# Traffic Matrix Prediction and Estimation Based on Deep Learning for Data Center Networks

Laisen Nie, Dingde Jiang*, Lei Guo
School of Computer Science and Engineering
Northeastern University
Shenyang 110819, China
* Email: jiangdd@mail.neu.edu.cn

Shui Yu
School of Information Technology
Deakin University
Burwood, VIC, 3125, Australia

Houbing Song
Department of Electrical and Computer Engineering
West Virginia University
WV 25136, USA

***Abstract*—Network traffic analysis is a crucial technique for systematically operating a data center network. Many network management functions rely on exact network traffic information. Although a great number of works to obtain network traffic have been carried out in traditional ISP networks, they cannot be employed effectively in data center networks. Motivated by that, we focus on the problem of network traffic prediction and estimation in data center networks. We involve deep learning techniques in the network traffic prediction and estimation fields, and propose two deep architectures for network traffic prediction and estimation, respectively. We first use a deep architecture to explore the time-varying property of network traffic in a data center network, and then propose a novel network traffic prediction approach based on a deep belief network and a logistic regression model. Meanwhile, to deal with the highly ill-pose property of network traffic estimation, we further propose a network traffic estimation method using the deep belief network trained by link counts. We validate the effectiveness of our methodologies by real traffic data.**

## I. Introduction

A data center has been a core technique in cloud computing. The management and planning of a data center network have received remarkable attention. Network traffic information is a crucial configuration input for network operators to carry out network management and planning in a data center network [1]–[4]. It can be succinctly represented by a traffic matrix (TM) that reflects the volume of traffic flows between all possible pairs of origin and destination (OD) nodes [5]–[8]. Besides, the decisions with respect to various network management tasks are made depending on different network traffic types. Network planning and predictive congestion control often use the future network traffic information as a configuration input. By contrast, traffic engineering and anomaly detection need to know the real-time status of network traffic.

The essential destination of TM prediction is obtaining an underlying predictor of the future network traffic via prior measurements [9] [10]. We denote a TM by an $N \times T$ matrix $X$ where each row of $X$ describes the time-varying property of an OD pair. We use $x_{n,t}$ to identify the $n$-th OD pair at time $t$. With $(x_{n,t}, x_{n,t-1}, x_{n,t-2}, ...)$ as the observations of traffic data up to time $t$, the TM prediction problem is defined as solving the predictor of $x_{n,t+1}$ (denoted by $\hat{x}_{n,t+1}$) from these observations. The main challenge here is how to extract (or model) the inherent and ubiquitous relationships between the observations so that one can exactly predict $x_{n,t+1}$. On the other hand, the fundamental problem of estimating a TM by the network tomography method is solving an inverse problem built by the link counts, the routing information and the TM [9]. If we denote link counts and routing information by $Y$ and $A$, then the network tomography model is

$$Y = AX. \tag{1}$$

The matrix $A$ is so-called routing matrix. If a data center network is built by $K$ nodes and $L$ links, the matrices $Y$, $A$ and $X$ are $L\times T$, $L\times N$ ($N = K^2$) and $N\times T$, respectively. Generally, link counts and routing information are obtainable for us. Link counts can be monitored by the simple network management protocol (SNMP) [9]. We can achieve the routing matrix $A$ from the configuration information of routers. Nevertheless, this inverse problem has a highly ill-posed property, because the number of OD pairs in a data center network is much larger than that of links. Hence, the main challenge for estimating traffic matrices is solving this inverse inference problem with the highly ill-posed property [9].

In traditional ISP networks, a great number of methods have been proposed for the TM prediction and estimation problems, such as the works in [9]–[13]. In fact, the methods of TM prediction and estimation have an inherent and implicit relationship. Most of TM estimation methods try to obtain a predictor of the TM by a statistical model at first, and then calculate an accurate estimator by recalibrating the initial predictor. These methods usually take into account the time-varying and spatio-temporal properties of a TM for modeling.

Unfortunately, these methods cannot be implemented in a center data network directly [1]. The statistical property of network traffic in a center data network is different from the traditional ISP network. In a traditional ISP network, the known statistical properties of network traffic contain self-similarity, long-range dependence, heavy-tailed distribution and so on [11] [14]. By contrast, the network traffic in a data center network exhibits much more fluctuations. These disordered fluctuations, which are insufficiently modeled by some simple models, result in a terrible predictor or estimator of the TM [1] [9]. Motivated by that, we propose two methods for TM prediction and estimation in a data center network,

978-1-5090-2482-7/16/$31.00 ©2016 IEEE

respectively. We use the deep learning theory [15]–[19] to explore the connotative properties of traffic flows.

The main contributions of this paper are as follows.

- We first use the deep learning method to explore the time-varying property of network traffic, and develop a novel network traffic prediction approach based on a deep belief network (DBN) architecture and a logistic regression model. Due to the unpredictable statistical property of network traffic, a faulty modeling has a negative effect on the accuracy of prediction. Instead of modeling an OD pair, we adopt a deep architecture to learn the statistical property of an OD pair. The proposed deep architecture can excavate the mutual dependence among traffic entries in various time slots. According to our simulations, the proposed method can capture the short timescale property of traffic flows faithfully.
- We also use a DBN architecture to solve the network tomography model. In our method, a deep architecture based on DBN is proposed to learn the ill-posed inverse inference system. In detail, we use some prior measurements of link counts and TM as the input and output of the proposed architecture. By training this architecture, it can extract the relationship between inputs and outputs. Therefore, we can estimate the TM using the corresponding link counts. The proposed method has a outstanding estimation bias.

The rest of this paper is organized as follows. We describe the related work of TM prediction and estimation in section 2. Section 3 introduces the deep learning and deep belief network. In sections 4 and 5, we will propose our TM prediction and estimation methods, respectively. We assess our methods in section 6, and then we conclude our work in section 7.

## II. Related Work

Statistical model techniques are usually performed in both TM prediction and estimation. Originally, some simple statistical models are used as additional information to deal with TM prediction and estimation, typically, Gaussian and Poisson [9]. The authors of [10] model traffic flows as a higher order Markov process, and join the Incremental Gaussian Mixture Model to estimate these traffic flows. Considering the various constructed properties of a TM (e.g., spatio-temporal and low-rank properties), many novel methods are proposed. For instance, in [9], the authors use the PCA method to acquire an approximated expression of TM for decreasing the level of the ill-posed property. The authors in [13] consider the spatio-temporal property of a TM, and propose a novel compressive sensing framework (termed as Sparsity Regularized Matrix Factorization) to estimate and predict it. In [11], the authors propose a probability model to estimate a Peer-to-Peer TM. Based on the PCA method, the authors in [12] propose a novel TM estimation method by minimizing the Mahalanobis distance.

As mentioned in the above, the methods of TM prediction and estimation have an inherent and implicit relationship. Hence, many TM estimation algorithms can perform TM prediction, e.g., the Kalman Filtering method [9] and the SRMF method.

These methods can predict and estimate a TM accurately in ISP networks. In a data center network, users acquire services (e.g., Infrastructure as a Service and Platform as a Service) by virtual machines [20], which refers to the distinction between ISP and data center networks in traffic characteristic. As a result, it is difficult to employ previous prediction and estimation methods in a data center network [1] [2]. Moreover, although many works have studied the characteristics of network traffic in data center networks, the TM prediction and estimation problems have not been attended enough.

## III. Deep Learning and Deep Belief Networks

Deep learning is an expanded technique compared with the shallow learning (i.e., artificial neural networks), which is a machine learning paradigm used to learn deep hierarchical models of data [15]. The DBN is one of the most prominent deep learning primitives [16] [17]. The fundamental ingredient of a DBN is the Restricted Boltzmann Machine (RBM). The framework of the RBM is shown in Fig. 1. It contains a visible layer and a hidden layer (denoted by $v$ and $h$) [16]. There are no connections between units of the same layer, but between visible units and hidden units. The DBN is made up of a series of RBMs (see Fig. 2).

For a RBM model, a jointly probability distribution function over visible and hidden units is defined as

$$p\left(v,h\right) = \exp\left(-E\left(v,h\right)\right) \Big/ \sum\nolimits_{v,h} \exp\left(-E\left(v,h\right)\right), \quad (2)$$

where $E\left(v,h\right)$ is termed as the energy function [14]. According to various probability distributions of visible units, the definition of the energy function $E\left(v,h\right)$ is diverse. The common distributions of visible units are Gaussian and Bernoulli. When the probability distributions of visible units and hidden units are Gaussian and Bernoulli, the energy function is

$$E\left(v,h\right) = -\frac{1}{2}\sum_{i=1}^{I}\left(b_i - v_i\right)^2 - \sum_{j=1}^{J} a_j h_j - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{i,j} v_i h_j, \quad (3)$$

where $I$ and $J$ are the numbers of visible and hidden units, respectively [17]. $b_i$ and $a_j$ are the biases of visible and hidden units. $w_{i,j}$ expresses the symmetric interaction term between the visible unit $v_i$ and the hidden unit $h_j$. Relatively, when the probability distributions of visible units are Bernoulli (the hidden units are consistently Bernoulli), the energy function is

$$E\left(v,h\right) = -\sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{i,j} v_i h_j. \quad (4)$$

Based on the above definitions, if the visible units are Gaussian or Bernoulli, then the conditional probabilities for hidden units is consistent, which can be defined as

$$P\left(h_j = 1|v\right) = sigm\left(a_j + \sum_{i=1}^{I} w_{i,j} v_i\right), \quad (5)$$

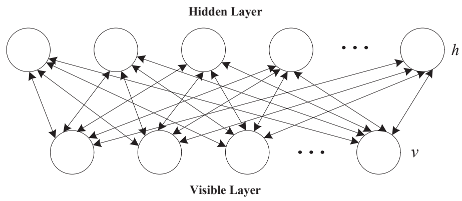


Fig. 1. Restricted Boltzmann Machine.

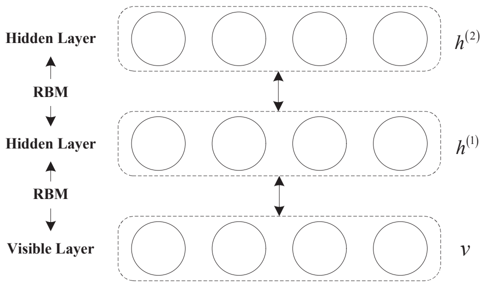


Fig. 2. DBN architecture.

where $sigm\,(z) = \exp(z)/(1+\exp(z))$ is the sigmoid function [14]. Similarly, consider the conditional probabilities for visible units, if the visible units are Gaussian, it can be calculated by

$$P\left(v_i|h\right) = N\left(b_i + \sum_{j=1}^{J} w_{i,j}h_j, 1\right), \quad (6)$$

where $N\left(b_i + \sum_{j=1}^{J} w_{i,j}h_j, 1\right)$ denotes the Gaussian distribution whose mean and variance are $b_i + \sum_{j=1}^{J} w_{i,j}h_j$ and 1 [14]. When the visible units are Bernoulli, the conditional probabilities for visible units are

$$P\left(v_i = 1|h\right) = sigm\left(b_i + \sum_{j=1}^{J} w_{i,j}h_j\right). \quad (7)$$

Compared with a shallow learning algorithm, the main challenge in deep learning is tremendous computational complexity [16] [17]. To train a deep model, a layer-wise greedy strategy should be employed. Considering a parameter (i.e., biases and weights of each RBM) update rule, the parameters are updated by computing the gradient of the negative log probability $-\log P\left(v\right)$.

## IV. Traffic Matrix Prediction

Traffic matrix prediction is defined as the problem that estimates the future network traffic from the previous and achieved network traffic data. In this section, we will propose a deep architecture for TM prediction. Then, by training the proposed deep architecture using a known network traffic data, it can predict the future network traffic. We assume that the known TM is $X$ whose entry is $x_{n,t}$. First, we here normalize the TM $X$ by dividing the maximum of $X$ so that all entries of $X$ are $[0,\,1]$. For simplicity, during the rest of this section, the TM is regarded as the normalized one without special statement.

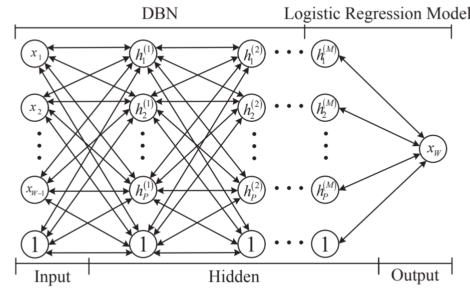


Fig. 3. DBN architecture for traffic matrix prediction.

**Deep architecture:** Figure 3 shows the proposed deep architecture. It consists of two parts, one is a DBN model, the other is a logistic regression model. The DBN that constitutes the hidden layer of the deep architecture has $M$ layers. There are $P$ units in each hidden layer. The input layer consists of $W-1$ units. The output layer takes advantage of a logistic regression model. The DBN carries out an unsupervised property learning for an OD pair. The logistic regression model viewed as an output layer is employed for prediction.

**Deep architecture learning:** According to the above deep architecture, we collect $W$ entries from an OD pair as the training data. The first $W-1$ entries are used as an input, similarly, the $W$-th entry is viewed as an output. The challenge of deep learning is computing the gradient of the negative log probability function, since the gradient consists of a sum of all states of the model [17]. We take advantage of the contrastive divergence algorithm proposed by Hinton [15] to approximately estimate this gradient instead of directly computing it. After learning this deep architecture, we can obtain the predictor $\hat{x}_{n,t+1}$ using $(x_{n,t}, x_{n,t-1}, .., , x_{n,t-W+1})$ as the input.

## V. Traffic Matrix Estimation

Traffic matrix estimation is solving the inverse inference system with ill-posed property shown in Eq. (1). In this paper, we will estimate the TM by the DBN introduced in the above section. In our estimation method, we first train the deep architecture by the priors of link counts and TM (denoted by $\tilde{Y}$ and $\tilde{X}$). We use the traffic data with $\tilde{T}$ time slots as the training data, and then the link counts $\tilde{Y}$ and the traffic matrix $\tilde{X}$ are $L\times\tilde{T}$ and $N\times\tilde{T}$. The deep architecture used to estimate the TM $X$ is shown in Fig. 4. It has $L$ input units, $N$ output units, and $M$ hidden layers. Each hidden layer has $P$ units. For the training data, each corresponding column of $\tilde{Y}$ and $\tilde{X}$ is regarded as an input and output pair of the deep architecture. If we denote each entry of $\tilde{Y}$ and $\tilde{X}$ by $\tilde{y}_{l,t}$ $(l = 1, 2, ..., L)$ and $\tilde{x}_{n,t}$ $(n = 1, 2, ..., N)$ , then the input and output pair is shown in Fig. 4. Notice that the training data is normalized by dividing the maximum of the link counts $\tilde{Y}$.

By training the model in Fig. 4, the deep architecture can learn the property of the inverse inference system, and

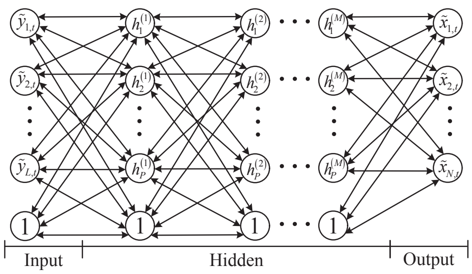


Fig. 4. DBN architecture for traffic matrix estimation.

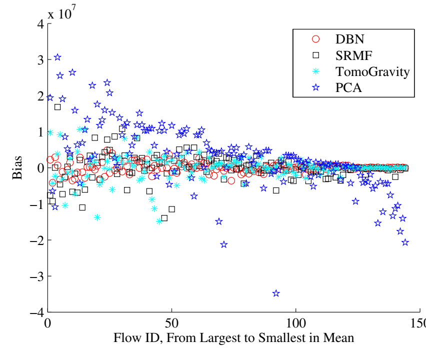


Fig. 5. Prediction Bias in data set 1.

then we can estimate the TM $X$ using $Y$ as the input data. Besides, the estimator of $X$ should obey the constraint of the inverse inference system, thus the Iterative Proportional Fitting algorithm [9] is employed in our method to correct the estimator of the TM.

## VI. Simulation Results and Analysis

In this section, we will validate the prediction and estimation properties of our methods using real network traffic data. To acquire a convictive and comprehensive evaluation, we collect two network traffic data sets. Both of them contain 2016 time slots. The tracks of traffic flows in the first data set are steep, and the second one is relatively smooth. In our simulation, we also compare our prediction and estimation methods with three state-of-the-art methods, that is, the SRMF, TomoGravity, and PCA methods.

### A. Prediction

This subsection will discuss the prediction effectiveness of our method. In our simulation for TM prediction, we set the parameters of DBN empirically, that is, the number of hidden layers is 8, and each hidden layer has 100 units. We first plot the prediction bias [9] of our method for two data sets in Figs. 5 and 6. The x-axis denotes the identity of each OD pair, which is arranged by descending order with respect to the means of traffic flows. The y-axis illustrates prediction bias. Figure 5 states that the biases of DBN, SRMF, and TomoGravity are decreased with the volume of traffic flows decreasing. By contrast, the biases of PCA are unordered, and some biases (i.e., for some small network traffic) are unsatisfactory. That is because the PCA method uses partial principal components to approximate a TM so that the network tomography model is an over-determined system. Hence, it performs poorly in bias for small network traffic. Specially, for the OD pairs with incisive jitters, the PCA method is not sensitive enough. Figure 5 declares that the most consistently unbiased method is DBN. The trends of traffic flows in the second data set are relatively steady. Compared with the bias in the first data set, four methods predict the traffic flows of data set 2 well as a whole (see Fig. 6). However, both DBN and PCA show a negative bias for small network traffic. For large network traffic, SRMF and TomoGravity appear a negative bias. By contrast, DBN and PCA yield negative and positive biases more or less.

A predictor with little bias is not equal to an effective one. Namely, once it has a high variance, an unbiased predictor also is viewed as an unavailable one. In other words, bias and variance need to be considered together for validating a predictor. Thus, we refer to the standard deviation [9] as a metric for variance to assess the effectiveness of our prediction method. Figures 7 and 8 plot the bias versus the standard deviation corresponding to two data sets. In general, to capture a fluctuation is much more difficult comparing with pursuing the trends of OD pairs. Therefore, we observe that the variance in data set 2 for the four methods is lower than that of data set 1. Figures 7 and 8 show that SRMF has the lowest variance for two data sets. Namely, SRMF prefers to predict an OD pair with short timescale variation. Contrarily, the PCA method tends to predict an OD pair over a long time interval. Besides, the variance of our method is relatively lower compared with PCA in data set 1 shown by Fig. 7. Our method deeply learns the statistical properties between the training data. Therefore, it is outstanding enough in predicting each traffic entry individually. In Fig. 8, our method exhibits relatively high variance. Hence, DBN obtains a different tradeoff between bias and variance.

Finally, we will overall compare our method with the other three methods using the error improvement ratio [7]. From Fig. 9, the error improvement ratios of our method against SRMF, TomoGravity, and PCA are 53.1%, 47.9%, and 83.6% in the first data set. Similarly, they are 9.8%, 5.7%, and 52.4% in the data set 2. Judging from the above results, it is clear that DBN is good at pursuing the fluctuations of network traffic.

### B. Estimation

In this subsection, we will assess the estimation property of our method. In these simulations, similarly, let the number of hidden layers be 12, and there are 100 hidden units in each hidden layer. Likewise, we also first explore the bias of our method dealing with the TM estimation problem. According to Fig. 10, with the volume of traffic flows increasing, the biases of four methods are increased. We find that our method shows the lowest estimation bias for the data set 1. The PCA method has some high biases for small network traffic once again. In Fig. 11, we observe that SRMF and TomoGravity

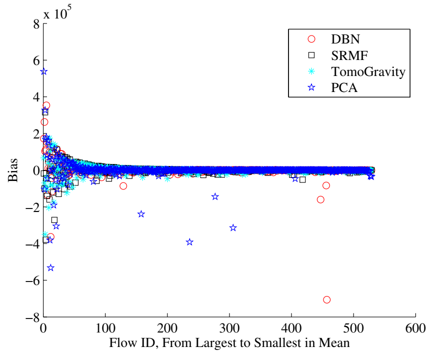


Fig. 6. Prediction Bias in data set 2.

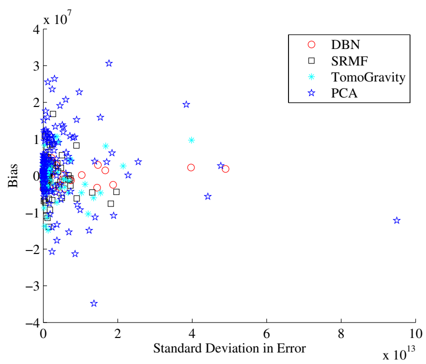


Fig. 7. Bias versus standard deviation in data set 1.

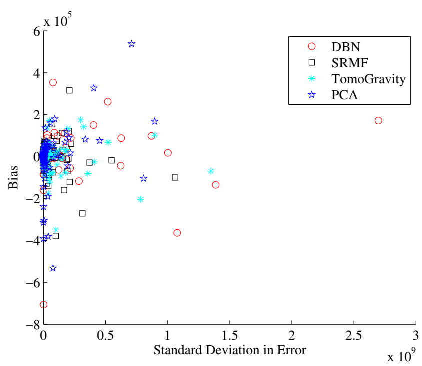


Fig. 8. Bias versus standard deviation in data set 2.

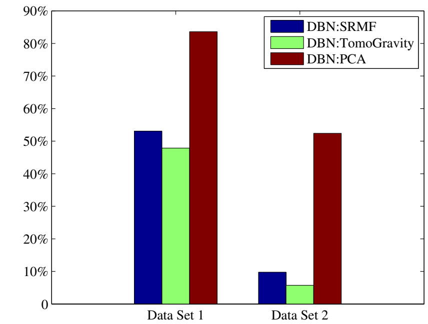


Fig. 9. Error improvement ratio.

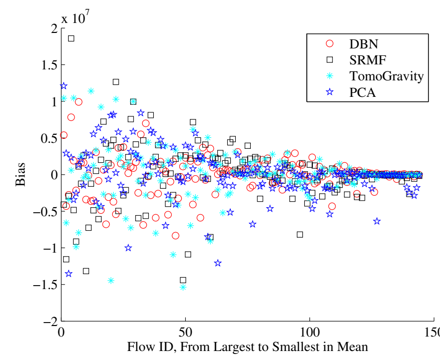


Fig. 10. Estimation Bias in data set 1.

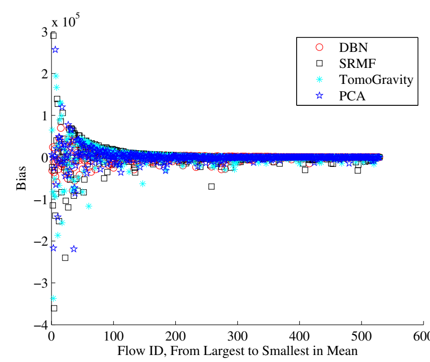


Fig. 11. Estimation Bias in data set 2.

show some significantly high biases with both positive and negative biases. PCA usually appears some negative biases.

The standard deviation is displayed by Figs. 12 and 13. PCA has the largest variance in both data sets. To most entries of the TM, the variance of DBN, TomoGravity, and SRMF is relatively lower. Hence, in TM estimation, it confirms that our method tends to estimate a short timescale OD pair. As shown by Fig. 14, the error improvement ratios of DBN against SRMF, TomoGravity and PCA are about 32.7%, 25.3%, and 23.4% in turn. They are 47.7%, 45.7%, and 72.0% in data set 2. The above simulation results prove that our method is a powerful method not only for prediction but also for estimation. It remarkably outstands in prediction and estimation biases.

## VII. Conclusion

This paper focuses on the problems of TM prediction and estimation in data center networks. Considering the various properties of traffic flows, we use deep learning techniques to capture the properties of network traffic. Then we first propose a TM prediction method based on a deep belief network and a logistic regression model. In terms of a deep belief network architecture, we further propose an effective deep architecture for TM estimation. We use this deep architecture to learn the network tomography model for TM estimation. Finally,

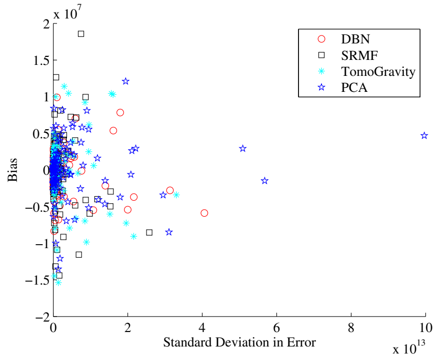


Fig. 12. Bias versus standard deviation in data set 1.

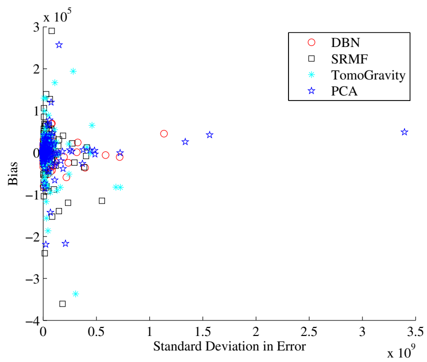


Fig. 13. Bias versus standard deviation in data set 2.

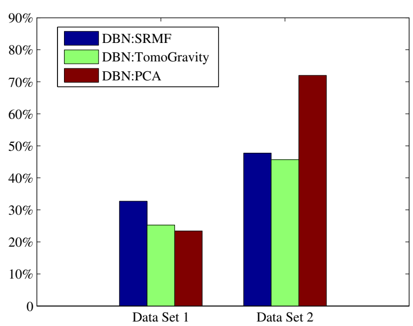


Fig. 14. Error improvement ratio.

we assess the effectiveness of the proposed prediction and estimation methods by real network traffic data.

The complexity of learning a deep architecture is a challenge. Therefore a reasonable deep architecture built by considering TMs' characteristic and an effective learning algorithm are essential to improve the complexity, which is the main work in the future.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Nos. 61571104, 61071124), the General Project of Scientific Research of the Education Department of Liaoning Province (No. L20150174), the Program for New Century Excellent Talents in University (No. NCET-11-0075), the Fundamental Research Funds for the Central Universities (Nos. N150402003, N120804004, N130504003), and the State Scholarship Fund (201208210013). The authors wish to thank the reviewers for their helpful comments.

## REFERENCES

[1] Z. Hu, Y. Qiao, and J. Luo, "Coarse-grained traffic matrix estimation for data center networks," *Computer Communications*, vol. 56, no. 2, pp. 25–34, 2016.

[2] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of ACM IMC*, 2010, pp. 267–280.

[3] Z. Ning, Q. Song, L. Guo, Z. Chen, and A. Jamalipour, "Integration of scheduling and network coding in multi-rate wireless mesh networks: Optimization models and algorithms," *Ad Hoc Networks*, vol. 36, no. 1, pp. 386–397, 2016.

[4] Z. Ning, L. Liu, F. Xia, B. Jedari, I. Lee, and W. Zhang, "Cais: A copy adjustable incentive scheme in community-based socially-aware networking," *IEEE Transactions on Vehicular Technology*, 2016.

[5] D. Jiang, Z. Yuan, P. Zhang, L. Miao, and T. Zhu, "A traffic anomaly detection approach in communication networks for applications of multimedia medical devices," *Multimedia Tools and Applications*, 2016.

[6] H. Ma, X. Zhang, and A. Ming, "A coverage-enhancing method for 3d directional sensor networks," in *IEEE INFOCOM 2009*, 2009, pp. 2791–2795.

[7] D. Jiang, Z. Xu, and H. Xu, "A novel hybrid prediction algorithm to network traffic," *Annals of Telecommunications*, vol. 70, no. 9, pp. 427–439, 2015.

[8] H. Ma and Y. Liu, "On coverage problems of directional sensor networks," in *The First International Conference on Mobile Ad-hoc and Sensor Networks*, 2005, pp. 721–731.

[9] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, "Traffic matrices: balancing measurements, inference and modeling," in *Proceedings of SIGMETRICS 2005*, 2005, pp. 362–373.

[10] D. Kumlu and I. Hokelek, "Network traffic estimation using markov chain and incremental gaussian mixture," in *Proceedings of 2015 23th Signal Processing and Communications Applications Conference*, 2015, pp. 1187–1190.

[11] K. Xu, M. Shen, Y. Cui, M.Ye, and Y. Zhong, "A model approach to the estimation of peer-to-peer traffic matrices," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1101–1111, 2014.

[12] E. Zhao and L. Tan, "A pca based optimization approach for ip traffic matrix estimation," *Journal of Network and Computer Applications*, vol. 57, no. 2015, pp. 12–20, 2015.

[13] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices (extended version)," *IEEE Transactions on Networking*, vol. 20, no. 3, pp. 662–676, 2012.

[14] D. Jiang, C. Yao, Z. Xu, and W. Qin, "Multi-scale anomaly detection for high-speed network traffic," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 3, pp. 308–317, 2015.

[15] G. E. Hinton, S. Osindero, and Y. The, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[16] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 2, pp. 1–127, 2009.

[17] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 2-3, pp. 197–389, 2014.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, p. 504C507, 2006.

[19] C. L. P. Chen, C. Zhang, L. Chen, and M. Gan, "Fuzzy restricted boltzmann machine for the enhancement of deep learning," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2163–2173, 2015.

[20] M. Dong, H. Lit, K. Ota, and H. Zhu, "Hvsto: Efficient privacy preserving hybrid storage in cloud data center," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2014, pp. 529–534.