Received 22 January 2021; revised 18 February 2021; accepted 18 February 2021. Date of publication 4 March 2021; date of current version 17 March 2021.

Digital Object Identifier 10.1109/OJCOMS.2021.3062636

Learning Based Methods for Traffic Matrix Estimation From Link Measurements

SHENGHE XU[1], MURALI KODIALAM[2] (Member, IEEE), T. V. LAKSHMAN[2] (Fellow, IEEE), AND SHIVENDRA S. PANWAR[1] (Fellow, IEEE)

[1]Department of ECE, New York University Tandon School of Engineering, Brooklyn, NY 11201, USA

[2]Network Systems Research, Nokia Bell Labs, Holmdel, NJ 07733, USA

CORRESPONDING AUTHOR: S. XU (e-mail: shenghexu@nyu.edu)

ABSTRACT Network traffic matrix (TM) is a critical input for capacity planning, anomaly detection and many other network management related tasks. The TMs are often computed from link load measurements. The TM estimation problem is the determination of the TM from link load measurements. The relationship between the link loads and the TM that generated the link loads can be modeled as an under-determined linear system and has multiple feasible solutions. Therefore, prior knowledge of the traffic demand pattern has to be used in order to find a potentially feasible TM. In this paper, we consider the TM estimation problem with limited prior information. Unlike previous methods that require past measurements of complete TMs, which are hard to obtain or protected by regulations, our method works even if only the distribution of TMs is known. We develop an iterative projection based algorithm to solve this problem. If large number of past TMs can be measured, we propose a Generative Adversarial Network (GAN) based approach for solving the problem. We compare the strengths of the two approaches and evaluate their performance for several networks using varying amounts of past data.

INDEX TERMS Traffic matrix, estimation, machine learning, generative adversarial networks.

I. INTRODUCTION

THE AMOUNT of traffic incident on a network is usually captured in the form of a traffic matrix (TM). A TM consists of the amount of traffic between each node pair in a network. Knowledge of the TM is essential to solving networking problems including link capacity planning, routing path design and network anomaly detection [1]–[3]. However, it is not easy for a network operator to directly measure the point to point traffic in a network. The most commonly used method to estimate the TM is to use link load measurements to infer the TM. The amount of traffic on a link is relatively easy to measure or estimate using traffic monitoring mechanisms like NetFlow.

However, previous methods for TM estimation assume that a certain amount of past TMs are available for algorithm design, model training or estimation. Nowadays, with larger network sizes and increasing privacy protection rules [4], this assumption is no longer valid. In fact, even papers in 2019 and 2020 [5]–[9] are still using the Abilene dataset [10] and GÉANT dataset [11], which were created over a decade ago. Technical issues and legal regulations make it difficult to create TM datasets from current networks. As stated in [11], even for the 23 node GÉANT dataset, hundreds of gigabytes of Netflow data had to be stored to derive the TMs. For large scale networks with hundreds or thousands of end points nowadays, up to thousands of terabytes of data need to be stored to generate sufficient data for algorithm design and model training. Even if the full TMs can be measured, it is difficult to share the TMs. Several U.S. Federal laws prohibit or restrict network monitoring and the sharing of records of network activity [12]. In the research community, there has also been ethical concerns over publications regarding traffic data [13]. In addition, many European countries have strict regulations on user privacy protection, which prohibits moving data between certain countries [4], [14]. For large cross-country networks, this makes it even harder to obtain a full TM. In this paper, we consider the TM estimation problems in the practical setting where full TMs cannot be used for estimation purposes. We consider two specific cases, in the first case, full TMs cannot be measured, only the distribution of the TMs is known. We propose a cyclic projection based method to utilize the distribution constraint. In the second case, a few TMs can be measured, but they cannot be directly used for the estimation of current TM. We proposed to use a Generative Adversarial Network (GAN) based method to improve estimation accuracy, while protecting users' privacy.

In a network with n nodes, the size of the TM is $O(n^2)$, whereas the number of links in the network is typically $O(n)$. Therefore, the problem of determining a TM from link load measurements is deriving a solution to an under-determined system of linear equations. This system has an infinite number of solutions even if we restrict the solutions to be non-negative. Therefore some additional information has to be used to restrict the solution space to this system and obtain a single TM. This additional information or extra knowledge typically takes the form of assuming some spatial or temporal correlations about the entries in the TMs. We give two examples of these assumptions, one spatial and one temporal.

- *Gravity Models* where a weight is associated with each node in the network and the amount of traffic between two nodes is proportional to the product of the weights.
- *Proportional Splitting* where it is assumed that the traffic from a given node is split proportionally to different destinations and these proportions are time invariant. Data can be collected across n time periods and jointly used to solve for the proportions.

Another class of assumptions is traffic sparsity in certain transform domain. See [10], [15]–[24] for examples of different assumptions for deriving a unique TM from link measurements.

In this paper, we consider constraints on the TM estimation problem that arises from TM observations. If the operator has measured a few TMs on the network of interest or some similar network, then it is reasonable to restrict the estimated TM to have properties similar to the measured or observed TMs.

Distribution Constraint: It has been observed in practice [25]–[27] that the point-to-point traffic in a network is generally not uniform. There are some large traffic source-destination pairs, and several small traffic source-destination pairs. Modeling the demand size variation as a distribution, the objective of the TM estimation problem is to determine a TM that achieves the measured link loads and follows the given distribution. In practice, the distribution model can be selected using hypothesis testing [28]. The value of the parameters can be obtained from maximum likelihood estimations [29].

One of the advantages of the distribution constraint is that it can be used when limited or partial measurements of TMs is available. Since we do not require full TMs to derive a distribution, measurements of demands between different source-destination pairs from different time can be combined to fit a distribution. In Section VII we show that fitting a distribution require much less data, reducing the cost and complexity of TM measurements. In addition, fitting a distribution only requires the value of the demands, the demands can be shuffled and anonymized before they are shared, helping the network operator better protect users' privacy and meet regulations.

Similarity Constraint: More generally, if we are given a previously observed set of TMs, the objective of the TM estimation problem is to derive a TM that achieves the given link loads and is "similar" to the previously observed TMs. In this case, it is possible to capture more complex spatial correlations between different source-destination pairs in the TM. The problem of determining a solution to an under-determined linear system has been studied in the signal processing literature [30]. One way of getting unique recovery is to assume sparsity and the objective is to determine a solution to the linear system with the minimum number of non-zero components or a solution that minimizes the L_1-norm. More recently, there has been work to construct a solution to a linear system that is close to the range space of a generative model[31]. The generative model can be specified by a Generative Adversarial Network (GAN) [32]–[34] or a Variable Autoencoder [35]. We make use of these new approaches to derive solutions to the TM estimation problem. In the mean time, similar to other applications of GANs for private data sharing [36], [37], training and sharing a GAN can avoid sharing of the original TMs, meeting the privacy protection regulations.

A. OUR CONTRIBUTIONS

In this paper we focus on the practical setting where past TMs can not be directly used for TM estimation, we propose two methods to solve the TM estimation problem.

- In the case where only the empirical distribution of the TMs is available, we develop an iterative projection based method to find a solution to the system $\boldsymbol{Ax} = \boldsymbol{b}$ where the solution $\boldsymbol{x}$ satisfies the distribution. To our knowledge, this is the first work that determines the solution of an under-determined system where the solution has to satisfy a distribution constraint.
- For the case where there are many prior TMs, but they can not be directly shared due to regulations, we develop a GAN based approach that first trains a GAN to learn the characteristics of these TMs. Then the GAN can be shared to derive a solution to the system that is "similar" to the previously observed TMs.

The rest of the paper is organized as follows. Section II briefly summarizes related work. In Section III we formulate the problem. The projection based method is proposed in Section IV. In Section V and Section VI we introduce the GAN based TM estimation method. Experiment setup is included in Section VII. The performance of the methods is evaluated in Section VIII. In Section IX we draw the conclusions and propose directions for future work.

II. RELATED WORK

Traffic matrix estimation, also called network tomography, is an extremely important first step for solving several network design and network management problems. This problem has been studied extensively under different assumptions about traffic demand information and estimation.

An example of research exploiting temporal correlation to estimate the TM is [15], where it is assumed that the traffic demands over time follow Poisson distribution and this information is used to derive a TM.

Several papers [10], [16]–[18] consider using spatial characteristics of the TMs to improve the recovery results. Zhang *et al.* [10] proposed gravity models to solve the problem of network tomography. In [18], the authors proposed an information-theoretic method for network tomography.

Later works [19], [23], [24] consider using both spatial and temporal information for better recovery results. A compressive sensing based method called Sparsity Regularized Singular Value Decomposition (SRSVD) was introduced in [19]. In addition to link measurements, measurements of demands between some of the origins and destinations are assumed to be available. Measurements of previous TMs are also used to improve estimation accuracy. The SRSVD utilizes sparsity of TMs in transform domain for recovery. There are also other methods [20]–[22] that utilize low rank or sparse characteristics of TMs for TM completion. Instead of forming sampled TMs into a 2D matrix, [23], [24] proposed to form TMs directly into 3D tensors. In this way the periodicity of certain traffic demand features can also be utilized by tensor completion methods for traffic demand estimation.

More recently, deep neural networks (DNNs) [38], [39] have achieved some of the state-of-the-art results in areas including image inpainting [40] and image compressive sensing [31]. Since TM estimation is also a similar problem, neural networks have also been used in this area. In [41], [42] the authors also proposed to use neural networks for TM completion.

All of these previous methods rely on temporal or spatial correlation for better estimation results. However, to utilize the temporal or spatial correlation, it is essential to have past measurements of the TMs. As stated in Section I, for current networks it is often impossible to obtain these measurements. In this paper we only assume that the estimation results should be similar to the previous TMs or follow the same distribution. Our method does not require direct access to the TMs. To our knowledge, this problem has not been addressed in the literature. Note that the other methods mentioned above will not work if only a distribution constraint is given.

III. PROBLEM DEFINITION

Assume that the network is represented as a directed capacitated graph $G = (V, E)$ with n nodes V and m directed links E. Assume that we are given the set of link weights $\mathbf{w} = (w(e_1), w(e_2), \ldots, w(e_m))$. The traffic in the network is specified in terms of a $n \times n$ TM between each pair of nodes in the network. The traffic between source node s and destination node d is represented by $\boldsymbol{x}_{s,d}$. In general, there may not be traffic between all source-destination pairs. We use p to denote the number of source-destination pairs between which there is non-zero traffic. In the rest of the paper, instead of viewing the traffic as a matrix, we represent the traffic as a p-vector $\boldsymbol{x}$. For a given set of link weights $\mathbf{w}$, traffic is routed between nodes s and d along the shortest path between s and d. We assume that ties between shortest paths are broken arbitrarily. It is easy to extend this approach to the case where traffic is split between equal cost paths (ECMP). Let $S(e)$ denote the set of source destination pairs that are routed on link e. A source-destination pair $(s, d) \in S(e)$ if link e is on the shortest path from s to d. Let $b(e)$ denote the measured flow on link e. The TM estimation problem is the determination of $\boldsymbol{x}_{s,d}$ given the link load measurements $b(e)$. Note that the traffic flow on link e

$$b(e) = \sum_{(s,d)\in S(e)} x_{s,d}. \tag{1}$$

We create a *routing matrix* A with m rows, one corresponding to each directed link, and p columns, one corresponding to each source-destination pair. For the case of ECMP routing, the elements in the matrix can be either zero of a fraction of the traffic going through the link. In the case of shortest path routing, we set

$$A_{ij} = \begin{cases} 1 & \text{if } M(j) \in S(i) \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

where M is the mapping from row index i to a source-destination pair (s, d). The objective of the TM estimation problem is to determine a *non-negative* solution to the system $\boldsymbol{Ax} = \boldsymbol{b}$ where $\boldsymbol{A}$ is an $m \times p$ routing matrix and $\boldsymbol{b}$ is the link load vector. If there is no additional information, the number of source-destination pairs will be much more than the number of links, then this system has an infinite number of solutions since $m \ll p$. Therefore, we impose additional constraints on $\boldsymbol{x}$ in order to narrow down the solution space.

A. DISTRIBUTION CONSTRAINT

In order to motivate the distribution constraint, we consider the TM estimation problem on a network (NET82) with 82 nodes and 296 directed links. Each TM comprises of $6724 = (82 \times 82)$ potential demands. The NET82 dataset is a real network with available measurements of the real TM. In the TM that was measured, there are 1939 non-zero demands. We show a plot of demand sizes on the left side of Figure 1.

Note that there are a few large demands and several medium to small demands. The right hand side of Figure 1 shows the cumulative distribution function of the normalized demand sizes where the demands are scaled such that the largest demand is one unit. Note that cdf is modeled well using a power law distribution $x^{0.01}$. The same pattern is observed in 4 other demand matrices on the same network. Therefore, when estimating a TM on this network

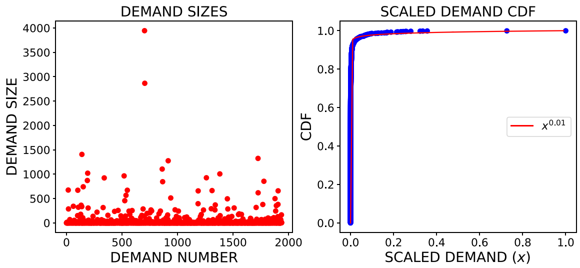


FIGURE 1. Plot of the Demands and the normalized Empirical Distribution Function.

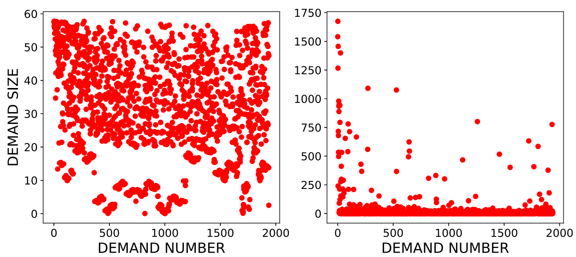


FIGURE 2. Two Different Traffic Matrix Estimates for the Same Link Load Observation.

from link load measurements, we would ideally like this TM to have the same pattern of distribution. Assume that we have observed a link load vector $\boldsymbol{b}$ from an unknown TM and we want to find a solution for the system $\boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \geq 0$. We show two alternative solutions to this system in Figure 2.

In the solution on the left, the TM comprises of uniformly distributed demands. In the solution shown in the right hand side of the Figure 2, the normalized demands follow the power law $x^{0.01}$. It is much more likely, given information about the demand distribution that the actual data looks like the traffic distribution on the right.

We want to caution the reader that even with this additional constraint on the demand size distribution, the TM reconstruction may not be unique. Since the total traffic in the network can change significantly over time, we have to normalize the demands before applying the distribution constraint. In order to formally define the distribution constraint, we first define the the *empirical cumulative distribution function* and the *normalized empirical cumulative distribution function* for a given data set.

Definition 1: Given a set of data points $y_1 \leq y_2 \leq \cdots \leq y_n$ drawn from a distribution, the empirical cumulative distribution function (empirical cdf) of these points is a step function that jumps up by $\frac{1}{n}$ at each of the n data points. Its value at any specified z, is the fraction of observations of the measured variable that are less than or equal to z. The normalized empirical cumulative distribution function (normalized empirical cdf) of these points is a step function that jumps up by $\frac{1}{n}$ at each of the n scaled data points $\frac{y_1}{y_n}, \frac{y_2}{y_n}, \ldots, 1$. Its value at any specified $z \leq 1$, is the fraction of observations of the measured variable that are less than or equal to z.

The domain of the normalized empirical cdf of a set of data points is $[0, 1]$. Assume that the observed TM has a normalized empirical cdf of $G(z)$ for $0 \leq z \leq 1$. As part of the solution procedure, we have to generate random variables having a normalized empirical cdf of $G(z)$. A random variable having cdf $F(z)$ can be generated easily using standard random variable generation procedure. We want to use this process to generated random variables having a normalized empirical cdf of $G(z)$. The following result relates the cdf of a random variable to the normalized cdf of n i.i.d. samples of the random variable.

Theorem 1: Let $X_1, X_2, \ldots, X_n$ represent independent, identically distributed samples from a probability density function $f(x)$ (with the corresponding distribution function $F(x)$). Let

$$Y_j = \frac{X_j}{\max_i X_i} \tag{3}$$

Then, Y_j are distributed with cdf

$$G(y) = n \int_t F(yt)[F(t)]^{n-2} f(t)dt, \quad 0 \leq y \leq 1. \tag{4}$$

Proof: Given $X_1, X_2, \ldots, X_n$ i.i.d. from a distribution function $F(x)$, we let

$$M = \max_{1 \leq i \leq n} X_i. \tag{5}$$

Then

$$Pr[M \leq t] = Pr[X_i \leq t, \quad \forall i] = [F(t)]^n, \tag{6}$$

with the corresponding density function $p_M(t) = n[F(t)]^{n-1} f(t)$. We set

$$Y_j = \frac{X_j}{M}, \quad 1 \leq j \leq n. \tag{7}$$

Then

$$\begin{aligned} Pr\big[Y_j \leq y\big] &= \int_t Pr\big[X_j \leq yt | M = t\big] p_M(t)dt \\ &= \int_t Pr\big[X_j \leq yt | X_j \leq t\big] p_M(t)dt \\ &= \int_t \frac{Pr\big[X_j \leq yt\big]}{Pr\big[X_j \leq t\big]} p_M(t)dt \\ &= n \int_t \frac{F(yt)}{F(t)} [F(t)]^{n-1} f(t)dt \\ &= n \int_t F(yt)[F(t)]^{n-2} f(t)dt, \\ & \quad 0 \leq y \leq 1. \end{aligned} \tag{8}$$

This concludes our proof. ■

We now give an example usage of this theorem that is also very useful in practice to generate samples with the desired normalized empirical cdf. In many examples, the normalized cdf of the demand sizes follows a power law distribution with parameter α. In this case, the $G(x) \sim x^\alpha$ for some specified value of α for $0 \leq x \leq 1$. Note that with a higher value of α, the number of larger demands will be smaller. In the next result, we use Theorem 1 to show that a suitable underlying beta distribution has a normalized power law cdf. The probability density function of a beta distribution is given by

$$f(x) = Cx^{\alpha-1}(1-x)^{\beta-1} \tag{9}$$

where C is a constant to ensure that the total probability is 1. This distribution covers a common case. It is possible to use the result of Theorem 1 to generate any desired normalized empirical cdf.

Normalized Empirical cdf of Beta Distribution: If $X_i \sim B(\alpha, 1)$ for $1 \le i \le n$ denote n i.i.d. samples from a beta distribution with parameters $(\alpha, 1)$ then the distribution and density functions of X_i are

$$F(x) = x^{\alpha}, f(x) = \alpha x^{\alpha-1}, \quad 0 \le x \le 1. \tag{10}$$

Therefore from Theorem 1, the normalized cdf is

$$\begin{aligned} G(y) &= n\int_0^1 F(yt)[F(t)]^{n-2}f(t)dt, \quad 0 \le y \le 1 \\ &= n\int_0^1 (yt)^{\alpha}t^{(n-2)\alpha}\alpha t^{\alpha-1}dt, \quad 0 \le y \le 1 \\ &= n\alpha y^{\alpha}\int_0^1 t^{n\alpha-1}dt, \quad 0 \le y \le 1 \\ &= y^{\alpha}, \quad 0 \le y \le 1 \end{aligned} \tag{11}$$

Note that the normalized empirical cdf is independent of n and is only a function of α. This is not true in general. Therefore, if we need to generate n random variates having a normalized empirical cumulative cdf of x^{α} we do the following:

- Generate $X_1, X_2, \ldots, X_n$ independent random samples from $B(\alpha, 1)$.
- Let $X_{\max} = \max_{1 \le i \le n} X_i$.
- Output

$$\frac{X_1}{X_{\max}}, \frac{X_2}{X_{\max}}, \ldots, \frac{X_n}{X_{\max}}$$

as the set of n samples with normalized empirical cdf x^{α}.

IV. PROJ-D: PROJECTION BASED TRAFFIC MATRIX ESTIMATION METHOD

Kakcmarz method [43] or the Algebraic Reconstruction Technique (ART) is a well known technique for finding a feasible solution to the system $\boldsymbol{Ax} = \boldsymbol{b}$. Assume that there are m rows in the matrix and p columns. Recall that each of the m rows corresponds to a link load measurement and each of the p columns corresponds to a demand. We can represent the set of equations as $\boldsymbol{a_i x} = b_i$ for $i = 1, 2, \ldots, m$, $\boldsymbol{a_i}$ and $\boldsymbol{x}$ are p dimensional vectors. ART is a cyclic projection technique where we start off from an arbitrary initial p-vector $\boldsymbol{x}$. The algorithm then projects this point onto the first constraint $\boldsymbol{a_1 x} = b_1$. Projection just involves finding the closest point to x on the hyperplane $\boldsymbol{a_1 x} = b_1$. This is the new point. This point is then projected onto the second hyperplane and so on until we reach hyperplane m. This point is then projected onto the first hyperplane and this process is repeated in a cyclic manner as shown in the Cyclic Projection Algorithm.

Algorithm 1 Cyclic Projection Method

```
1: Pick an arbitrary p-vector x.
2: for k = 1, 2, ..., K do
3:   for i = 1, 2, ..., m do
4:     x ← x + a_i^T (b_i − a_i x)/(a_i a_i^T)
5:   end for
6: end for
```

Theorem 2: The Cyclic Projection Algorithm shown above converges to a feasible solution to $\boldsymbol{Ax} = \boldsymbol{b}$ after a sufficient number of iterations.

See [43] for a proof of this result. In the description of the cyclic projection algorithm, we refer to one iteration through all m constraints as a *cycle.* This cyclic projection algorithm can be extended directly to the case where we want to find a non-negative feasible solution to the system $\boldsymbol{Ax} = \boldsymbol{b}$ by modifying the projection step by

$$\boldsymbol{x} \leftarrow \max\left\{0, \boldsymbol{x} + \boldsymbol{a_i}^T(b_i - \boldsymbol{a_i x})/\left(\boldsymbol{a_i a_i}^T\right)\right\} \tag{12}$$

where the max operation is a pointwise maximum. In other words, if after computing the projection, some components of $\boldsymbol{x}$ are negative, then we set these components to zero. More recently randomized versions of the cyclic projection method where the next hyperplane to project onto is picked at random has been shown to have linear expected convergence [44]. In order to ensure that the solution satisfies the distribution constraint, we periodically move the current solution to the a compatible point in the distribution. This is done as follows:

- Once every t cycles, we take the current solution $\boldsymbol{x}$ and assume that we renumber the components such that $x_1 \le x_2 \le \cdots \le x_p$.
- We generate p random variates $y_1 \le y_2 \le \cdots \le y_p$ that have the desired normalized empirical distribution. For instance, if we want $\boldsymbol{x}$ to have a power law distribution with power law exponent α, then we generate p i.i.d. samples from a beta distribution $B(\alpha, 1)$.
- We set $x_i = \lambda y_i$ for $1 \le i \le p$ for a suitably chosen scaling parameter λ.

The scaling parameter λ is chosen to minimize the deviation D where D is defined as

$$D = \sum_{j=1}^{m}\left(\lambda \boldsymbol{a_j y} - b_j\right)^2. \tag{13}$$

Note that D is sum of the squared deviation over all the constraints. Using calculus, it is easy to see that the optimal solution is

$$\lambda = \frac{\sum_{j=1}^{m}\left(\boldsymbol{a_j y}\right)b_j}{\sum_{j=1}^{m}\left(\boldsymbol{a_j y}\right)^2} \tag{14}$$

We now scale all the y values by λ and map the y variables to the corresponding x variables, that is, $x_i = \lambda y_i$ for $1 \le i \le p$. This is the new starting point for the next cycle. The algorithm is terminated after K cycles. We can view

Algorithm 2 Proj-D: Projection Based TM Estimation

```
Pick an arbitrary p-vector x.
for k = 1, 2, ..., K do
    for j = 1, 2, ..., t do
        for i = 1, 2, ..., m do
            x ← max{0, x + a_i^T (b_i − a_i x)/(a_i a_i^T)}
        end for
        Reorder x such that x_1 ≤ x_2 ≤ ··· ≤ x_m
        Generate y_1 ≤ y_2 ≤ ··· ≤ y_p with the desired
        normalized empirical distribution
        Compute λ = (Σ_{j=1}^{m} (a_j y) b_j) / (Σ_{j=1}^{m} (a_j y)^2)
        Set x_i = λ y_i for 1 ≤ i ≤ p
    end for
end for
```

this process as running the cyclic projection method with K starting solutions having the desired normalized empirical cdf. The overall description of the algorithm is shown below.

Proj-D assumes that the there is only enough data or operator experience to specify the (normalized) distribution of the demands. The projection based approach is tailor made for the TM estimation problem with a distribution constraint, and it outperforms the GAN based approach if we only have distribution information in our experiment.

V. GENERATIVE ADVERSARIAL NETWORKS

The idea of using a GAN based approach to capture spatial correlation in the TM was motivated by the impressive capabilities demonstrated by GANs for generating samples that resemble real world images [32]–[34]. In addition, GAN based methods can also be used for data sharing with privacy protection [36], [37]. The training of a GAN involves a game between the generator network and discriminator network. The generator and discriminator are both neural networks. The generator learns a mapping from random noise to the space of the given signal. The discriminator tries to distinguish between the real signal and the generated signal. During the game of GAN training, the discriminator is updated by learning from the real and generated images. The generator is updated by the gradient provided by the discriminator so that the generator learns to generate samples that resemble the real images.

The game between the generator T and discriminator D can be written as the objective:

$$\min_T \max_D \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r}\left[log(D(\boldsymbol{x}))\right] + \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim\mathbb{P}_t}\left[1 - log(D(\widetilde{\boldsymbol{x}}))\right] \quad (15)$$

where $\mathbb{P}_r$ is the distribution of real data and $\mathbb{P}_t$ is the distribution of the data generated from the generator network T.

The game involved in the training process of a GAN requires that there exists some kind of balance between the generator and discriminator. If the discriminator is too strong then it fails to provide useful gradient for the training of generator. Various kinds of methods have been proposed to stabilize the training process of GANs [33], [34]. In [34], the method WGAN was proposed, where the Wasserstein-1 distance was used for the training of GANs. The WGAN value function is constructed using the Kantorovich-Rubinstein duality [45] to achieve

$$\min_T \max_D \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})] - \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim\mathbb{P}_t}[D(\widetilde{\boldsymbol{x}})] \quad (16)$$

The value function in WGAN makes the gradient of the critic function with respect to its input behave better, thus making optimization of the generator easier. However, according to [34], to enforce the Lipschitz constraint on the critic, the weights of the critic has to be clipped within a compact space, which could lead to capacity under use and exploding or vanishing gradients, which makes the training process unstable [33]. The authors in [33] proposed a gradient penalty approach for the training of GANs called WGAN-GP, which shows even better performance for the task of image generation. In this case the new objective is

$$\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})] - \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim\mathbb{P}_t}[D(\widetilde{\boldsymbol{x}})] + \lambda\mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{x}}}\left[\left(\left\|\nabla_{\hat{\boldsymbol{x}}}D(\hat{\boldsymbol{x}})\right\|_2 - 1\right)^2\right] \quad (17)$$

where $\mathbb{P}_{\hat{\boldsymbol{x}}}$ is the distribution of uniform samples along the straight lines between pairs of points sampled from distribution of data $\mathbb{P}_r$ and distribution of the generator $\mathbb{P}_t$. According to their experiment results, the added gradient penalty term helps further improve the performance of GANs. In this paper we adopt the method of WGAN-GP as the training process of the GAN.

VI. TRAFFIC MATRIX ESTIMATION USING GENERATIVE ADVERSARIAL NETWORKS

Since GANs can capture the characteristics of given data, the authors in [31] proposed to use a GAN as a mapping from latent space to signal space for the application of compressive sensing.

As shown in [31], if we assume the vector we want to recover is k-sparse in some basis, with d-layer neural networks as GANs, $O(kd \log n)$ Gaussian measurements can guarantee good reconstruction with high probability. Their results show that the GAN based compressive sensing method achieves better performance when the sampling rate of the signal is low. And the performance is also robust with non-Gaussian measurements, such as in the case of image inpainting.

The problem of TM estimation given link measurement has the same format as the problem of image compressive sensing. Since the link measurements of a TM are also relatively low, we propose to solve the TM estimation problem with a GAN as the generator for the TM. Suppose the latent variable of the GAN is ℓ, the generator T generates the estimated TM $T(\ell)$, then the problem of TM estimation can be written as:

$$\min_{\boldsymbol{l}} \|\boldsymbol{y} - \boldsymbol{A}T(\boldsymbol{\ell})\|_2^2. \quad (18)$$

A properly trained generator T provides a mapping from the lower dimensional latent space to the space of possible TMs.

Algorithm 3 GAN Based TM Estimation Method

Generate random Gaussian noise $\boldsymbol{n_0}$.
$\hat{\boldsymbol{n}} = \hat{\boldsymbol{n}}_0$
for $i = 1;\ i < N_i;\ i++$ **do**
  Generate random Gaussian noise $\boldsymbol{n_i}$
  if $\|\boldsymbol{y} - \boldsymbol{A}T(n_i)\|_2^2 < \|\boldsymbol{y} - \boldsymbol{A}T(\hat{n})\|_2^2$ **then**
    $\hat{\boldsymbol{n}} = \boldsymbol{n_i}$
  end if
end for
for $j = 0;\ j < N_2;\ j++$ **do**
  $\hat{\boldsymbol{n}} = \hat{\boldsymbol{n}} + \nabla_{\boldsymbol{n}} L$
end for

Since the function T is differentiable, the objective function can be optimized by simple gradient descent.

Compared with the projection approach, the estimation method using a GAN can be applied for more general cases. If we only have knowledge of the normalized empirical distribution then the GAN can be trained with data generated from the given normalized empirical distribution. If measurements from the past are available, the GAN can also be trained with the data from the past and shared for estimation purposes.

A. TRAFFIC MATRIX ESTIMATION UNDER DISTRIBUTION CONSTRAINT

We first consider the problem of TM estimation under a distribution constraint. Unlike the constraint of signal sparsity from previous compressive sensing methods, which can be enforced by adding sparsity regularization terms to the objective function, it is unclear how a distribution constraint can be incorporated into the objective function. However, since a GAN is able to capture the characteristics of given data and generate samples with similar features, the distribution constraint can be included in the objective function by training a GAN that generates samples following a similar distribution. Then the optimization can be conducted in the latent space. Given the cost function

$$L = \|\boldsymbol{y} - \boldsymbol{A}T(\boldsymbol{\ell})\|_2^2, \tag{19}$$

the gradient of L can be easily computed by the chain rule. Therefore L can be updated step by step by using simple stochastic gradient descent or any other optimizer such as the adaptive moment estimation (Adam) optimizer [46]. For the experiments in this paper we use the Adam optimizer as the optimizer over the latent space. In the experiments, we find that choosing a better initial point in the latent space can help reduce the optimization steps and provide better estimation results. So we generate N_i random vectors $\boldsymbol{n_i}$ in the latent space and select the one that provides link measurements that is closest to the given link measurements. Then we run the optimization for N_2 steps. We show the details of this method in Algorithm 3. This GAN based estimation method under a distribution constraint is denoted as GAN-D.

B. TRAFFIC MATRIX ESTIMATION WITH TRAINING DATA

In some cases, in addition to link measurements, some TMs from the past may be also available. In this case the GAN can be directly trained with the available data. In addition to the distribution of demands, the TM data may also contain spatial information that can be learned by the GAN. With the trained generator, the optimization steps will be the same as those with a distribution constraint.

VII. EXPERIMENT SETUP

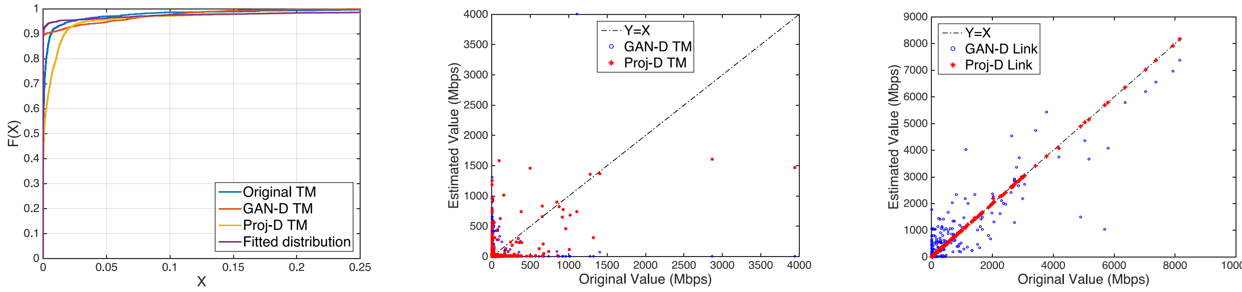
We evaluate the performance of our methods with three datasets. The first dataset is the NET82 dataset which contains one TM measured on a real topology with 82 nodes. The second dataset is the Abilene dataset [10] which contains TMs with 12 nodes and 54 links. The third dataset is the GÉANT dataset [11], which has 23 nodes and 38 links. Note that when $\beta = 1$ the Beta distribution becomes a power law distribution. In our experiments we found that the power law distribution is sufficient for fitting the distribution of the TMs. And the α values are the maximum likelihood estimates from the measured TMs [29].

Firstly we test the performance of our method assuming only the distribution of the demands is known. For the first dataset a Beta distribution with $\alpha = 0.01154$, $\beta = 1.0$ is used for the projection based method (Proj-D) and the GAN based method (GAN-D). The parameters are directly used for Proj-D. For GAN-D, we first train the GAN with random matrices generated from the fitted distribution, then we use the GAN for TM estimation.

For the Abilene dataset we use the TMs collected from March to June for distribution fitting. We use 1000 of the TMs collected in July for testing. We fit a Beta distribution with $\alpha = 0.0107$, $\beta = 1.0$ according to all the demands collected from March to June. Similar to the case of the first dataset, for Proj-D we use the Beta distribution directly. For GAN-D, the TMs from March to June are available and the TM estimation is conducted for the data in July. So the TMs from March to June can be used for the training of the GAN. The GAN is trained for 300 epochs, with 27360 TMs collected from March to June.

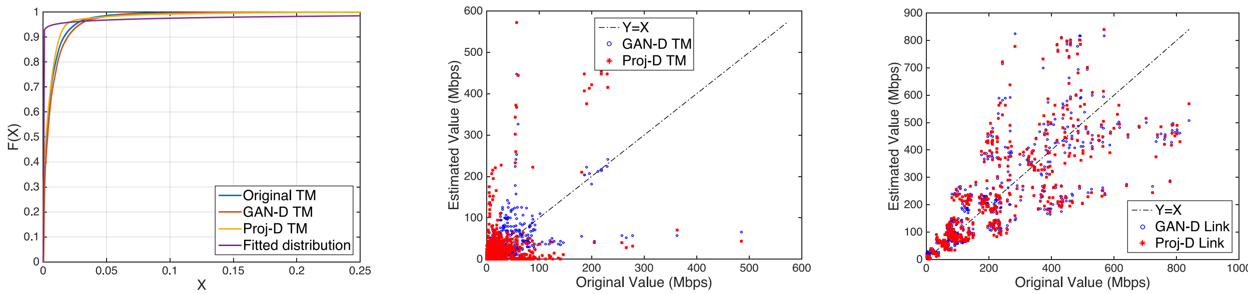
For the GÉANT dataset, a Beta distribution with $\alpha = 0.01411$ and $\beta = 1.0$ is used for Proj-D. For GAN-D, 8016 TMs collected from January to March are used for the training of GAN. Network parameters for the GAN are the same as those for the Abilene dataset. Both methods are tested on 1000 TMs collected in April.

In our experiment, all the data from the training set is used to fit the distribution. However, we found that the distribution can be fitted with much less data. To simulate a practical scenario, we randomly selected 35000 demands from the training set of the Abilene topology. The sampled data is used for distribution fitting. This experiment is repeated 100 times, among them the maximum fitted value of α is 0.0136 and the minimum value is 0.0107. Similarly, for the GÉANT dataset, 60000 demands



(a) Empirical CDF (b) Demands (c) Link measurements

FIGURE 3. Performance evaluation on the NET82 dataset (shortest path).



(a) Empirical CDF (b) Demands (c) Link measurements

FIGURE 4. Performance evaluation on the Abilene dataset (shortest path).

TABLE 1. Performance comparison.

Shortest Path			
Method	NET82	Abilene	GÉANT
Proj-D (RMSE/Mbps)	125.94	40.47	87.96
GAN-D (RMSE/Mbps)	194.81	25.74	65.69
Proj-D (NMAE)	1.20	0.94	1.51
GAN-D (NMAE)	1.93	0.66	1.18
ECMP			
Method	NET82	Abilene	GÉANT
Proj-D (RMSE/Mbps)	153.35	42.05	87.12
GAN-D (RMSE/Mbps)	191.34	25.74	65.81
Proj-D (NMAE)	1.33	0.97	1.50
GAN-D (NMAE)	1.95	0.66	1.18

were randomly sampled from the training set. The maximum value of α is 0.0146 and the minimum value is 0.0137.

Network Parameters: We use the same structure for the GAN for all the datasets. The generator of the GAN is a fully connected neural network with hidden layers of size 32, 64 and 128. The discriminator is also a fully connected neural network with hidden layers of size 512, 256, 256 and 256. We do not focus on finding the best parameters of the GAN in this paper. However we found it beneficial to use a larger neural network for the discriminator, so that the discriminator can more efficiently capture the difference between TMs and random matrices. ReLU is used as the activation function for the neural networks. To keep the balance between the capability of the discriminator and the generator, we update the discriminator 64 times after each training step of the generator.

VIII. PERFORMANCE EVALUATION

In this paper, we focus on the practical scenario where the past TMs are not available for estimation purpose. The existing methods for TM estimation either require access to past TMs, or can not exploit the distribution constraint, therefore it is unsuitable to compare our methods with the previous spatial or temporal correlation based methods. Performance of the methods are evaluated with two different metrics: the root mean square error (RMSE) and the normalized mean absolute error (NMAE) of the estimation results. The RMSE can be written as:

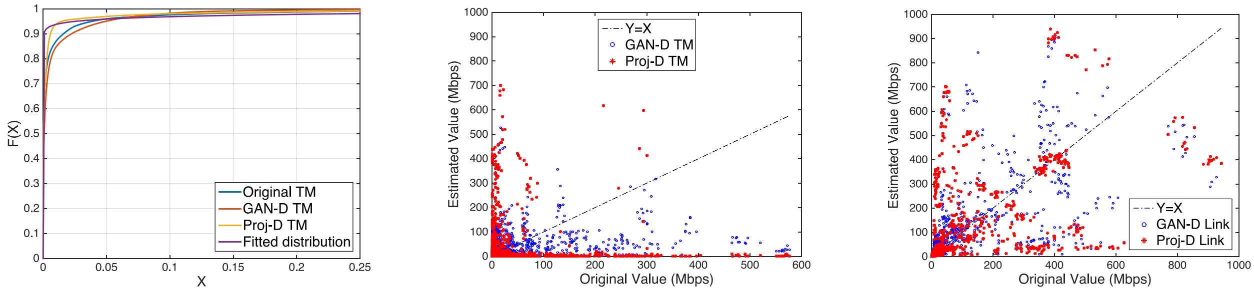
$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x-\hat{x})^2}{n}}. \quad (20)$$

The NMAE can be written as:

$$NMAE = \frac{\|x-\hat{x}\|_1}{\|x\|_1}. \quad (21)$$

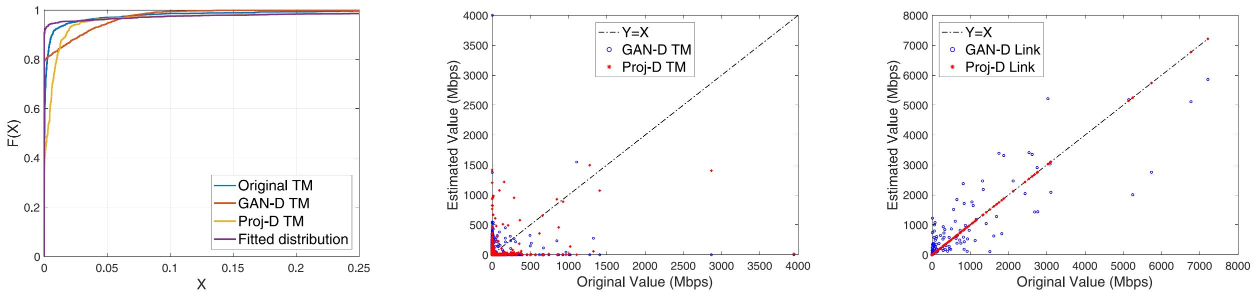
The results are shown in Table 1. Errors are calculated for the non-zero demands.

For the NET82 dataset, Proj-D achieves RMSE of 125.94 Mbps and NMAE of 1.20. The RMSE of the results from GAN based method is 194.81 Mbps and the NMAE is 1.93. To evaluate the method's ability to meet the distribution constraint, we also compare the empirical cumulative distribution function (CDF) of the solutions. Figure 3 evaluates the performance of the methods on NET82. Figure 3(a) shows the CDF of the solutions, the fitted distribution and original data. Figure 3(b) shows the recovered demands versus the original demands. Figure 3(c) shows the fitted link



(a) Empirical CDF (b) Demands (c) Link measurements

FIGURE 5. **Performance evaluation on the GÉANT dataset (shortest path).**



(a) Empirical CDF (b) Demands (c) Link measurements

FIGURE 6. **Performance evaluation on the NET82 dataset (ECMP).**

measurements versus the given link measurements. For the CDF plot, the TMs are normalized by the maximum value of all the TMs. Since very few of the normalized values are greater than 0.25, we show the CDF plot from 0 to 0.25 to better evaluate how well the estimated TMs fit the original distribution.

Figure 4 shows the CDF, demands and link measurement of recovery results of the two methods on the Abilene dataset. The CDF plot is generated in the same way as in Figure 4. The demand plot and link measurement plot are generated from the first ten TMs.Proj-D achieves RMSE of 40.47 Mbps and NMAE of 0.94, while GAN-D achieves RMSE of 25.74 Mbps and NMAE of 0.66. Figure 5 shows the CDF, demands and link loads of recovery results of the two methods on the GÉANT dataset. The demand plot and link measurement plot are generated from the first ten TMs. Proj-D achieves NMAE of 1.51, GAN-D achieves NMAE of 1.18. In terms of RMSE, Proj-D has RMSE of 87.96 Mbps, GAN-D has RMSE of 65.69 Mbps.

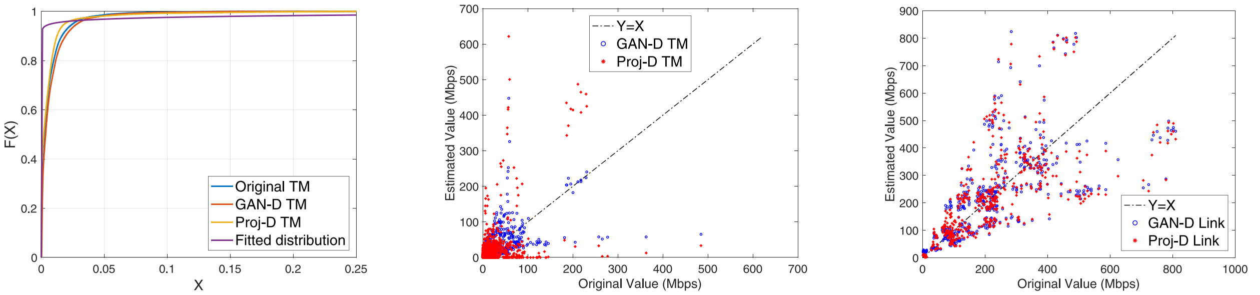
Comparing results shown in Figure 3, both methods are able to generate results that satisfy the distribution constraint. GAN-D is able to generate results that are closer to the original distribution. However Proj-D is able to generate data that fit better to the link measurement constraint, with the cost of diverting a bit from the given distribution constraint. Though the GAN learns to generate samples according to the given distribution, it is not able to cover all possible space of the distribution, therefore GAN-D performs worse than Proj-D in terms of RMSE and NMAE.

For the Abilene dataset, both methods are able to provide estimation results that closely meet the distribution constraint and link measurement constraints. Since the GAN is trained with real TMs measured from the past, it is able to learn the spatial correlations and other structural information of the TMs from the training data. For GAN-D the number of optimization steps N_2 also determines how well the results meet the link measurement constraints; with more optimization steps the results will fit the link measurement constraints better, but the elements of the estimated TMs will start to divert from the real value after certain number of steps. We use $N_2 = 10000$ in our experiment. For Proj-D the results can better meet the link measurement constraints, at the cost of diverting a bit from the distribution constraint.

For the GÉANT dataset, both Proj-D and GAN-D are able to generate results that fit the distribution constraint. This may be because that there are fewer links in this dataset so both methods can meet the distribution constraint without over-fitting. However, in terms of NMAE and RMSE the GAN based method still performs better than Proj-D. So the GAN is still able to learn spatial and structural information from the TMs used for training.

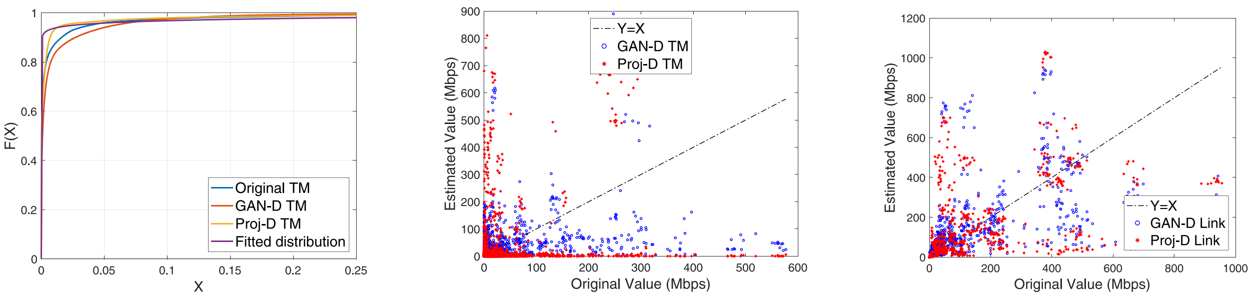
In addition to shortest path routing, Figures 6, 7 and 8 show the results with ECMP routing. Since the methods do not depend on any specific routing mechanism, they achieve similar performance with ECMP routing.

In general, there exists the choice between meeting the distribution or similarity constraint better or meeting the link measurement constraint better. The GAN based method is



(a) Empirical CDF (b) Demands (c) Link measurements

FIGURE 7. Performance evaluation on the Abilene dataset (ECMP).



(a) Empirical CDF (b) Demands (c) Link measurements

FIGURE 8. Performance evaluation on the GÉANT dataset (ECMP).

able to provide estimation results that meets the distribution constraint better. The Projection based method is able to generate results that have better fit of the link measurement constraints. The users can choose either one of the methods based on their requirements in the specific use cases.

Due to the practical problem settings in this paper, the other spatial-temporal based method or learning based methods will not work in this setting due to their requirement of direct access to TM measurements. We compare our method with the simple gravity model in [10], which requires the least amount of additional information in our problem setting. The simple gravity model achieves a RMSE of 18.4 Mbps and NMAE of 0.51 on the Abilene dataset, performing slightly better than GAN-D. This suggests that the GAN based method is able to learn from the training set and exploit the spatial correlation, similar to hand-crafted models. Since Proj-D only considers a distribution constraint and does not incorporate network domain knowledge, it performs worse than the gravity model. However, the gravity model requires measurement of total inbound and outbound traffic at edge links. Sharing of such kind of data can violate regulations [4], [12], [14]. GAN-D and Proj-D can work without edge link measurements and better protect users' privacy.

IX. CONCLUSION AND FUTURE WORK

In this paper we focus on the practical scenario of TM estimation where past TMs are not directly accessible for estimation purposes, due to measurement complexity or privacy protection regulations. We proposed two methods for the problem of TM estimation given link measurements under a constraint of the distribution of demands. Experiment results show that both the method Proj-D and GAN-D are able to generate estimation results that fit the link measurements and the distribution constraint. The Projection based method is able to provide estimation results that fits the link constraints better, while the GAN based method generates TMs that better fit the given distribution. In addition, if TMs measured in the past are available, the GAN based method is able to learn the spatial and structural correlations of the TM data and provide better estimation results. Future work includes extending these methods to other similar problems and finding the suitable kind of GAN for the GAN based method.

REFERENCES

[1] E. Lawrence, G. Michailidis, V. N. Nair, and B. Xi, "Network tomography: A review and recent developments," in *Frontiers in Statistics*. Singapore: World Sci., 2006, pp. 345–366.

[2] M. Mardani and G. B. Giannakis, "Estimating traffic and anomaly maps via network tomography," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1533–1547, Jun. 2016.

[3] S. Xu, M. Kodialam, T. V. Lakshman, and S. S. Panwar, "Tomography based learning for load distribution through opaque networks," 2020. [Online]. Available: arXiv:2007.09521.

[4] *Privacy and Data Protection by Country*, Forrester Res., Cambridge, MA, USA, 2020. [Online]. Available: https://goo.gl/BVL8ww

[5] F. Xiao, L. Chen, H. Zhu, R. Hong, and R. Wang, "Anomaly-tolerant network traffic estimation via noise-immune temporal matrix completion model," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1192–1204, Jun. 2019.

[6] L. Nie, H. Wang, X. Jiang, Y. Guo, and S. Li, "Traffic measurement optimization based on reinforcement learning in large-scale its-oriented backbone networks," *IEEE Access*, vol. 8, pp. 36988–36996, 2020.
[7] V. An Le, P. Le Nguyen, and Y. Ji, "Deep convolutional LSTM network-based traffic matrix prediction with partial information," in *Proc. IFIP/IEEE Symp. Integr. Netw. Serv. Manag. (IM)*, Arlington, VA, USA, 2019, pp. 261–269.
[8] A. Kumar, S. Vidyapu, V. V. Saradhi, and V. Tamarapalli, "A multi-view subspace learning approach to Internet traffic matrix estimation," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 1282–1293, Jun. 2020.
[9] G. Liu, S. Guo, B. Xiao, and Y. Yang, "SDN-based traffic matrix estimation in data center networks through large size flow identification," *IEEE Trans. Cloud Comput.*, early access, Oct. 1, 2019, doi: 10.1109/TCC.2019.2944823.
[10] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 206–217, 2003.
[11] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 83–86, 2006.
[12] D. C. Sicker, P. Ohm, and D. Grunwald, "Legal issues surrounding monitoring during network research," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, 2007, pp. 141–148.
[13] C. Partridge and M. Allman, "Ethical considerations in network measurement papers," *Commun. ACM*, vol. 59, no. 10, pp. 58–64, 2016.
[14] A. C. Zhou, Y. Xiao, Y. Gong, B. He, J. Zhai, and R. Mao, "Privacy regulation aware process mapping in geo-distributed cloud data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1872–1888, Aug. 2019.
[15] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *J. Amer. Stat. Assoc.*, vol. 91, no. 433, pp. 365–377, 1996.
[16] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas.*, 2002, pp. 71–82.
[17] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 61–72, 2004.
[18] 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 Trans. Netw.*, vol. 13, no. 5, pp. 947–960, Oct. 2005.
[19] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 267–278, 2009.
[20] G. Gürsun and M. Crovella, "On traffic matrix completion in the Internet," in *Proc. Internet Meas. Conf.*, 2012, pp. 399–412.
[21] Y.-C. Chen, L. Qiu, Y. Zhang, G. Xue, and Z. Hu, "Robust network compressive sensing," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 545–556.
[22] K. Xie *et al.*, "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Hong Kong, 2015, pp. 2443–2451.
[23] K. Xie, L. Wang, X. Wang, G. Xie, J. Wen, and G. Zhang, "Accurate recovery of Internet traffic data: A tensor completion approach," in *Proc. IEEE INFOCOM 35th Annu. Int. Conf. Comput. Commun.*, San Francisco, CA, USA, 2016, pp. 1–9.
[24] K. Xie, C. Peng, X. Wang, G. Xie, and J. Wen, "Accurate recovery of Internet traffic data under dynamic measurements," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Atlanta, GA, USA, 2017, pp. 1–9.
[25] K. Fukuda, "Towards modeling of traffic demand of node in large scale network," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, 2008, pp. 214–218.
[26] A. B. Downey, "Evidence for long-tailed distributions in the Internet," in *Proc. 1st ACM SIGCOMM Workshop Internet Meas.*, 2001, pp. 229–241.
[27] B. Chandrasekaran, *Survey of Network Traffic Models*, vol. 567, Waschington Univ. St. Louis CSE, St. Louis, MO, USA, 2009.
[28] A. Nucci, A. Sridharan, and N. Taft, "The problem of synthetically generating IP traffic matrices: Initial recommendations," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 19–32, 2005.
[29] G. Hahn and S. Shapiro, "Statistical models in engineering," in *Statistical Models in Engineering*, G. J. Hahn and S. S. Shapiro, Eds. New York, NY, USA: Willey, 1967.
[30] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[31] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 537–546. [Online]. Available: JMLR.org
[32] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2014, pp. 2672–2680.
[33] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 5767–5777.
[34] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017. [Online]. Available: arXiv:1701.07875.
[35] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013. [Online]. Available: arXiv:1312.6114.
[36] Z. Xiong, W. Li, Q. Han, and Z. Cai, "Privacy-preserving auto-driving: A GAN-based approach to protect vehicular camera data," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, Beijing, China, 2019, pp. 668–677.
[37] Y. Qu, S. Yu, W. Zhou, and Y. Tian, "GAN-driven personalized spatial-temporal private data sharing in cyber-physical social systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2576–2586, Oct./Dec. 2020.
[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[39] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[40] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2012, pp. 341–349.
[41] L. Nie, D. Jiang, L. Guo, S. Yu, and H. Song, "Traffic matrix prediction and estimation based on deep learning for data center networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, USA, 2016, pp. 1–6.
[42] J. Zhao, H. Qu, J. Zhao, and D. Jiang, "Spatiotemporal traffic matrix prediction: A deep learning approach with wavelet multiscale analysis," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 12, p. e3640, 2019.
[43] S. Kaczmarz, "Angenäherte auflösung von systemen linearer gleichungen," *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques*. Cracovie, Poland: Impr. de l'Université, 1937, pp. 355–357.
[44] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *J. Fourier Anal. Appl.*, vol. 15, no. 2, p. 262, 2009.
[45] C. Villani, *Optimal Transport: Old and New*. Berlin, Germany: Springer, 2008, vol. 338.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980.

SHENGHE XU received the B.Sc. degree in information engineering from Xi'an Jiaotong University in 2015, and the Ph.D. degree in electrical engineering from the New York University Tandon School of Engineering in 2020. He worked as a summer research intern with Cisco Systems, San Jose, CA, USA, in 2017. In 2019, he worked as a summer research intern with Nokia Bell Labs, Holmdel, NJ, USA. His research interests include wireless communications, computer networks, and machine learning.

MURALI KODIALAM (Member, IEEE) received the Ph.D. degree from the Massachusetts Institute of Technology. In 1992, he joined Bell Laboratories, where he is currently a Bell Labs Fellow. He has authored over 100 refereed conference and journal publications, and holds over 40 patents. His general research interests are in resource allocation in communication networks. He has worked on IP routing, switch scheduling and data structures and algorithms for fast packet processing in wired and wireless networks. He was a co-recipient of the 2008 IEEE Leonard Abraham Prize, the 2011 IEEE Bennett Prize, and the IEEE INFOCOM 2014 best paper awards. He served as an Associate Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING.

T. V. LAKSHMAN (Fellow, IEEE) received the master's degree in physics from the Indian Institute of Science, Bengaluru, India, in 1984, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1986. He is currently the Head of the Networks Research Group, Nokia Bell Laboratories. His research contributions span a spectrum of networking topics, including switch architectures, network design, TCP performance, traffic management, and software-defined networking. He was a recipient of several IEEE and ACM awards, including the IEEE Leonard Abraham Prize, the IEEE Communication Society William R. Bennett Prize, the IEEE INFOCOM Achievement Award, the IEEE Fred W. Ellersick Prize Paper Award, and the ACM SIGMETRICS Best Paper Award. He also received the 2010 Thomas Alva Edison Patent Award from the R&D Council of New Jersey. He has been an Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING and the IEEE TRANSACTIONS ON MOBILE COMPUTING. He is a Fellow of Bell Laboratories and ACM.

SHIVENDRA S. PANWAR (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 1986. He is a Professor with the Electrical and Computer Engineering Department, NYU Tandon School of Engineering. He is the Director of the New York State Center for Advanced Technology in Telecommunications (CATT), the Faculty Director and a Co-Founder of the New York City Media Lab, and a Member of NYU Wireless. His research interests include the performance analysis and design of networks. His current research focused on cross-layer research issues in wireless networks, and multimedia transport over networks. He has coauthored a textbook: *TCP/IP Essentials: A Lab based Approach* (Cambridge University Press). He was a winner of the IEEE Communication Society's Leonard Abraham Prize for 2004, the ICC Best Paper Award in 2016, and the Sony Research Award. He was also co-awarded the Best Paper in 2011 Multimedia Communications Award. He has served as the Secretary of the Technical Affairs Council of the IEEE Communications Society.