network traffic. In our experiment, we demonstrate that the proposed approach with high precision outperforms existing data interpolation techniques. We also expand our design by employing a federated learning model to address scalability and improve performance. Such a model increases the accuracy of our inference with respect to its centralized counterpart, significantly lowering the number of training epochs.**

***Index Terms*—traffic estimation, super resolution, deep learning, federated learning**

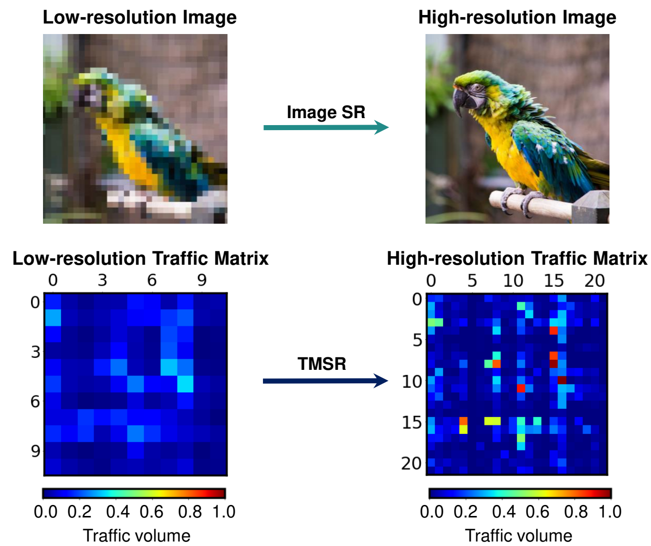

Fig. 1: Image Super-Resolution (ISR) problem (top) and proposed Traffic Matrix Super-Resolution (TMSR) technique (bottom). Generative methods such as Super-Resolution can be effectively used even in network measurements to cope with missing values and data amputations.

I. Introduction

The improvement of computer network efficiency, once low-hanging fruits have been harvested, necessarily passes through advanced analytic and non-trivial timely troubleshooting. Collecting the right amount of traffic at the right time is, however, a challenging operation, given the scale of today's networks and the "hidden" network spots. The analysis of network traffic data to obtain insights about the behavior of a distributed system is an important area of study. There are numerous methods for developing network models, but there are great difficulties and cost in the measurement of network traffic. Much of this difficulty comes from a large number of nodes at which traffic must be measured, as well as the amount of data that must be collected.

Many aspects of the Internet are characterized by decreasing visibility of important network properties, which is in tension with the Internet's role as critical infrastructure [1]. Researchers have sought out ways to infer network patterns from the least amount of data stored and collected [2], sometimes because such data is not available, since, before the (performance) problem arose, the frequency of measurements was kept low.

A goal of this work is to explore how a computer vision technique called super-resolution can help infer visibility during network monitoring and telemetry operations of a distributed system. In particular, we focus on the inference of fine-grain network traffic details only using aggregate measurements.

In our approach, we consider a Low Resolution (LR) traffic matrix, represented as a heat-map. Using a Super-Resolution (SR) technique, we can reconstruct a more accurate, i.e., High Resolution (HR) representation of the traffic we would have seen on a given subnetwork, with the smallest possible inference error. To do so, we first use a supervised learning approach, training a deep neural network with several high-resolution traffic matrices and their low-resolution counterparts. Then, we use the trained neural network to infer the high-resolution traffic matrix when only its low-resolution version is available.

The surprising result that we present is based on the intuition that despite the lack of clear data proximity features present in previous work using super-resolution, visual representations resembling traffic matrices still preserve key properties that we can exploit in a traffic inference problem. We apply super-resolution to network traffic matrices, proposing an algorithm that we call *Traffic Matrix Super-Resolution (TMSR)*. In Figure 1, we illustrate an example of super-resolution when applied to a standard image (example from the DIV2K dataset [3]) and a pixel representation of the traffic matrix of a real network [4].

Federated learning to scale the training process and to improve traffic inference performance. Converting traffic matrices to images may be practical in some cases; however, in wide-area networks, large data transfers, and privacy data sharing constraints may render this approach impractical. For example, some service or content distribution network providers have customers that are unwilling or incapable of sharing their network measurement data. In those cases, moving the deep learning model where the data is located may be preferable when feasible. While existing work has shown that federation alone is insufficient to guarantee privacy [5] to improve our traffic inference performance and accelerate the training process, (and so not to improve privacy) we exploit federated learning [6]. Aside from allowing a neural network to be trained by sending copies of it where data resides (as opposed to sending the data to the model), federated learning techniques are also valuable as they avoid the transfer of large amounts of traffic data collected by various measurement sites. By sending only model updates rather than the entire dataset, federated learning may reduce training communication rounds by orders of magnitude compared to centralized learning. We analyze the benefits of training our traffic matrix super-resolution model via federated learning. Our evaluation using real traces confirms the effectiveness of our approach to improve the accuracy performance compared to super-resolution models trained in a centralized way. Surprisingly, we also obtained better traffic inference accuracy results with a significantly lower number of training epochs.

The rest of the paper is organized as follows: Section II discusses the related work; in Section III we formulate the traffic matrix super-resolution (TMSR) problem, starting from the traffic matrix inference problem. In Section IV we detail our neural network model and its federated learning counterpart. In Section V we describe our data preparation methodology, that is, the process of collecting, analyzing, and augmenting the traffic matrices generated from the traffic measurement. In Section **??** we describe our experimental setting. In Section VI we discuss our evaluation results, showing the benefits obtained through the data preparation process, and the results achieved after training our traffic matrices super-resolution model both in a centralized and federated fashion. Finally, in Section VII we present our conclusion.

II. Related Work

In this section, we cite a few representative solutions on key areas related to our project: traffic inference methods, super-resolution techniques, and federated learning.

Traffic Inference and Telemetry. Being able to infer which are the routes that pass through the network of an operator is a complex but profitable task, with applications to traffic engineering, performance analysis, capacity planning, traffic loads change and its causes, network security, and business intelligence.

Related work in this area can be divided into two main categories: those that consider a TM a purely-spacial concept [7], [8] and those for which a TM is a time series of TMs [9], [10]. The latter approach implies a strong correlation between TMs over time. These data are low effective rank, as shown in [9], i.e., there are strong correlations between columns (or rows), such that a measured TM can be approximated by a matrix having a relatively small rank. The results of the methods using these low-rank data show a strong dependence on this temporal correlation. However, spatiotemporal compressive sensing cannot work with the highly non-linear relationships between high- and low-resolution network traffic samples, as they expect that linear relationships subsist between sparse traffic and inference matrices. Similar to the approach adopted by [11], in our work, a TM element to be estimated is never visible over time, and so past history is less useful. However, many of the proposed traffic inference methods require the collection and analysis of fine-grained traffic matrices, as well as detailed information on the topology and structure of the network under consideration. Our method is able to overcome these limitations. During the training phase of our super-resolution algorithm, we learn the mapping between low-resolution matrices and their high-resolution counterpart. This training procedure allows us to infer information about network topology, which is needed for the super-resolution task and hence for the traffic-matrix estimation.

Super-resolution Techniques. Super-resolution is a generative process that represents a fundamental tool for a wide range of real-world applications, such as medical imaging, security, surveillance, and other computer vision tasks. Super-resolution approaches learn mapping functions from low-resolution (LR) images to high-resolution (HR) images from a large number of examples.

The first super-resolution algorithm was proposed by Dong et al. [12], overcoming the state-of-art image reconstruction accuracy.

The architecture on which the proposed EDTMSR model is based is the *Enhanced Deep Super-Resolution* [13] (EDSR) network, which removes unnecessary modules from conventional ResNet architecture and employs residual scaling techniques to train large models more stably.

Nonetheless, we are not the first to apply SR to computer networks. ZipNet-GAN [14] used mobile network traffic maps on a city scale and applied super-resolution to learn the telecommunication power patterns from fewer measurement probes. ZipNet-GAN utilized super-resolution in networking for the creation of physical mobile network matrices, characterized by the presence of a spatial relationship between the cells of the matrices themselves.

Our work, on the other hand, focuses on the traffic matrices generated within any network. Given the nature of Internet traffic flows, these traffic matrices *lack physical location association*. We created a *location-independent* network traffic inference method. Furthermore, we trained the proposed EDTMSR model both in a centralized and distributed way, with federated learning. Using such distribute learning paradigm, we are able to show that by following specific data distribution strategies among the various clients, the model trained outperforms the results obtained with traditional centralized learning.

We conduct experiments to show how our model trained via federated learning is robust to the varying participation of federated clients in the training process, and that the model trained with federated learning is able to outperform the results obtainable with the traditional centralized training methods.

III. Problem Definition

In this section, we formulate the Traffic Matrix Inference and TMSR problems. The former problem can be described as the task of inferring traffic between all origin-destination pairs in a given network, despite a lack of a complete dataset. The related TMSR problem instead entails applying super-resolution methods on traffic matrices to infer fine-grained network traffic data from sets of coarse-grained network measurements. While training our super-resolution model, we learn how to map coarse-grain traffic matrices to their high-resolution counterparts, grasp information about network topology, and improve the accuracy of traffic estimation.

Traffic Matrix Inference: Notations and Setup. The element $M(i,j)$ of a TM describes the volume of traffic, expressed in bytes, packets, or flows, measured between source i and destination j. We express a TM as a 2-dimensional array $M \in \mathbb{R}^N \times \mathbb{R}^N$, where N is the number of nodes in the network. Usually with $M(i,j;t)$ we denote the traffic from node i to j, averaged over the time interval $[t, t+\Delta t)$. In this work, we consider traffic snapshots omitting the time dependency. We consider router-level *ingress-egress* TMs, in which $M(i,j)$ represents the traffic from router i to router j, hereinafter also referred to as $TM_{i,j}$.

Traffic Matrix Super-Resolution. The objective of the Traffic Matrix Super-Resolution (TMSR) problem is to infer fine-grained network traffic data, using as a starting point sets of coarse-grained measurements collected in the network. The low-resolution (LR) traffic matrix TM^{LR} can be defined as the output of a degradation process:

$$TM^{LR} = D(TM^{HR}; \delta), \tag{1}$$

where D is a degradation mapping function, TM^{HR} is the High-Resolution (HR) traffic matrix and δ denotes the parameters of the degradation process, e.g., the noise or scale factor. Generally, in a super-resolution (SR) problem setting, D and δ are unknown and the LR traffic matrix represents the only input. The goal is to recover an approximation $\widehat{TM}^{HR}$ of the ground truth HR traffic matrix, denoted as TM^{HR}, starting from its LR version TM^{LR}:

$$\widehat{TM}^{HR} = F(TM^{LR}; \theta), \tag{2}$$

where F is the super-resolution model and θ represents its parameter vector. The recovered HR traffic matrix $\widehat{TM}^{HR}$ is also denoted as *super-resolved traffic matrix* TM^{SR}. We can directly model the degradation function as a single downsampling operation:

$$D(TM^{HR}; \delta) = (TM^{HR}) \downarrow_s, \ s \subset \delta, \tag{3}$$

where $\downarrow_s$ is the downsampling operation associated with the scale factor s. The most commonly used downsampling operation, which is also the one we use in this paper, is the *bicubic interpolation* [15]. The goal of interpolation is to estimate values at unknown points using known data. In particular, bicubic interpolation is a technique commonly used in image processing to perform *image resizing*, which consists of increasing or decreasing the resolution, i.e., the total number of pixels of an image, obtaining an HR image from its LR version, or vice-versa.

The Traffic Matrix Super-Resolution (TMSR) problem is hence modeled by the following *optimization*:

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmin}} \, \mathcal{L}(\widehat{TM}^{HR}, TM^{HR}) + \lambda \Phi(\theta), \tag{4}$$

where $\mathcal{L}(\widehat{TM}^{HR}, TM^{HR})$ is the loss function between the predicted HR traffic matrix $\widehat{TM}^{HR}$ and the ground truth traffic matrix TM^{HR}, $\Phi(\theta)$ is a regularization term and λ is the trade-off parameter.

We adopt the mean absolute error (MAE) loss function to measure the quality of our super-resolution traffic matrix inference and to train our proposed EDTMSR neural network model, which architecture is detailed in the following section.

IV. Our Solution: Super-Resolution for Traffic Matrix Inference

In this section, we describe the *Enhanced Deep Traffic Matrix Super-Resolution Network (EDTMSR)*, the deep learning model that we propose to solve the TMSR problem. We also explain the training process of EDTMSR through federated learning.

In general, the architecture of a Convolutional Neural Network is defined by the number of layers, also called *depth*, and the number of feature channels for each convolutional layer, also called *width* or F. We developed our EDSR variant for traffic matrices by setting $F = 64$, whereas the depth depends on the number of residual blocks RB stacked, which we set to 8, and on the value of the scale factor s considered.

In particular, our approach is based on the EDSR [13] deep learning architecture, to carry out the super-resolution of traffic matrices, defining what we called EDTMSR.

A. Federated Learning for Performance and Partial Traffic Visibility

In this section, we describe our design choice of training our EDTMSR with Federated Learning (FL). Training a neural network with a federated approach allows each client to perform traffic measurements according to different criteria and with arbitrary sampling rates. Moreover, as compared to a centralized approach we obtain savings in terms of communication costs and in the number of epochs necessary to complete the training process (Sec. VI-C).

Provided that we train our EDTMSR with a federated learning approach, we are to decide how to divide our dataset among clients participating in the training process. We explore two possible federated training configurations, and we will refer to them as *Federated for Performance* and *Federated for Partial Traffic Visibility*, respectively. As the name suggests, the first has as its main objective the performance improvement of the SR model accuracy. The second has the objective of providing a client with visibility on the network traffic while still maintaining acceptable SR accuracy. Both solutions use *FedAvg* [6] as a federated learning algorithm to train a more robust and performing model. The following sections describe in detail the characteristics of both configurations.

Federated Learning to Boost Performance. In this configuration, we assume that all clients participating in the federated training process can observe and measure the traffic of all nodes of the network. We distributed the traffic matrices equally among the various clients, but at random. A consequence of such random splitting of traffic matrices is that the distribution of traffic patterns on each client training set may be unbalanced. In the evaluation section, we show that such unbalance distribution does not hinder the performance of our traffic inference. One of the characteristics that distinguish the federated learning technique is the variability of clients' participation in the training process. During each training round, only a random subset of the clients participate.

Federated Learning with Partial Traffic Visibility. In this configuration, we remove the assumption that all clients participating in the federated training process can observe and measure the traffic of all nodes of the network. This is because clients belonging to different network partitions or regions may be unable to obtain such information but still be interested in estimating traffic volumes. With this configuration, we aim at assessing to what extent the distributed training approach enables visibility gains into the network traffic of other clients, despite the lack of global knowledge on traffic matrices. By limiting visibility to only a sub-portion of the entire network, each federated client will have to monitor, aggregate, and process a smaller amount of data. To reproduce the concept of partial visibility of network traffic, we have divided each traffic matrix into four non-overlapping, or partially overlapping sub-matrices.

V. Data Processing and Methodology

In this section, we describe the process of collecting, processing, and augmenting the traffic matrices that make up the real-world dataset we use in our experiments.

A. Dataset Description and Pre-Processing

In this section, we describe all the details of the traffic dataset we used to train our EDTMSR architecture. Deep Learning (DL) techniques require extensive and representative data to build an effective neural network model, characterized by both high performances in terms of training time and accuracy in the reconstruction of TMs.

In this work, we utilized a dataset collected from GEANT [4], a research and educational European network. This anonymized dataset consist of $10,772$ traffic matrices built using Interior gateway protocol (IGP) routing information, NetFlow data collected overall edge links, and Border Gateway Protocol (BGP) routing information of the GÉANT network sampled every 15 minutes for 4 months. The network in which the traffic matrices were collected is formed by 23 nodes. The value (i, j) of a traffic matrix corresponds to the traffic going from node i to node j, expressed in *Kbit*. To increase the size of the dataset, we use a data augmentation technique described in the following section.

B. Data Augmentation and Downsampling

In this section, we detail how the datasets used in the training, validation, and testing phase of the proposed EDTMSR model were constructed, with an in-depth analysis of the process of generating low-resolution traffic matrices starting from their high-resolution version.

Starting from the original $10,772$ traffic matrices of the GÉANT dataset [4], we decided to split the data in train/valid/test according to the commonly used ratio $80/10/10$.

We increase the size of the training set by applying a sliding window over each traffic matrix. Unless explicitly indicated otherwise, the data and plots shown refer to traffic matrices obtained through a sliding window with a size of 18×18, which proved to be sufficient to produce enough training samples. This procedure yields 36 training samples from each traffic matrix, increasing the overall size to $310,248$ TMs. These samples serve as the HR ground truth, from which we extract the LR matrices that will be utilized to train our Super-Resolution neural network.

1) Low-Resolution Traffic Matrices via Bicubic Downsampling: Using well-known downgrade functions, we can obtain LR traffic matrices from their HR version, creating a large training dataset used by our self-supervised learning algorithm. To generate LR traffic matrices in the TMs processing pipelines, we used bicubic downsampling [15], a known downgrade function. We also used various scale factors s to analyze the performance of our model as the resolution of the traffic matrices varies. The size of the TMs of our datasets, having side 23×23, has limited the range of possible testable scale factors. More specifically, we used scale factors $s = \{2, 3, 6\}$, hereinafter also referred to as $\times 2$, $\times 3$, and $\times 6$, respectively.

VI. Evaluation Results

In this section, we evaluate the TM inference using federated learning. We then dissect the benefits of our deep learning model over an SR obtained with algebraic inference. We evaluate the performance obtained after training our traffic matrices super-resolution model both in a centralized way and using federated learning. All results shown are obtained averaging on our test set consisting of 1077 TMs, unless otherwise stated.

A. Evaluating TM Inference with Federated Learning

Depending on the federated training configuration used to train our EDTMSR model (defined in Section IV-A), we use the notation *EDTMSR-PERF* and *EDTMSR-PTV*, to indicate the model trained in the FL for Performance and FL for Partial Traffic Visibility configuration, respectively. Each

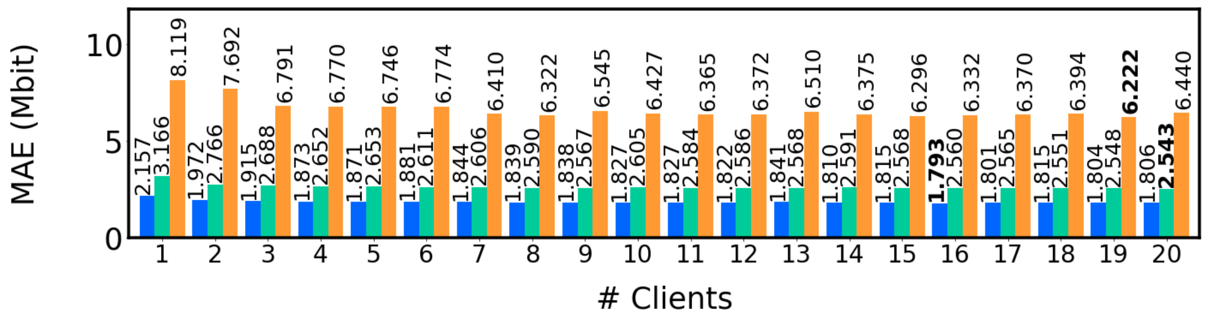

Fig. 2: Traffic Matrix inference accuracy comparison of the proposed EDTMSR trained both centralized (only 1 client) and federated (from 2 to 20 clients), in terms of mean absolute error (MAE). We consider three scale factors instances (different color bars).

configuration is characterized by a "key parameter" of the federated training process that determines the characteristics of the distributed environment in which our EDTMSR model is trained. In the case of FL for Performance, the key parameter is the number NC of clients participating in the distributed training process. Instead, in FL for Partial Traffic Visibility, the key parameter indicates how sub-matrices are obtained from the original TMs and how these are distributed among the federated clients. For both configurations, we conducted experiments aimed at determining the EDTMSR-PERF and EDTMSR-PTV models that produced the best results for all the considered evaluation metrics.

1) TM Inference Accuracy with EDTMSR-PERF: The purpose of this experiment is to determine the optimal number of federated clients that we should use to distribute the training of our EDTMSR model to obtain the best results in terms of inference accuracy for our TMs.

We have distributed the training of our model to several clients ranging from a minimum of 2 to a maximum of 20, for a total of 19 EDTMSR-PERF models. We denote with EDTMSR-PERF-NC, with $NC \in [02, 20]$, the model that has been trained by NC federated clients. For each model, we trained 3 instances associated with the scale factors $\times 2$, $\times 3$, and $\times 6$. As we can see from Figures 2, the models trained by 16, 19, and 20 clients are those that have produced the best results for the scale factors $\times 2$, $\times 6$, and $\times 3$, respectively.

Figure 2 also shows that, regardless of the number of clients, the federated model always outperforms the performance of the centralized model. We number the take-home messages of our evaluation. *(1) We found that the performance improvement decreases as the number of federated clients increases, suggesting a decreasing marginal gain.* In particular, increasing from 1 to 10 clients, there is a continuous decrease in the value of the MAE, with a significant improvement of 20.8%. Beyond 10 clients, the value of the MAE remains almost stable, and in some cases, it even slightly returns to increase.

B. Algebraic Inference vs Deep Learning

In this experiment set, we assess if the TMSR problem can be solved, with acceptable accuracy, without using a

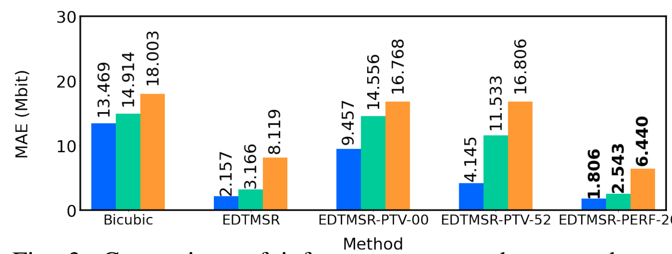

Fig. 3: Comparison of inference accuracy between the existing Bicubic interpolation SR technique and the proposed EDTMSR in its centralized, federated for partial traffic visibility (EDTMSR-PTV) and federated for performance (EDTMSR-PERF) variants, in terms of MAE.

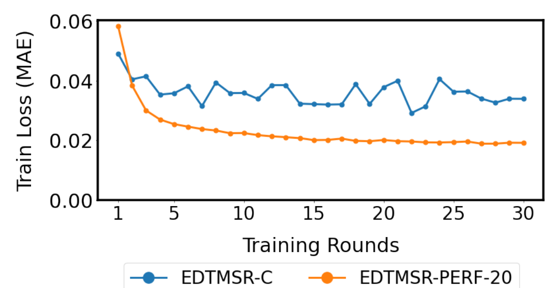

Fig. 4: Trend of the loss function during the training phase with a scale factor x6. The proposed EDTMSR in its centralized version (EDTMSR-C) is compared with its federated counterpart distributed over 20 clients (EDTMSR-PERF-20).

deep learning architecture. To do so, we compare the proposed centralized and federated architectures with an algebraic super-resolution method: the bicubic interpolation. In particular, we compare the TM inference accuracy performance of the bicubic interpolation algorithm, the centralized EDTMSR, EDTMSR-PTV-00, EDTMSR-PTV-50, and EDTMSR-PERF-20 models. These models turned out to be the most performant for the federated learning configurations we have investigated. We have chosen the EDTMSR-PTV models with 0% overlap and 52% overlap to analyze two extreme cases. On the one hand, there is a complete absence of visibility on the internal traffic of other clients' subnets, since there is traffic visibility overlap across clients. On the other hand, each client has at most a 50% visibility on the internal traffic flows of the other clients' subnets.

Figure 3 shows the values of the evaluation metrics measured on the aforementioned models. As we can see, the centralized version of the EDTMSR model overcomes the performance of the bicubic interpolation. We obtain an improvement of MAE of 84%.

These results, although already significant, are further improved by the federated approach, which allows us to obtain an overall improvement of 86.6% of MAE compared to the bicubic method. The EDTMSR-PERF-20 model is the one that obtained the smallest MAE value among all the models compared and for all the scale factors considered. We can hence conclude that *(8) both EDTMSR-PTV models outperform the performance of the bicubic method and demonstrate the validity of the proposed architecture even in contexts of partial visibility on the traffic flows inside the network.*

C. Scale Factor Impact on SR Accuracy

In this experiment, we compare the statistical distribution of the evaluation metric values measured on each of the models. In particular, Figure 4 shows the comparison between the trend of the loss function, a measure of the model accuracy, during the training phase of the EDTMSR and EDTMSR-PERF-20 models with a scale factor $\times 6$. Results show that the federated model outperforms the centralized counterpart after only two training rounds. Moreover, note how the trend of the loss function of the federated model is much smoother.

(12) The federated model achieves the same performance as the centralized model after $\mathbf{80\%}$ *fewer training rounds/epochs. (13) Finally, our results show that training a deep learning architecture to solve the TMSR problem yields a significant improvement compared to the bicubic method both in terms of prediction error and structural fidelity of the inference of fine-grained traffic patterns.*

VII. Conclusion

In this work we proposed the use of a computer vision technique known as super-resolution to infer network traffic volumes with fine granularity. We presented the design of the Enhanced Deep Traffic Matrix Super-Resolution Network (EDTMSR), a neural network architecture built to perform super-resolution on traffic matrices. To address scalability and boost performance, we also expanded our design by employing federated learning and measuring the performance of a few representative distributed learning policies.

Our evaluation showed that the proposed method reduces the prediction error (MAE) of existing interpolation super-resolution approaches by 87%, and achieves up to 69% higher fidelity (PSNR) and $3.4\times$ greater structural similarity (SSIM). Furthermore, the federated learning approach allowed us to achieve the same results as the centralized model but performed 80% fewer model training rounds.

VIII. Acknowledgment

This work has been supported by the National Science Foundation (NSF) Awards #1836906, #1908574, and the Comcast Innovation Award.

References

[1] K. Claffy, D. Clark, J. Heidemann, F. Bustamante, M. Jonker, A. Schulman, and E. Zegura, "Workshop on overcoming measurement barriers to internet research (wombir 2021) final report," *SIGCOMM Comput. Commun. Rev.*, vol. 51, no. 3, p. 33–40, jul 2021. [Online]. Available: https://doi.org/10.1145/3477482.3477489

[2] M. Yu, L. Jose, and R. Miao, "Software defined traffic measurement with opensketch," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. Lombard, IL: USENIX Association, Apr. 2013, pp. 29–42. [Online]. Available: https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/yu

[3] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 114–125.

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

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[7] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 161–174, 2002.

[8] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach," *IEEE/ACM Transactions on networking*, vol. 13, no. 5, pp. 947–960, 2005.

[9] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proceedings of the joint international conference on Measurement and modeling of computer systems*, 2004, pp. 61–72.

[10] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, 2009, pp. 267–278.

[11] V. Bharti, P. Kankar, L. Setia, G. Gürsun, A. Lakhina, and M. Crovella, "Inferring invisible traffic," in *Proceedings of the 6th International COnference*, 2010, pp. 1–12.

[12] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, 2014, pp. 184–199.

[13] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1132–1140.

[14] C. Zhang, X. Ouyang, and P. Patras, "ZipNet-GAN: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network," in *CoNEXT 2017 - Proceedings of the 2017 13th International Conference on emerging Networking EXperiments and Technologies*, vol. 17, 2017, pp. 363–375.

[15] R. E. Carlson and F. N. Fritsch, "Monotone piecewise bicubic interpolation," *SIAM journal on numerical analysis*, vol. 22, no. 2, pp. 386–400, 1985.