On the Use of Random Neural Networks for Traffic Matrix Estimation in Large-Scale IP Networks

Pedro Casas[*] and Sandrine Vaton[*]
[*]Télécom Bretagne, Brest, France - Email: firstname.lastname@telecom-bretagne.eu

***Abstract*—Despite a large body of literature and methods devoted to the Traffic Matrix (TM) estimation problem, the inference of traffic flows volume from aggregated data still represents a major issue for network operators. Directly and frequently measuring a complete TM in a large-scale network is costly and difficult to perform due to routers limited capacities. In this paper we introduce and evaluate a new method to estimate a TM from easily available link load measurements. The method uses a novel statistical learning technique to unveil the relation between links traffic volume and origin-destination flows volume. By training a system based on Random Neural Networks, we provide a fast and accurate TM estimation tool that attains proper results without assuming any traffic model or particular behavior. Using real data from an operational backbone network, we compare this new method to the most well known and accepted TM estimation techniques, including in the evaluation some more accurate and up-to-date methods developed in recent works. Results show that current TM estimation techniques can still be improved.**

***Index Terms*—Traffic Matrix Estimation, Statistical Learning, Random Neural Networks.**

I. Introduction

Knowing and understanding the traffic that flows through a large-scale network represents a key issue in the design and engineering of new network services and architectures. A network-wide view of the traffic in a large-scale IP network is typically described by a Traffic Matrix (TM). The TM represents the volume of traffic transmitted between every pair of nodes in a network, also referred to as the origin-destination (OD) traffic flows. The measurement of the TM is a subject of continuous debate between researchers, network operators and technology vendors. Some of them claim that the evolution of new access technologies and the development of optical access networks is such, that the overheads incurred in the direct measurement of the TM will become too costly and prohibitive in the future. On the other hand, the progress in monitoring and measurement technology of the past years makes some others believe that the challenge of directly measuring the TM can be solved by improving equipment measurement capabilities. Whatever the result of this struggle between traffic increase and progress in measurement capabilities, network analysis requires efficient TM estimation methods that use both aggregated data and direct measurements to improve results.

Let us formally introduce the TM estimation problem. Throughout the paper, the vector $X_t = [x_t(1), .., x_t(m)]^T$ represents the value of the TM at time t, where $x_t(k)$ stands for the traffic volume of each OD flow $k = 1..m$ at time t. Similarly, the vector $Y_t = [y_t(1), .., y_t(l)]^T$ represents the value of the links aggregated traffic volume, where $y_t(i)$ represents the total traffic volume in link $i = 1..l$ at time t. This aggregated data is available through the standard and well-known SNMP protocol, so it will be usually referred to as the SNMP measurements. In practice, time t is a discrete variable and the SNMP measurements consist of the cumulative number of bytes that enter or exit an IP network interface between two consecutive times of measurement. We additionally introduce the link capacities vector $C = [c_1, c_2, .., c_l]^T$, where c_i stands for the capacity of link i. The OD flows traffic and the SNMP measurements are related through the routing matrix R, a $l \times m$ matrix in which element R_{ij} is equal to 1 if OD flow j traverses link i, and 0 otherwise:

$$Y_t = R\, X_t \tag{1}$$

The computation of X_t from Y_t represents a poorly-posed problem, as the number of unknown OD flows is much larger than the number of links, $m >> l$. This simple relation is the basis of the celebrated TM Estimation (TME) problem, introduced in 1996 by Vardi [1].

II. State of the Art

The first approach to tackle the TME problem was to search for direct solutions, introducing additional information to create additional constraints. This was achieved by Poisson and Gaussian TM modeling assumptions in [1], [2], deriving higher order statistics of OD flows traffic as the additional constraints. In [3], authors showed that the basic assumptions underlying these statistical models were not always justified, and that these methods performed badly when the underlying assumptions were violated.

Additional spatial information about the TM was included into the problem, taking into account the network topology and the routing process. This encouraged the application of gravity models to the TME issue [4]. Authors in [5] made a breakthrough in the TME problem with their Tomo-Gravity approach, combining network tomography methods [1] with gravity models to highly improve accuracy.

A further step was achieved by considering the strong diurnal patterns found in the TM. Authors in [6] proposed a pure data-driven approach to analyze OD flows, using a Principal Component Analysis (PCA) method to capture spatial correlations in the TM. A dynamic model was adopted in [7] to capture the temporal correlation of the TM, using a

Kalman Filter for recursive estimation. These methods assume that the TM can be directly measured during limited periods of time for calibration purposes. Although accurate enough for many management tasks, results presented in [7]–[9] showed that they can be highly unstable if the underlying models are not periodically re-calibrated.

New methods have emerged in the last few years. In [10] we have presented a novel TME method based on spatial parsimonious modeling and Maximum Likelihood estimation techniques. This method presents quite accurate results and optimality properties on the estimate, but estimation errors are similar to those obtained by the Tomo-Gravity approach. The last method that we highlight was recently introduced in [11], where a three-layers feed-forward Artificial Neural Network (ANN) model is used to learn the relation between links traffic and OD flows traffic. This method is appealing, but presents a major conception drawback: statistical learning with ANNs provides results which are very sensitive to the particular definition of the neural network topology [13], [14], turning current implementation of the method highly unstable and difficult to calibrate.

Contributions of the Paper

In this paper we develop a new method to estimate the TM from easily available links traffic measurements. Our method draws on the main ideas of [11], but improves the technique by using a new kind of neural network, introduced in recent years by E. Gelenbe [16]: the Random Neural Network (RNN). As we will see and as it has been shown in many previous applications [12], RNNs are a very powerful tool to capture the intrinsic model behind the data. Using a three-layers neural network topology as the one considered in [11], we show that ANNs performance is highly dependent on the number of neurons in the topology, while results obtained with RNNs are remarkably stable.

Using the real topology and real TMs from an Internet2 backbone network, we compare the performance of this new method against two classical and very well known TME methods, namely the Gravity and the Tomo-Gravity methods. We also include three modern algorithms in the comparative evaluation, developed in [6], [10]. Our analysis shows that the RNN-based method is more accurate than current methodologies for large-scale TME, but that the learning technique is highly dependent on the particular characteristics of the routing matrix.

The remainder of this paper is organized as follows. In section III we present the new TME technique, based on Random Neural Networks. In this section we also evidence the stability problems of ANNs for TME w.r.t. RNNs. Section IV presents the evaluation of the RNN-based method in different large-scale networks, comparing its performance against the one attained by the rest of the algorithms in an operational Internet2 backbone network. Finally, section V concludes this work.

III. The RNN for TM Estimation

The RNN-TME method introduced in this work has its origins in the statistical learning field. The method uses multiple Random Neural Networks to reconstruct OD flows volume X_t from SNMP measurements Y_t. From (1), we know that traffic volume at link i and time t is a linear combination of OD flows volume at time t, $y_t(i) = R_i\, X_t$, where R_i is the i-th row of R. The main idea of our method is to find a certain non-linear transfer-block $f_k(\cdot) : \mathbb{R}^{n_k} \to \mathbb{R}$ for each OD flow k, such that:

$$x_t(k) = f_k\left(Y_t(\boldsymbol{\delta}_k)\right) = f_k\left(y_t(\delta_k^1), y_t(\delta_k^2), .., y_t(\delta_k^{n_k})\right) \quad (2)$$

The vector $Y_t(\boldsymbol{\delta}_k)$ contains the traffic volume of the n_k links which are crossed by OD flow k, where vector $\boldsymbol{\delta}_k = \left(\delta_k^1, \delta_k^2, .., \delta_k^{n_k}\right)$ has the indexes of the n_k elements in the k-th column of R that are different from zero. The non-linear transfer-block $f_k(\cdot)$ "extracts" the volume of OD flow k from the trace that this flow leaves in the n_k links. We use the term transfer-block instead of function because $f_k(\cdot)$ can not be formally defined as it. It is easy to see that, in theory, the same values of links volume can result from different combinations of different OD flows traffic. However and as we will see in the results, this does not happen in practice. Computing $f_k(\cdot)$ can be simply thought as computing a pseudo-inverse matrix from routing matrix R, for a particular element of the TM. Indeed, we will show in the evaluations that the structure of $f_k(\cdot)$ is strongly related to the characteristics of R. The idea of the method is then to learn m individual transfer-blocks $f_k(\cdot)$ from measurements, one for each OD flow of the TM, using a single RNN model to build each block.

The RNN model can be described as a merge between the classical ANN model and queuing networks. RNNs are, as ANNs, composed of a set of interconnected neurons. Each neuron exchanges impulse signals with other neurons and with the environment, and has a potential associated with it, which is an integer random variable. The potential of neuron i at time t is denoted by $q_t(i)$. If the potential of neuron i is strictly positive, the neuron is *excited*; in this state, it randomly sends signals according to a Poisson process with rate r_i. In this model, neurons exchange *positive* and *negative* signals. The probability that a signal sent by neuron i goes to neuron j as a positive/negative signal is denoted by $p^+_{i,j}/p^-_{i,j}$. The signal leaves the network with probability d_i. When a neuron receives a positive signal, its potential is increased by 1; if it receives a negative signal or if it sends a signal, its potential decreases by 1. The lowest potential is 0. The flow of positive and negative signals arriving from the environment to neuron i is also a Poisson process of rate λ_i^+ and λ_i^- respectively. Finally, instead of working with branching probabilities $p^+_{i,j}$ and $p^-_{i,j}$, we use the neural network *weights* $w^+_{i,j} = r_i p^+_{i,j}$ and $w^-_{i,j} = r_i p^-_{i,j}$, in analogy to standard ANNs. In this context, let us define ρ_i as the limit probability in which neuron i is excited, which corresponds to a strictly positive potential:

$$\rho_i = \lim_{t\to\infty} \Pr\left(q_t(i) > 0\right) \tag{3}$$

Similar to the classical Jackson's result for open queuing networks, E. Gelenbe proved in [16] that this RNN model allows a simple system of equations with unique solution ρ_i, given the rates λ_i^+ and λ_i^- of incoming signals. In a traditional statistical learning application, a RNN with N interconnected neurons can be seen as a black-box, where the incoming signal rates $\boldsymbol{\lambda}^+ = \left(\lambda_1^+, \lambda_2^+, .., \lambda_N^+\right)$ and $\boldsymbol{\lambda}^- = \left(\lambda_1^-, \lambda_2^-, .., \lambda_N^-\right)$ are the inputs, and the probabilities of neuron excitement $\boldsymbol{\rho} = (\rho_1, \rho_2, .., \rho_N)$ are the outputs. As in most RNN applications, we shall consider that λ_i^- is 0 for every neuron. In this context, this black-box has a certain transfer-block $f(\cdot)$ that relates the N inputs with the N outputs:

$$\boldsymbol{\rho} = f\left(\boldsymbol{\lambda}^+\right) \tag{4}$$

where $f(\cdot)$ depends on the number of neurons N, the connection topology of the RNN, and the neural network weights $\mathbf{w} = \left\{w_{i,j}^+, w_{i,j}^-\right\}$. The weights $\mathbf{w}$ are thus the free parameters of the RNN model, which can be calibrated to build a non-linear transfer-block $f(\cdot)$ as the one we need.

In general, some λ_i^+ in (4) are set to 0, and only a subset of $\boldsymbol{\rho}$ is used as output. In our application (2), each block $f_k(\cdot)$ has n_k inputs and one single output. The n_k inputs correspond to the traffic volume of the $\boldsymbol{\delta}_k$ links that are crossed by OD flow k. The output is the OD flow volume $x_t(k)$. The calibration of each block $f_k(\cdot)$ is performed by supervised learning, using a *learning dataset* composed of T input-output pairs $\{Y_t(\boldsymbol{\delta}_k), x_t(k)\}\ ,\ t = 1, .., T$. We do not provide the details of the learning algorithm in this paper, but we refer the interested reader to [17].

Given that the output of the RNN ρ_o is a probability, we must scale the value of OD flow volume $x_t(k)$ to be consistent with the RNN model. We do this by simply normalizing $x_t(k)$ by the smallest link capacity of the n_k links that it crosses, defined as $c_{\delta_k^{\min}}$. In other words, we suppose that the routing process is always stable, in the sense that $y_t(i) < c_i, \forall i = 1, .., l$ and $\forall t$, even in the occurrence of strong congestion situations. We shall use $z_t(k) = x_t(k)/c_{\delta_k^{\min}}$ as the normalized volume of OD flow k, with $0 \leq z_t(k) \leq 1$. In a similar way, and even if the inputs in the model can take any arbitrary positive value, i.e. $\lambda_i^+ > 0$, we use links utilization values $U_t(\boldsymbol{\delta}_k) = Y_t(\boldsymbol{\delta}_k)/C(\boldsymbol{\delta}_k)$ as input instead of $Y_t(\boldsymbol{\delta}_k)$.

As in most applications of neural networks for learning purposes, we use a three-layers feed-forward network topology, which simplifies the RNN model and speeds-up computations. In such a topology, the set of N neurons is divided into three subsets: a set of I input neurons, a set of H hidden neurons, and a set of O output neurons. Input neurons receive positive signals from the environment and send signals to hidden neurons. Hidden neurons do no interact with the environment and only send signals to output neurons. Output neurons only send signals to the environment. It is easy to see that in this topology, the number of weights is $2H(I+1)$. The number of input neurons I is equal to n_k. This value is highly optimized in large-scale networks, so it is generally very small. For example, in our datasets, the mean number of links traversed by every OD flow is below 5. The number of output neurons is $O = 1$. The number of hidden neurons H is not a-priori fixed, and there is no foolproof method for setting it [14]. Too many degrees of freedom may cause over-fitting problems, and too few may reduce the *expressive* power to capture the underlying model. A convenient heuristic to choose H in an ANN is that the total number of weights is roughly $T/10$ [15]. The number of weights in a RNN is twice that of an ANN (negative weights), and thus this relation reduces to $T/5$.

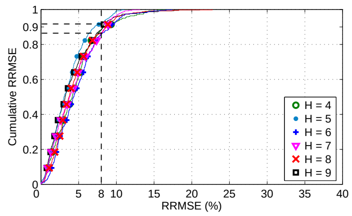
Using RNNs for TME has a paramount advantage w.r.t. ANNs as regards the choice of H. Similar to [13], we shall evidence the strong sensitivity of the three-layers feed-forward ANN model used in [11] to the number of hidden neurons H. We shall use real TMs and the real topology of the Abilene network, an Internet2 backbone network at the US. Abilene consists of 12 PoPs connected by 30 very-high-speed links, and $m = 132$ OD flows. Traffic consists of 5' sampled TMs, collected via Netflow [19]. We use exactly the same neural network topology and the same learning/estimation schemes for both the ANN and the RNN models.

We take 8 consecutive days of traffic from Abilene and divide it into two disjoint datasets. The *learning* dataset is used in the calibration of the models and it is composed of 24 hours of direct OD flow measurements, representing a total of 288 patterns. The *validation* dataset is used to verify the properties of the estimation methods and it is composed of 1 week of traffic, which accounts for 2016 measurements. As a global indication of the accuracy of the RNN/ANN-TME estimates, we use the relative root mean squared error $\text{RRMSE}(t)$:

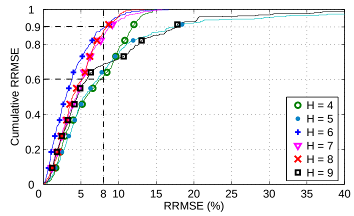
$$\text{RRMSE}(t) = \frac{\sqrt{\sum_{k=1}^{M}\left(x_t(k) - \hat{x}_t(k)\right)^2}}{\sqrt{\sum_{k=1}^{M} x_t(k)^2}}\ ,\ \ \forall t \in T_{\text{val}} \tag{5}$$

The $\text{RRMSE}(t)$ index has been used in previous works [7], [8] as a summary of the relative TM estimation error produced at every time t. In this sense, we shall refer to the $\text{RRMSE}(t)$ as the *temporal* estimation error. The value M corresponds to the number of OD flows that are compared in the $\text{RRMSE}(t)$ index. Small-volume OD flows are well known to be hard to estimate [5], [8], and are generally not considered in (5), simply because they have little impact on Traffic Engineering tasks, and so are generally less important to estimate. Following previous works [5], [8], we shall generally exclude from the $\text{RRMSE}(t)$ index about 5% of the total traffic, corresponding to these small OD flows.

Figure 1 depicts the cdf of the $\text{RRMSE}(t)$ for the 2016 TMs in the validation dataset, varying the mean number of hidden neurons $\bar{H}$ between 4 and 9. $\bar{H}$ is the rounded average of hidden neurons H_k used to build each transfer-block $f_k(\cdot)$, i.e., $\bar{H} = \text{round}\left(1/m\sum_{k=1}^{m} H_k\right)$. Using the rule presented above with $I = 4$ and $T = 288$ gives an expected value of $\bar{H}$ around 6, which justifies our choice of limits 4 and 9. Figure



(a) RNN model.



(b) ANN model.

Fig. 1. RRMSE(t) cdf, $\bar{H} = 4, \ldots, 9$ in Abilene.

1(a) shows that changing $\bar{H}$ around 6 has little impact on the RNN model. For example, for a relative error of about 8% (a reasonable value of RRMSE according to previous works [7], [8]), the variation of the cdf is below 6%. On the contrary, figure 1(b) shows that the cdf can vary almost a 30% for the same value of RRMSE in the ANN model, even for a change of only 1 hidden neuron, from 5 to 6 or from 8 to 9 for example. This strong sensitivity seriously limits the usefulness of the ANN-TME estimation method previously proposed in [11].

IV. Evaluation and Analysis

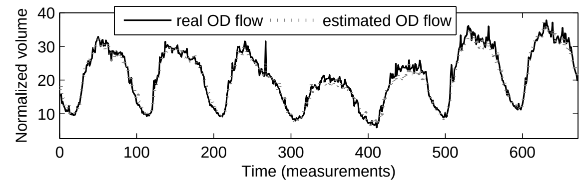
In this section we evaluate the RNN-TME method using real measurements from two different backbone networks. Firstly, we study the performance of the method facing two different traffic scenarios: the former corresponds to normal-operation traffic, the latter presents unexpected and large traffic variations. Secondly, we present a comparative analysis of the RNN-TME algorithm against six different TME techniques developed in previous works: our implementation of the ANN-TME method, the Simple-Gravity and Tomo-Gravity Estimation methods (SGE and TGE) [4], [5], our Spatial Maximum-Likelihood method (SMLE) [10], the PCA method (PCAE) [6], and an enhanced version of the Recursive Kalman Filter method (RKFE), proposed by us in [10].

A. RNN-TME in Normal Operation

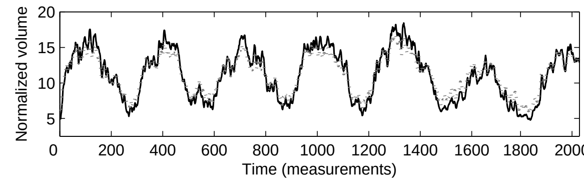
We shall consider the estimation of 1 week of normal-operation TMs from two operational networks: the Abilene network and the GEANT network, a European large-scale research network. The GEANT network consist of 23 aggregation nodes interconnected through 74 high-speed links, representing a total of $m = 506$ OD flows. GEANT data consists of 15' sampled TMs, built from IGP and BGP routing information and Netflow data in [18]. As before, we take 8 consecutive days of traffic, using the first 24 hours of measurements as the learning dataset, and the following 7 days for validation. Given that the sampling rate in GEANT is smaller than the one used in Abilene, we interpolate intermediate measurements in the learning dataset of the former topology. In the following evaluations, we assume that X_t is only known during the learning period of the RNNs models and consider Y_t as the input known data.

Figure 2 depicts the real and estimated values of the normalized volume of a single OD flow k, namely $z_t(k)$, for 1 week of traffic in 2(a) GEANT and 2(b) Abilene. In both cases the estimation is accurate and stable during the 7 days of the validation period. Figure 2(c) presents the cdf of the RRMSE for the validation week in both networks. The accuracy of the RNN-TME approach is quite impressive; the RRMSE(t) index is below 8% for more than 90% of the samples. The mean values of RRMSE(t) are 3.46% for GEANT and 4.22% for Abilene, comparable to those obtained in the literature, which may vary between 5% and 15% [7], [8].

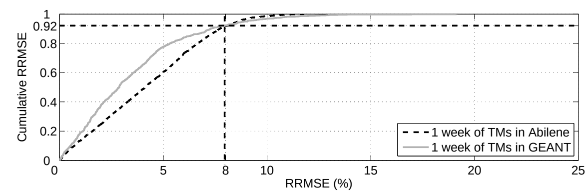
These results are a-priori quite impressive, especially because the learning period is probably not that long so as to cover all the possible input and output cases. However, the key issue in the learning process of each transfer-block $f_k(\cdot)$ is that, in fact, we are not learning any function but one very particular, which strongly depends on the structure of the routing matrix R. Each block $f_k(\cdot)$ is nothing but a particular pseudo-inverse of R_i. If we can correctly learn $f_k(\cdot)$ for a certain learning dataset, then this transfer-block should perform correctly even for new data, not seen before. The obvious drawback is that the performance of the method depends on the particular characteristics of R.



(a) OD flow 365 in GEANT



(b) OD flow 130 in Abilene



(c) RRMSE(t) cdf, 1 week of TMs in GEANT & Abilene

Fig. 2. RNN-TME estimation performance.

B. RNN-TME under Traffic Anomalies

Let us now consider the estimation of an OD flow in the presence of a large and abrupt traffic variation. Figure 3 depicts the normalized traffic volume of a single OD flow in Abilene. Before time 370 there is little traffic in this OD flow, but after this time a BGP egress-point shift causes a large and sustained volume increase during almost 18hs, until time 580. We use the first 24hs of normal-operation TMs to calibrate the RNN-TME, the RKFE, and the PCAE methods. We use one hour of SNMP measurements Y_t to calibrate the SMLE method [10].

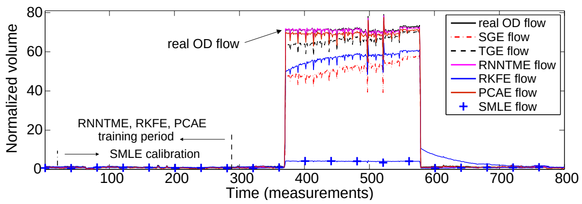


Fig. 3. OD flow analysis under anomalies.

The SGE and TGE methods do not require calibration.

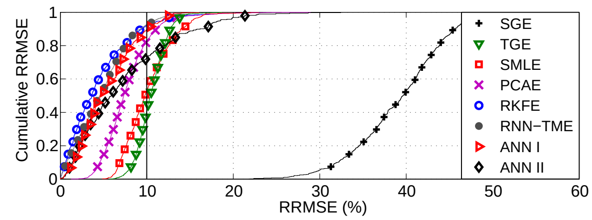
All methods properly track the OD flow volume evolution before the occurrence of the anomaly. However, only the RNN-TME and the PCAE methods achieve a proper estimation of the anomalous traffic. The reason for this success is that no traffic model is assumed by none of them. Very interesting is the tracking power achieved by the RNN-TME method, which is not a-priori justified, especially because no anomalous traffic patterns were present in the learning step. This evaluation permits to exemplify the strong dependence of the RNN-TME method on the particular structure of R that we mentioned before. In fact, one of the links that are crossed by the anomalous OD flow is only used by this particular OD flow, and thus the learning process results in this case in simply setting the RNN weights so as to copy at the output a scaled copy of the corresponding input. As regards the accuracy obtained by the PCAE method, we can say that the analyzed OD flow was captured in the space defined by the principal components. The rest of the methods impose assumptions on the underlying traffic that are clearly modified in the event of an anomaly.

C. Comparative Analysis

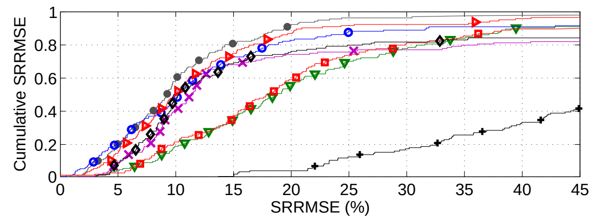
Figure 4 presents a comparative summary on the performance of all methods in Abilene, using a validation dataset composed of 672 consecutive TMs. The calibration of all methods is done as before. Figure 4(a) depicts the cdf of the RRMSE(t). The RNN-TME and the RKFE produce estimation relative errors below 10% for approximately 90% of the TMs. 80% is the approximated result obtained by the PCAE method. The estimation performance drops to nearly 55% for the SMLE method, 40% for the TGE method, and to 0% for the SGE method. The mean values of RRMSE are 4.95%, 4.48%, 6.53%, 10.3%, 11.2%, and 39.1% respectively. As regards the performance achieved by the ANN-TME method, we have included the best and worst-case results obtained when varying the mean number of hidden neurons $\bar{H}$ as in section III. It is easy to appreciate the same stability issues that were evidenced before. Figure 4(b) depicts the cdf of the Spatial RRMSE, defined as:

$$\text{SRRMSE}(k) = \frac{\sqrt{\sum_{t \in T_{\text{val}}} (x_t(k) - \hat{x}_t(k))^2}}{\sqrt{\sum_{t \in T_{\text{val}}} x_t(k)^2}}, \;\; \forall k = 1 \ldots m \quad (6)$$

The SRRMSE(k) index summarizes the error produced in the estimation of each single OD flow k over its lifetime.



(a) RRMSE(t) cdf



(b) SRRMSE(k) cdf

Fig. 4. 672 TMs and 132 OD flows in Abilene.

Spatial errors are fairly more spread than temporal errors, showing that some OD flows are more difficult to estimate than others. A deeper analysis of the SRRMSE shows that large SRRMSE values correspond to small OD flows. The RNN-TME estimation method outperforms the rest of the estimation algorithms, even for small OD flows, producing spatial errors below 20% for about 90% of the 132 OD flows. This is a direct consequence of the learning technique used in the method, where a single RNN is trained for each single OD flow, capturing its particular characteristics.

V. Conclusions

In this paper we have revisited the TME problem, introducing a novel method to accurately estimate a large-scale TM from aggregated measurements. There is an important effort of comparison in this paper, which categorically highlights the virtues of this new proposal. Using real data and a real large-scale network topology, we have shown that our method is more accurate than many other classical and modern approaches, regarding not only the estimation of a TM but also the analysis of individual OD flows. The RNN-TME method is particularly attractive to estimate small-volume OD flows, a task that other techniques can not even realize due to large errors. The last contribution of this work is related to a previous implementation of a Neural Network model to tackle the TME problem. Compared to the classical ANN model, our method based on RNNs is definitely more stable in the definition of the Neural Network topology, a critical issue regarding a reliable TME technique.

References

[1] Y. Vardi, "Network tomography: estimating source-destination traffic intensities from link data", in *J. Amer. Statist. Assoc*, 91, pp. 365-377, 1996.

[2] J. Cao et al, "Time-varying network tomography", in *J. Amer. Statist. Assoc*, 95, pp. 1063-1075, 2000.

[3] A. Medina et al, "Traffic Matrix Estimation: Existing Techniques and New Directions", in *ACM Sigcomm*, 2002.

[4] M. Roughan et al, “Experience in Measuring Backbone Traffic Variability: Models, Metrics, Measurements and Meaning”, in *ACM Sigcomm IMW*, 2002.
[5] Y. Zhang et al, “Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Load Measurements”, in *ACM Sigmetrics*, 2003.
[6] A. Lakhina et al, “Structural Analysis of Network Traffic Flows”, in *ACM Sigmetrics*, 2004.
[7] A. Soule et al, “Traffic Matrix Tracking using Kalman Filters”, in *LSNI*, 2005.
[8] A. Soule et al, “Traffic Matrices: Balancing Measurements, Inference and Modeling”, in *ACM Sigmetrics*, 2005.
[9] H. Ringberg et al, “Sensitivity of PCA for Traffic Anomaly Detection”, in *ACM Sigmetrics*, 2007.
[10] P. Casas et al, “Efficient Methods for Traffic Matrix Modeling and On-Line Estimation In IP Networks”, in *ITC21*, 2009.
[11] D. Jiang et al, “Large-Scale IP Traffic Matrix Estimation Based in Backpropagation Neural Network”, in *IEEE ICINIS*, 2008.
[12] H. Bakircioglu et al, “Survey of Random Neural Network Applications”, in *E.J.O.R.*, v. 126 (2), pp. 319-330, 2000.
[13] S. Mohamed et al, “A Study of Real-Time Packet Video Quality Using Random Neural Networks”, in *Trans. Circ. Syst. Video Tech.*, v. 12, pp. 1071-1083, 2000.
[14] C. Bishop, “Neural Networks for Pattern Recognition”, *Oxford Uni. Press*, 1995.
[15] R. Duda et al, “Pattern Classification”, *Wiley-Inter*, 2000.
[16] E. Gelenbe, “Random neural networks with negative and positive signals and product form solution”, in *Neural Computation*, v. 1, pp. 502-511, 1989.
[17] E. Gelenbe, “Learning in the Recurrent Random Neural Networks”, in *Neural Computation*, v. 5, pp. 154-164, 1993.
[18] S. Uhlig et al, “Providing Public Intradomain Traffic Matrices to the Research Community”, in *ACM Sigcomm CCR*, 2006.
[19] Y. Zhang, “Abilene Dataset 2004”, http://www.cs.utexas.edu/~yzhang, 2004.