A SDN-Based Network Traffic Estimating Algorithm in Power Telecommunication Network

Renxiang Huang[1(✉)], Huibin Jia[2], and Xing Huang[3]

[1] Sichuan University of Arts and Science, Dazhou 635000, China
wanddy@gmail.com
[2] School of Electrical and Electronic Engineering, North China Electric Power University, Baoding, China
[3] State Grid Liaoning Electric Power Company, Shenyang, China

Abstract. Most of network management tasks in traffic engineering such as traffic scheduling, path planning, both of them are required the accurate and fine-grained network traffic. However, it is difficult to capture and estimate the volume of network traffic due to its time-varying nature. In this paper, we study the network traffic estimation scheme to estimate the fine-grained network traffic. Firstly, the network traffic is constructed as a time series and the autoregressive moving average (ARMA) method is used to characterize and model network traffic. Secondly, in order to decrease the estimation errors of the ARMA model, we use the optimization theory to adjust the estimation results. We construct an objective function with constraints. We find that objective function is an NP-hard problem, then we introduce a heuristic algorithm to find the optimization results. Finally, to evaluate the performance of our proposed scheme, we construct a simulation platform and compare our scheme with that of the other methods in an SDN simulation platform. The simulation results indicate that our approach is effective and our method can reflect the network traffic characteristics.

Keywords: Network traffic · Traffic estimation · Software-defined network · Optimization · ARMA

1 Introduction

With the rapid growth of applications in the power telecommunication network, network performance and quality of service issues are increasing. In the case of limited network resources, establishing a network traffic model, predicting network load, and timely controlling or adjusting will greatly improve network performance and service quality [1, 2]. Software defined network SDN is a new network innovation architecture which is an implementation of network virtualization [3]. SDN separates the control plane of the network device from the data plane and centralizes the control plane into the controller for centralized management. The controller is the brain of the network and has the global view of the network, and then it is flexible control the network traffic and makes the network more intelligent as a pipeline, providing a good platform for

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 94–103, 2019.
https://doi.org/10.1007/978-3-030-32216-8_9

innovation of core networks and applications. For time-frequency synchronization applications in (SDN), the network traffic, especially end-to-end network traffic in the network, represents the network-level behavior of users and applications. In the network, the network traffic from the origin node to the destination node is called an OD pair. There are many OD pairs in the network and the traffic for each OD pair directly affects the performance of the SDN. However, the network traffic in networks is difficult to be estimated and predicted due to their high variability over time. Therefore, network traffic estimation has become one of the hottest topics and has received increasing attention [4].

Liu et al. proposed two iterative algorithms to estimate TM between tomogravity space and gravity space, and use similar-Mahalanobis distance as a metric to control estimation errors in DCN(Data center network) [5]. Hashemi et al. presented a real-time traffic network state estimation and prediction system with built-in decision support capabilities for traffic network management [6]. Kawasaki et al. proposed a state-space model that estimates traffic states over a two-dimensional network with alternative routes available by a data assimilation technique that fuses probe vehicle data with a traffic flow model [7]. Dias et al. presented a classification module for video streaming traffic, based on machine learning, as a solution for network schemes that require adequate real-time traffic treatment [8]. Nie et al. propose a novel network traffic prediction approach based on a deep belief network [9]. Ermagun et al. studied examines the spatiotemporal dependency between traffic links and model the traffic flow of 140 traffic links in a sub-network of the Minneapolis-St [10]. Jiang et al. investigated how to estimate and recover the end-to-end network traffic matrix in fine time granularity from the sampled traffic traces which is a hard inverse problem [11]. Some of these methods had relatively large estimation errors, while others were very sensitive to prior information [6, 11]. Hence, the above models and methods are difficult to accurately capture network flow traffic, so it is still significantly necessary to find more accurate model to depict network flow traffic, to lower the complexity of algorithms, and to improve the estimation accuracy.

Different from these algorithms, this paper proposes a new estimation approach to model the network traffic in power telecommunication network. Firstly, the network traffic is described as linear-correlation random process over time and constructed as a time series. Then, we use the autoregressive moving average (ARMA) to characterize and model network traffic. Secondly, the ARMA model is trained to describe network traffic changes over time. Additionally, network traffic sample data are used to establish and determine the model parameters. In such a case, the ARMA model can be effectively and correctly capture the dynamic nature of network traffic in power telecommunication networks. We can effectively estimate network traffic in the next time. Then, we construct an objective function with constraints. We find that objective function is an NP-hard problem, then we introduce a heuristic algorithm to find the optimization results. Finally, to evaluate the performance of our proposed scheme, we construct a simulation plat-form and compare our scheme with that of the other methods in an SDN simulation platform.

The rest of this paper is organized as follows. Section 2 is a problem statement. Section 2.1 is to derive our prediction approach. Section 3 is simulation results and analysis. Finally, our work in this paper is concluded in Sect. 4.

2 Problem Statement

Origin-Destination (OD) traffic refers to traffic between two nodes in the network. Given the training set $D: \{X_i, t_i\}_{i=1}^{N}$ as the network traffic in power telecommunication networks, where X_i is the number of training samples, X_i is the vector of network traffic corresponding to time t_i, then the network traffic can be represented as

$$X = \{x(t_1), x(t_2), \ldots., x(t_N)\} \tag{1}$$

The flow traffic in the network is aggregated into the link on the transmitting path, then the relationship between link load and traffic can be expressed as that

$$Y = AX \tag{2}$$

where Y is a column vector representing link traffic, X is also a column vector representing the traffic matrix and A is the routing matrix. The problem of flow calculation is an inverse problem solving of an underdetermined and ill-conditioned system.

In the network, the flow traffic in networks can be presented as a time-series model and has time correlation. The autoregressive moving average (ARMA) model is used to predict the time series; it consists of the autoregressive (AR) model and the moving average (MA) model. However, ARMA is more widely used and has lower prediction errors than AR model and MA model. The AR model presents the correlation of flow traffic in time, so the traffic sequence $x(1), x(2), \ldots, x(t)$ of a flow can be written as

$$x(t) = \sum_{i=1}^{p} \phi_i x(t-i) + Z(t) \tag{3}$$

where $x(t-i)$ is the observed value of the predicted object, $Z(t)$ is the error; $\phi_i (i = 1, 2, \ldots, p)$ are the autoregressive coefficients; p is the order. As the prediction object $x(t)$ is affected by its own change. The error $Z(t)$ is the white noise, it is a random sequence. The MA model of random error can be expressed as

$$\begin{aligned} Z(t) &= u(t) + \theta_1 u(t-1) + \theta_2 u(t-2) + \cdots \theta_q u(t-q) \\ &= u(t) + \sum_{i=1}^{q} \theta_i u(t-i) \end{aligned} \tag{4}$$

where $u(t)$ is the white Gaussian noise, so the mean and variance of $u(t)$ are $E(u(t)) = 0$ and $E(u(t)^2) = \sigma^2$, respectively; q is the moving average order; $\theta_j (j = 1, 2, \ldots, q)$ are the moving average coefficients. Then, the ARMA(p, q) model can be written as

$$x(t) = \sum_{i=1}^{p} \phi_i x(t-i) + u(t) + \sum_{i=1}^{q} \theta_i u(t-i) \tag{5}$$

The accuracy of the ARMA(p, q) prediction is determined by the order p and q. When $q = 0$, the ARMA model becomes the AR model; and when $p = 0$ the ARMA model degrades the MA model. Then, we introduce the AIC (Akaike Information Criterion) principle and BIC (Bayesian Information Criterion) principle to determine the order of the ARMA model. The AIC criterion is a weighting function of the fitting precision and the number of orders, and the model that makes the AIC function minimum is considered to be the optimal model. Define the AIC criterion function as follow:

$$AIC = N\log\hat{\sigma}^2 + 2(p + q + 1) \tag{6}$$

$$BIC = AIC + (\log(N) - 2)(p + q + 1) \tag{7}$$

where N is the number of sampling points; $\hat{\sigma}^2$ is the variance of the filling residual. Then, we take the order of the best ARMA(p, q) model.

In the SDN-based network, we use the pull-based scheme to collect coarse-grained network traffic statistic. We use the ARMA(p, q) model to predict the network traffic

$$\hat{x}(t) = \mathrm{ARMA}(x(t)) \tag{8}$$

With the ARMA model, we estimate traffic with the measured time series $x(t)$. However, estimation results of flows have big errors with the actual flow traffic. In the network, the link load reflects the integrated traffic transmission in the network. So, we use the pull-based method to obtain the fine-grained link load Y in networks. We try to decrease the network traffic error, the objective function can be written as

$$f = \left\|Y - A\hat{X}\right\|_2 + \left\|\hat{X}\right\|_2 \tag{9}$$

In order to decrease the deviation between the estimations and the actual traffic results, we construct an objective function to optimize the estimation results. The objective function with constraints as

$$\begin{cases} \min f \\ s.t. \\ C1: X \geq 0 \\ C2: Y_m \geq \sum\limits_n a_{mn}\hat{X} \\ C3: \sum_{i=1}^{N} x_{ij} = \sum_{j=1}^{N} x_{ji} \end{cases} \tag{10}$$

where $\hat{X}$ is estimated by the ARMA model. Constraint $C1$ shows the link load is non-negativity; constraint $C2$ is a limitation of flows on each link; constraint $C3$ represents that the traffic that input and output a switch are constants, i is the source node and j is the destination node. In the network, the routing matrix A has M rows and N columns. However, the OD pairs are much larger than links, namely: $M \ll N$, then the routing matrix A is an underdetermined matrix, therefore, there are infinite traffic matrices X.

The objective function (10) is an NP-hard problem and is difficult to solve directly. Then, we use a heuristic method to solve it.

2.1 Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) algorithm is one of the heuristic algorithms which utility the swarm intelligence computational model based on the natural swarm systems. The particle swarm optimization is an optimization technique based on the sociological behavior associated with birds flocking, which is a population-based stochastic optimization technique, so it is suitable to solve the non-linear optimization problem. The particle swarm optimization algorithm is a robust swarm optimization method which dynamically adjusts according to the particle movement velocity and the particle companions' status.

PSO is initialized with a population of random solutions and searches for optimal by updating particles' positions. The velocity of particles is influenced by three components namely, initial, cognitive and social components. Each particle updates its previous velocity and position vectors according to the following model.

$$\begin{aligned} v_k(t+1) = wv_k(t) + c_1 r_1 (Pbest_k(t) - x_k(t)) \\ + c_2 r_2 (Gbest(t) - x_k(t)) \end{aligned} \tag{11}$$

$$x_k(t+1) = x_k(t) + v_k(t+1) \tag{12}$$

which $x_k(t+1)$ and $v_k(t+1)$ represent the particle position and the particle moving velocity respectively. The term c_1 and c_2 denote the personal and global learning factors respectively which are also defined as constants. $Pbest_k$ and $Gbest$ are the personal best and global best of each particle respectively. r_1 and r_2 are both random values in the range $[0, 1]$. The term w is the inertia weight.

For PSO, the personal best status and the goal best statue are the two terms which should be shared among all the particles. We assume that there are K particles in the swarm, due to that there are N flows in the network, so each particle has expressed a vector with the set of flows, then each particle can be written as the vector $x_k = [x_{k1}, \ldots, x_{kn}, \ldots x_{kN}]$ for each particle position and the velocity vector can be written as $v_k = [v_{k1}, \ldots, v_{kn}, \ldots v_{kN}]$. Each particle flying based on its personal best status and the global best status during each iteration.

3 Simulation Result and Analysis

3.1 Simulation Environment

In this section, we perform some simulations to evaluate the performance of our proposed algorithm AMPSO. In order to justify the performance of our method, we construct a simple network topology with Mininet, and use Ryu as a controller. We use Iperf to generate some origin-destination (OD) pairs and measure the traffic at different nodes deployment at different places in the network. The PCA, SRSVD [12], WABR [8] are the methods studied has better performance. Here we compare AMPSO against them. The mean absolute error (MAE), mean relative error (MRE) for the network in traffic are for different methods. Finally, we discuss the performance improvement of AMPSO against PCA, SRSVD, WABR. In our simulation, we use the first 300 time slots as the training set to train the prediction model, and then we embed the prediction model into optimization module and insert them into the controller to measure the network traffic and validate the performance of all algorithms.

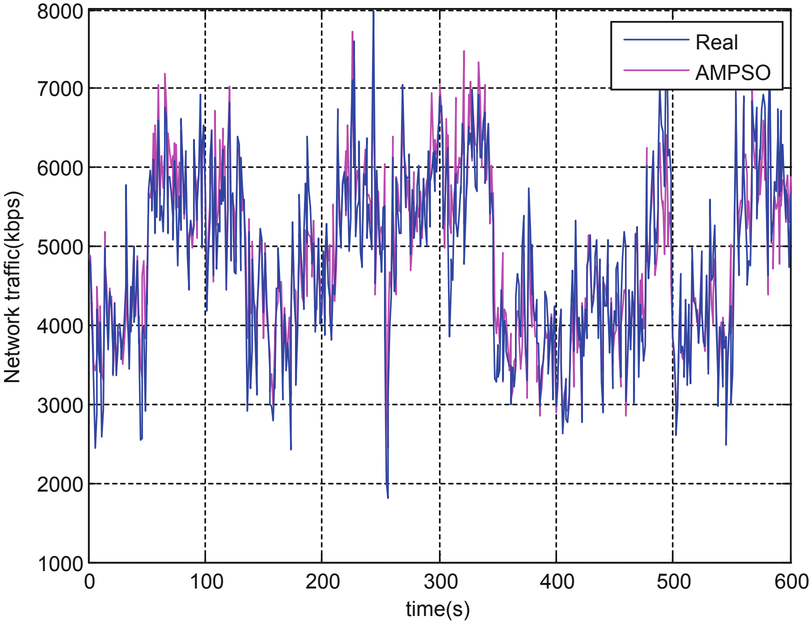

Fig. 1. Measurement results of network traffic.

Figure 1 shows the prediction results of network traffic flows, where network traffic flows f is selected randomly from the origin-destination (OD) pairs in the network. As our simulation tests, other OD traffic pairs holds similar results. Without loss of generality, we only discuss the network traffic flows f1 in this paper. In Fig. 1, we find that the network traffic flow is fluctuation over time as the blue line in Fig. 1. The network traffic estimation results of AMPSO can catch the trend of network traffic. Next, we will further discuss the performance of our algorithm, and compare our method against other algorithms. Although network traffic in Fig. 1 can intuitively reflect the characteristics of network traffic, it is difficult to observe the fluctuation characteristics of network traffic in detail. To further verify the performance of the proposed algorithm, we use the indicators MAE and MRE to analyze the estimation error of the network traffic. We repeated the simulation 100 runs to avoid the randomness of the simulation process.

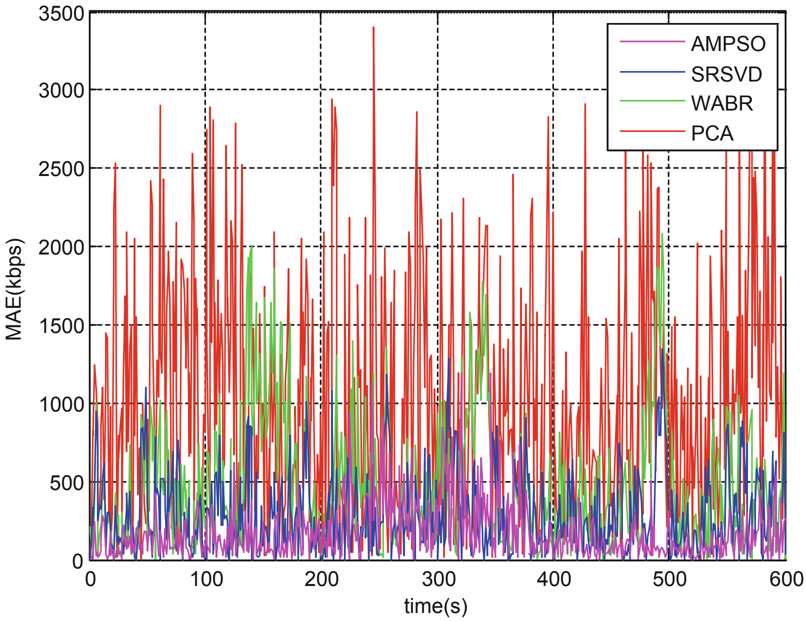

Fig. 2. Mean absolute errors for network traffic.

The mean absolute errors and mean relative errors over time for the network traffic are defined as:

$$\mathrm{MAE} = \frac{1}{K}\sum\nolimits_{i=1}^{K} |\hat{x}_i(t) - x_i(t)| \tag{13}$$

$$\mathrm{MRE} = \frac{1}{K}\sum\nolimits_{i=1}^{K} \frac{|\hat{x}_i(t) - x_i(t)|}{x_i(t)} \tag{14}$$

where $i = 1, 2, \ldots, K$, K indicates the number of repetitions in the simulation process, and $\hat{x}_i(t)$ indicates the network traffic measurement, and $x_i(t)$ is the actual network traffic generate by Iperf in the network.

The mean absolute errors of the network traffic over time are shown in Fig. 2. We can find that for network traffic WABR and SRSVD exhibit lower relative errors while PCA holds the larger prediction bias. For Fig. 2, we can also see that that SRSVD holds the lowest relative errors. This shows us that in contrast to PCA, WABR and SRSVD, AMPSO holds a better performance of the network traffic prediction, while AMPSO holds the best prediction ability. We can also find that the AMPSO has the lowest fluctuation over time in terms of mean absolute errors than the other three algorithm, and has mean absolute error is smallest than the other three algorithms. AMPSO can more effectively model the network traffic with time-varying and correlation features.

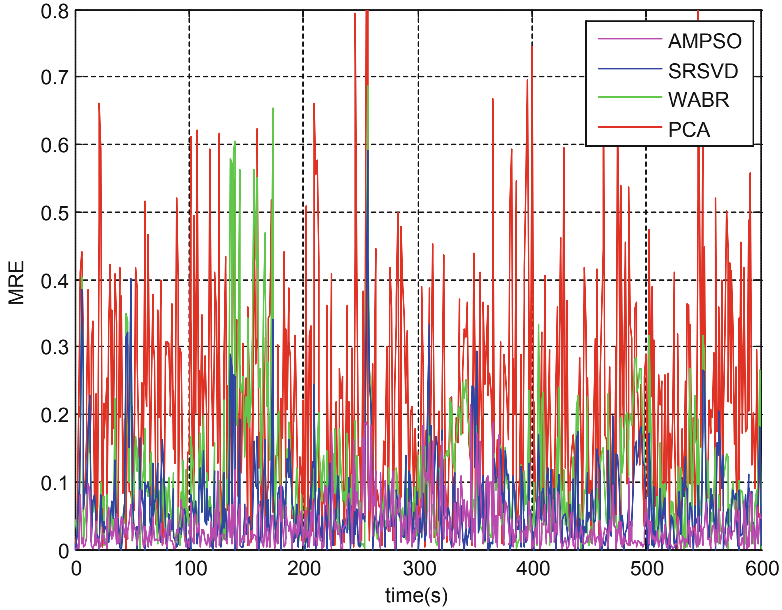

Fig. 3. Mean relative errors for network traffic.

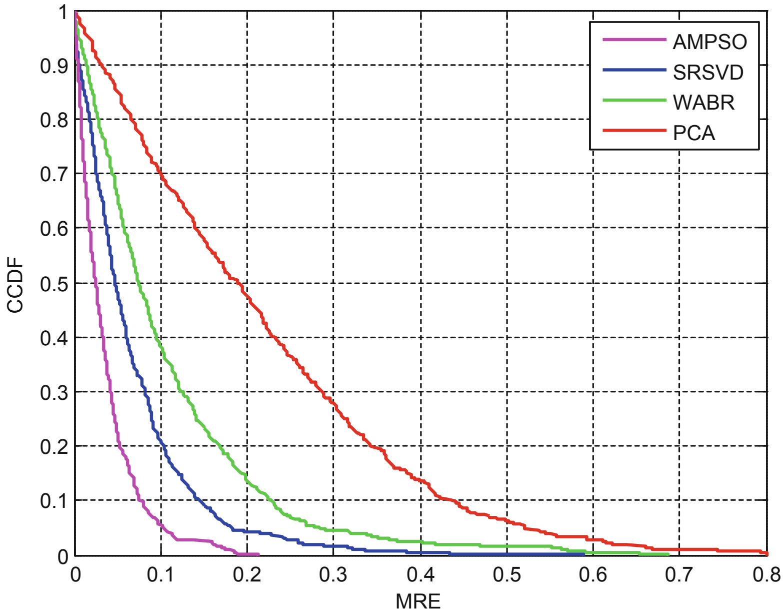

Fig. 4. The CCDF of the mean relative errors for network traffic.

Figure 3 depicts the mean relative errors of the network traffic. The mean relative errors reflect the ratio of estimation errors. Figure 3 shows that AMPSO has the lowest

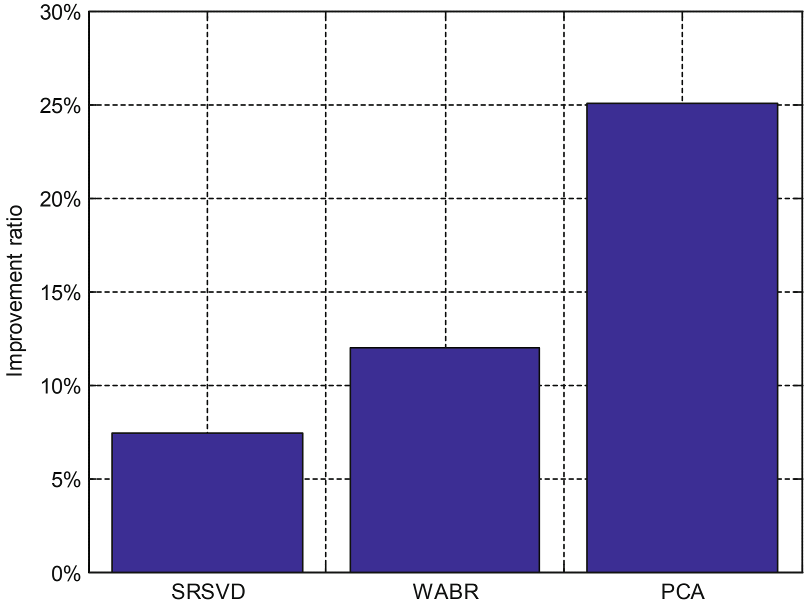

Fig. 5. Improvement ratio of network traffic.

mean relative errors than that of the PCA, WABR and SRSVD, and most of the mean relative errors of AMPSO is smaller than 0.1, which means that the estimation error is smaller than 10%. Then, the AMPSO is effective to estimate the network traffic in the power telecommunication networks.

Figure 4 depicts the curve of the CCDF (Complementary Cumulative Distribution Function) of mean relative errors for the different measurement methods. Mean relative errors are the standard deviation of the residuals, it shows how concentrated the data is around the line of actual flow traffic. The CCDF of the measurement Mean relative errors error in Fig. 4 reflects that 10% mean relative errors of the flow of the AMPSO, SRSVD, WABR, PCA is more than 0.078, 0.146, 0.229 and 0.435, respectively. So, the network traffic measurement scheme of AMPSO is stable and can reflect the network traffic with the mean relative error lower than 0.078.

Now, we analyze the performance improvement of AMPSO relative to the other three algorithms for the network traffic. Figure 5 exhibits the performance improvement ration of network traffic flow. In Fig. 5, AMPSO attains the performance improvement of about 7.4%, 12%, 25% against SRSVD, WABR, PCA, respectively. This clearly denotes that compared with PCA, WABR, and SRSVD, our algorithm can more accurately model the network-level network traffic. Moreover, Fig. 5 also tell us that relative to PCA and WABR, our scheme can reach larger performance improvement.

4 Conclusions

This paper uses the ARMA method to model network traffic in power telecommunication networks. By the ARMA model, we can capture the dynamic and time-varying features over time of the network traffic. Network traffic is converted into a time series which can be predicted by the ARMA model with some history data. Then we use the optimization theory to decrease the estimate errors. Because the objective function of the optimization process is an NP-hard problem, we propose to use a heuristic algorithm to find the solution. Then, we introduce the PSO to optimize the network traffic. Finally, we perform some simulation to verify the performance of the proposed algorithm in this paper. Simulation results show that the proposed approach in this paper is feasible.

References

1. Jiang, D., Xu, Z., Liu, J., et al.: An optimization-based robust routing algorithm to energy-efficient networks for cloud computing. Telecommun. Syst. **63**(1), 89–98 (2016)
2. Jiang, D., Nie, L., Lv, Z., et al.: Spatio-temporal Kronecker compressive sensing for traffic matrix recovery. IEEE Access **4**, 3046–3053 (2016)
3. Guo, Y., Wang, Z., Yin, X., et al.: Traffic engineering in hybrid SDN networks with multiple traffic matrices. Comput. Netw. **126**, 187–199 (2017)
4. Jiang, D., Zhao, Z., Xu, Z., et al.: How to reconstruct end-to-end traffic based on time-frequency analysis and artificial neural network. AEU-Int. J. Electron. Commun. **68**(10), 915–925 (2014)
5. Liu, G., Guo, S., Zhao, Q., et al.: Tomogravity space based traffic matrix estimation in data center networks. Transp. Res. Part C: Emerg. Technol. **86**, 39–50 (2018)
6. Hashemi, H., Abdelghany, K.F., et al.: Real-time traffic network state estimation and prediction with decision support capabilities: application to integrated corridor management. Transp. Res. Part C: Emerg. Technol. **73**, 128–146 (2016)
7. Kawasaki, Y., Hara, Y., Kuwahara, M.: Traffic state estimation on a two-dimensional network by a state-space model. Transp. Res. Part C: Emerg. Technol. **5**, 1–17 (2019)
8. Dias, K.L., Pongelupe, M.A., Caminhas, W.M., et al.: An innovative approach for real-time network traffic classification. Comput. Netw. **158**, 143–157 (2019)
9. Jiang, D., Huo, L., Lv, Z., et al.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **pp**(99), 1–15 (2018)
10. Ermagun, A., Levinson, D.: Spatiotemporal short-term traffic forecasting using the network weight matrix and systematic detrending. Transp. Rese. Part C: Emerg. Technol. **104**(5), 38–52 (2019)
11. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS One **13**(5), 1–23 (2018)
12. Roughan, M., Zhang, Y., Willinger, W., et al.: Spatio-temporal compressive sensing and internet traffic matrices. IEEE/ACM Trans. Netw. (ToN) **20**(3), 662–676 (2012)