Network Traffic Prediction for Intelligent Transportation Systems: A Reinforcement Learning Approach

Jian Song*, Hua Liu‡‡, Laisen Nie*‖, Zhaolong Ning‡‡, Mohammad S. Obaidat‡(Fellow of IEEE), and Balqies Sadoun**
*School of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710072, P.R.China.
‖School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China.
‡‡School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China.
‡Chair & Professor, Computer Science Department, and Director of Cybersecurity Center, University of Texas-Permian Basin, Odessa, TX 79762, USA, with King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan and University of Science and Technology Beijing, Beijing, China.
**College of Engineering, Al Balqa' Applied University, Al Salt, Jordan.
Email: songjian@mail.nwpu.edu.cn, nielaisen@nwpu.edu.cn, z.ning@ieee.com, msobaidat@gmail.com, m.s.obaidat@ieee.org, sadounbalqies@gmail.com

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10000764

***Abstract*—Vehicular Ad-Hoc Networks (VANETs), as the crucial support of Intelligent Transportation Systems (ITS), have received a great attention in recent years. Network traffic prediction is useful for network management and security in VANETs, such as network planning and anomaly detection. Due to the movement of nodes, the traffic flow in VANETs consists of a great number of irregular fluctuations, which is the main challenge for network traffic prediction. This paper proposes a novel algorithm, which combines Deep Q-Learning (DQN) and Generative Adversarial Networks (GAN) for network traffic prediction. We use DQN to carry out network traffic prediction, in which GAN is involved to represent Q-network. Meanwhile, the generative network can increase the number of samples to improve the prediction error. We evaluate the performance of our method by implementing it on two real network traffic data sets. Finally, we compare the two state-of-the-art competing methods with our method.**

***Index Terms*—network traffic prediction, deep reinforcement learning, generative adversarial networks**

I. Introduction

Vehicular Ad-Hoc Networks (VANETs) provide ubiquitous connections, which can be used to implement secure Intelligent Transportation Systems (ITS) [1]. The rapid development of VANETs makes network traffic show an explosive growth trend, and various network management and security problems have also increased. A large increase in network traffic will affect the link quality of the network, resulting in network congestion and even causing some network security issues [2]. To prevent network congestion and enhance network stability, the problem of network traffic prediction becomes critical. For a network, the traffic between end-to-end is defined as Origin-Destination (OD) flow [3]. Predicting OD flows is also helpful for network planning management and network link prediction [4].

The network traffic prediction problem can be solved by modeling traffic flows based on mathematical knowledge such as statistics and probability distribution. Network traffic is essentially a time series. Therefore, stationary and non-stationary time series models can be used to predict network traffic (e.g., the Autoregressive Moving Average model, Autoregressive Integrated Moving Average model). Although these models can describe the nonlinear characteristics of network traffic, there are problems of low precision in parameter prediction and model fitting. With the rapid development of artificial intelligence techniques, machine learning has gradually become a popular direction for prediction models. Many algorithms for predicting time series have been proposed by using deep learning, such as Long Short-Term Memory (LSTM), Temporal Convolutional networks (TCN), and Gate Recurrent Unit (GRU), which is a variant of LSTM. There are also some deep learning methods to extract the spatial characteristics of data sets into higher-level features, such as Graph Convolution Network (GCN), Convolutional Neural Networks (CNN), etc. These methods have better accuracy in network traffic prediction [5]–[11].

So far, researchers have proposed a great number of approaches for network traffic prediction. However, they are not suitable for predicting traffic flows in VANETs, due to the features of nodes in ITS [12]. The main challenges faced by the current network traffic prediction problem are:

- VANETs rely on the vehicle itself to provide information exchange, thus network traffic is heterogeneous. Due to the complex relationship between different services, the nonlinear problem of traffic prediction may be complicated, thereby reducing the accuracy during the process of network traffic prediction.

- For added nodes in VANETs, the historical information of network traffic generated by communication with other nodes is limited. How to use less historical traffic data to predict network traffic for a long time is the current challenge.
- Due to the constant movement of vehicles, the network traffic flow has the characteristics of suddenness and chaos, which will greatly affect the prediction effectiveness of the network prediction model. How to deal with traffic mutation in VANETs is a difficult problem in network traffic prediction.

Motivated by that, this paper proposes a novel method based on deep reinforcement learning combined with Deep Q-Learning (DQN) and Generative Adversarial Networks (GAN) for network traffic prediction. DQN is used for network traffic prediction in our method. The volume of network traffic in VANETs is huge, which causes the issue of constructing the Q-network. Meanwhile, network measurement is expensive for a vehicle, hence GAN is used to build an effective Q-network. The generative network can be used to produce the sampling for the training of the discriminative network, including network traffic samples and immediate reward. To further improve network traffic prediction accuracy, we train the discriminative network by using multi-task learning. With the adversarial training between the generative network and the discriminative network, the discriminative ability and prediction accuracy of the DQN network are continuously enhanced. The main contributions of this paper are:

- We propose a DQN-based method for network traffic prediction, in which various characteristics are extracted by the Q-network to overcome the problem of nonlinear traffic flows in VANETs. The problem of network traffic prediction is viewed as a Markov decision processing, in which the prior of network traffic and its predictor are regarded as the current and next states, respectively.
- A hybrid DQN combing with GAN is designed to handle the problem of limited samples. The generative network can produce the training samples with respect to network traffic and immediate reward, and the discriminative network is used to represent the Q-network.
- A multi-task learning scheme, in which immediate rewards are used as an auxiliary basis, is devised to improve the accuracy of network traffic prediction.

The rest of this paper is organized as follows. Related work is presented in Section II. Section III introduces our method for network traffic prediction. Section IV evaluates the proposed method by analyzing the experimental results. Finally, in Section V, we conclude the paper and propose future work.

II. Related Work

Network traffic is essentially time-series data, so its prediction problem can be transformed into a time-series modeling problem. Existing research mainly focuses on two categories of methods: traditional parametric methods and non-parametric deep learning methods. The former methods mainly model and predict network traffic based on mathematical statistics such as statistics and probability distributions, e.g., the multivariate multi-order Markov transformation method, the fast Fourier transformation method, and the Gaussian-based method. This type of method models network traffic with limited parameters and does not depend on the size of the data set. Non-parametric deep learning methods mainly capture the spatio-temporal correlation of OD flows with neural networks, e.g., RNN, LSTM, and TCN methods.

Deep learning, as a deep neural network, is a new research field in machine learning and has gradually become a common method for non-parametric prediction models. Owing to the dependence of network traffic on the spatio-temporal dimension, it is necessary to use the powerful learning ability of deep learning to accurately predict large-scale network traffic. Network traffic prediction can be combined with the reinforcement learning process. For instance, the authors model the network traffic in [5] with the Markov model and find the most possible network traffic at the next moment by Q-learning method based on Monte Carlo learning. The authors in [6] propose a novel GCN method that can extract the spatial characteristics and uses a gate-controlled recursive unit that can extract the temporal characteristics to predict network traffic. The pulsed neural network is a new network architecture, which works in a way closer to biological neurons. To solve the abnormal increase of network traffic caused by the surge of applications in the network, the authors in [7] construct a pulse neural network that can predict burst network traffic, in which the contraction factor is used to improve the accuracy of network traffic prediction. LSTM can be used to process sequence data, which solves the problem that traditional neural network models do not have the memory function and can effectively learn long-term dependencies between time series data. The authors in [8] utilize standard machine learning tools for network traffic prediction, including LSTM and Autoregressive Integrated Moving Average on top of per-user data. Meanwhile, the influence of different parameters is analyzed to improve network traffic prediction accuracy, such as the time granularity and the length of future predictions. Network traffic has an obvious spatio-temporal correlation. The research work in [9] proposes a network traffic prediction model for cellular networks, which uses LSTM to capture the temporal characteristics, and it also takes advantage of a multi-graph convolution network to capture the spatial characteristics. The paper in [11] proposes a new hybrid method for network traffic prediction, called Savitzky-Golay and TCN-based LSTM. The proposed method synergistically combines the power of the Savitzky-Golay filter, TCN, and LSTM. The proposed method first uses the Savitzky-Golay filter to remove noise in raw data. Furthermore, it applies TCN to extract short-term characteristics from the sequence and finally utilizes LSTM to capture long-term dependencies of traffic flows.

III. Our Methodology

Reinforcement learning is a machine learning algorithm based on Markov Decision Process, which can simply abstract the learning task into three factors, i.e., state, action, and reward [13]. The agent is the main body of learning, which can train and learn autonomously by continuous interaction with the environment. The state of the agent at time t is recorded as s_t, and the agent can take different actions according to different policies. After taking an action a_t, the agent changes the current state and then reaches a new state s_{t+1} at the next moment. The change of state can make the environment give the agent a feedback reward r_t. At time $t+1$, the agent will take a new action a_{t+1} according to state s_{t+1}, and it will get a feedback reward r_{t+1}. The purpose of reinforcement learning is to gain an action policy that can maximize the cumulative reward. The cumulative reward is R_t can be calculated by:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}. \quad (1)$$

In the above process, the rule of how to select an action according to the state s_t becomes the policy π. Since the reward is random in each game, more specifically, it is to maximize the expectation of cumulative reward.

Q-learning is a value-based reinforcement learning method that uses the update rule of time difference [13]. The expected reward for each state-action combination is defined as $Q(s,a)$. The main view of this method is to construct a two-dimensional Q table, where the Q value represents the expected value of state–action combination. We can select the action with optimal value in the current state according to the Q table, which maximizes the reward value. After the agent is trained and learned many times, each Q value eventually converges to the expected reward in the current game. The following is the iterative update formula of the Q table:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha\left(r + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\right), \quad (2)$$

where α defines the proportion of new value learning from old value, and the discount factor γ describes the weight of future rewards.

Q-learning can only deal with simple task environments, while real tasks are often more complex, especially for some tasks whose state space and action space are continuous. To solve this problem, the concept of deep reinforcement learning which combines the function fitting of deep learning to deal with continuous data and the decision-making ability based on the rewards of reinforcement learning, is proposed. The most representative algorithm in the field of value function approximation is DQN [13]. DQN uses a parameterized continuous function (i.e., Neural Network) $Q(s,a;\theta)$ to approximate the value function of state-action combination. The back propagation of the algorithm is obtained by minimizing the loss function $L(\theta)$, which is denoted by:

$$L(\theta) = \left(y - Q_{D1}(s,a;\theta)\right)^2, \quad (3)$$

where y represents the expected value of state-action combination, and $Q_{D1}(s,a;\theta)$ is the output value in the neural network. The target of the DQN loss function is to minimize the distance between y and $Q_{D1}(s,a;\theta)$.

GAN is an unsupervised learning model in deep learning, which uses the idea of game theory to establish a generative network that is used to generate fake data and a discriminative network to judge whether the input is true or not. [14]. With the adversarial training of the generative network and the discriminative network, GAN can generate artificial data that is closer to real data according to the distribution of real data. Leveraging the powerful feature learning and data generation capabilities of GAN can provide data support for our method.

The objective function of GAN is:

$$\min_G \max_D V(G,D) = E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[\log\left(D(\boldsymbol{x})\right)\right] + E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[\log\left(1 - D\left(G(\boldsymbol{z})\right)\right)\right], \quad (4)$$

where $V(G,D)$ denotes the objective function. E is the expectation of output after data is entered into the discriminative network. G and D are the generative network and discriminative network, respectively. Here, $p_{\boldsymbol{z}}(\boldsymbol{z})$ denotes a normal distribution, and $\boldsymbol{z}$ is the noise vector. Vector $\boldsymbol{z}$ input into the generative network, which generates fake data. Further, $p_{data}(\boldsymbol{x})$ is the distribution of real data, and $\boldsymbol{x}$ denotes the input vector. The two optimization models are combined to form the maximum and minimum objective functions, which include both the optimization of the discriminative model and the generative model to make it real. For the final output result, the parameters of both parties can be tuned at the same time. The training continues until the two have entered a state of balance and harmony.

In a network with N nodes and L links, if we do not consider the OD flow when the origin node and the destination node are the same, the network has $N(N-1)$ OD flows. At time t, the OD flow representation of network traffic can form a two-dimensional traffic matrix where each element represents the size of an OD flow at a given time. Here, $x_n(t)$ represents the traffic size of the nth OD flow at time t. The sequence of time-varying network traffic for the nth OD flow is known as $[x_n(1), \cdots, x_n(t-1), x_n(t)]$. Our objective is to build prediction models by extracting many characteristics of network traffic, so that we can acquire the predictor $\hat{x}_n(t+1)$.

From a probabilistic point of view, if

$$P\{X_n(t+1) = x_n(t+1) \,|X_n(1) = x_n(1), \cdots X_n(t) = x_n(t)\} = P\{X_n(t+1) = x_n(t+1) \,|X_n(t) = x_n(t)\}, \quad (5)$$

then the process of changing the current traffic is called the Markov process. At this time, the network traffic prediction model can be defined as $x_n(t) = f(x_n(t-1))$. Based on this Markov property, we associate network traffic prediction with the process of reinforcement learning. Network traffic $x_n(t)$ is defined as the state at time t in reinforcement learning, $x_n(t+1)$ is defined as the state at the next moment, and the jump process from network traffic $x_n(t)$ to $x_n(t+1)$ is defined as an action $a(x_n(t) \to x_n(t+1))$. Let the historical experience in the experience pool of reinforcement learning be

$$(x_n(t), a(x_n(t) \to x_n(t+1)), r(x_n(t) \to x_n(t+1)), x_n(t+1)), \tag{6}$$

where $r(x_n(t) \to x_n(t+1))$ represents the immediate reward. By evaluating the value between each state-action combination, the action with the largest value determines the state at the next moment, which also predicts the traffic at the next moment. In our method, the immediate reward is defined as $r(x_n(t) \to x_n(t+1)) = |a(x_n(t) \to x_n(t+1))|$, where $|a(x_n(t) \to x_n(t+1))|$ represents the frequency times from $x_n(t)$ transfers to $x_n(t+1)$. The larger $r(x_n(t) \to x_n(t+1))$ is, the greater the probability value of the next traffic state $x_n(t+1)$.

In our method, the environment represents the above-mentioned network traffic prediction model over time, and the state s shows the size of the OD flows. Since our traffic size has a large range of variation, we divide it by a large integer and normalize it. This has the advantages of narrowing the state space in the reinforcement learning process and speeding up the training process. The disadvantage of our method can only predict a range of traffic, which reduces the prediction accuracy. After inputting the size of network traffic at a certain time into the D1 network, the most probable size of the traffic at the next moment can be obtained. The transfer of network traffic will be collected by the experience pool. Moreover, after D1 learning, the approximate distribution of network traffic changes will be obtained. In our method, the vector $\boldsymbol{z}$ is sampled from a normal distribution and used as the noise of the current generative network. The experience pool is responsible for storing historical data $(x_n(t), a, r, x_n(t+1))$ in reinforcement learning tasks. The generative network is used to generate fake data, and the update of its network mainly depends on the discrimination of the D1 network. If a fake data item is judged as real data by the discriminative network, the current data item can be stored in the experience pool. The D1 network serves both as a discriminative network in the generative adversarial network and Q-network in DQN, combining the two tasks through multi-task learning [15]. The parameters of the D1 network and the generative network are updated by back-propagation. The loss function is composed of two parts, one is the distance between the reward value output by the neural network and the reward value given by the environment, and the other is the probability p of whether the data input is real data. The D2 network periodically replicates parameters from the D1 network to maintain the stability of the reward value.

In each round of network prediction, we initialize the first state $x_n(t)$ of the network traffic prediction uniformly. For action selection $a(x_n(t) \to x_n(t+1))$ in each state, the agent uses the ε-greedy algorithm to select actions based on the current state. In the process of interacting with the environment, you get new states s_{t+1} and rewards, and then we update the parameters in the D1 network and continue to loop to the end. The whole algorithm aims to continuously update the values in the neural network. Then, we select the best action $\underset{a(x_n(t)\to x_n(t+1))\in\mathcal{D}}{\arg\max} Q(a(x_n(t) \to x_n(t+1)))$ to take in a certain state according to the updated value. Different from the original DQN method, we use a GAN that generates $\mathcal{D}_n$ to deal with the problem of the insufficient state in the DQN learning process. DQN and GAN are directly combined through multi-task learning, and the loss function is $(y - Q_{D1}(x_n(t+1), a_n(t+1)))^2 + (1-p)^2$. The specific process is shown in Algorithm 1.

Algorithm 1: Network Traffic Prediction Based on GAN and DQN

input : $x_n(t), a(x_n(t) \to x_n(t+1))$
output: $r(x_n(t) \to x_n(t+1))$

Initialize the experience pool $\mathcal{D}$, and store $(x_n(t), a(x_n(t) \to x_n(t+1)), r(x_n(t) \to x_n(t+1)), x_n(t+1))$ in the experience pool;
Initialize the network parameters ϕ of the discriminant network Q_{D1};
Initialize the network parameters $\hat{\phi} = \phi$ of the discriminant network Q_{D2};
Initialize the generated network Q_G;
for $episode = 1, 2, ..., M$ **do**
  $x_n(t)$ = a state is randomly selected from the OD flow n;
  if $rand() <= \varepsilon$ **then**
    select action $a(x_n(t) \to x_n(t+1))$
  else
    $a_n(t) = \underset{a(x_n(t)\to x_n(t+1))\in D}{\arg\max} Q(a(x_n(t) \to x_n(t+1)))$
  $s_{t+1} \leftarrow x_n(t+1)$;
  Input state $(x_n(t), a_n(t))$ into Q_{D1} network, and then get output $Q_{D1}(x_n(t), a_n(t))$ and p;
  $y = r_n(t) + \gamma \max_{a_n(t+1)} Q_{D2}(x_n(t+1), a_n(t+1))$;
  for $k = 1, 2, ..., K$ **do**
    Taking $(y - Q_{D1}(x_n(t+1), a_n(t+1)))^2$ and $(1-p)^2$ as the loss function, $\mathcal{D}_n$ is the training sample, training the Q_{D1} network
  Train the generation network Q_G, and put the generated data into the experience pool $\mathcal{D}_n$;
  $params(Q_{D2}) = params(Q_{D1})$

IV. Experiments

A. Data Set

Abilene is a high-performance backbone network used in education and scientific research, which consists of 12 nodes, 30 internal links, and 24 external links. The data set of the Abilene backbone network is sampled at 5-minute intervals. The GÉANT network in Europe is a larger-scale backbone network consisting of 23 routers and 120 links, and the sampling interval is 15 minutes. In our method, since the range of network traffic size is too large, the traffic size should be preprocessed.

We first evaluate the prediction bias of traffic data, and the prediction bias is defined as [3]:

$$error_{bias}\left(n\right)=\frac{1}{T}\sum_{t=1}^{T}\left(x_n\left(t\right)-\hat{x}_n\left(t\right)\right) \tag{7}$$

where $x_n\left(t\right)$ and $\hat{x}_n\left(t\right)$ represent the real network traffic and its predictor, respectively. To evaluate the ability of the algorithm to capture the long correlation characteristics, the sampling standard deviation (SD) of the prediction bias is further involved based on analyzing the prediction bias in our simulation. The sampling SD is defined as [3]:

$$error_{sd}\left(n\right)=\sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(e\left(n\right)-error_{bias}\left(n\right)\right)^2} \tag{8}$$

where $e\left(n\right)=\hat{x}_n\left(t\right)-x_n\left(t\right)$. Furthermore, we evaluate the proposed method by involving the Spatial Relative Error (SRE) and the Temporal Relative Error (TRE), which are defined as [3]:

$$error_{SRE}\left(n\right)=\frac{\left\|\hat{x}_n\left(t\right)-x_n\left(t\right)\right\|_2}{\left\|x_n\left(t\right)\right\|_2} \tag{9}$$

$$error_{TRE}\left(t\right)=\frac{\left\|\hat{x}_n\left(t\right)-x_n\left(t\right)\right\|_2}{\left\|x_n\left(t\right)\right\|_2} \tag{10}$$

where $n=1,2,\ldots,N^2$,$t=1,2,\ldots,T$ and the notation $\left\|\cdot\right\|_2$ denotes the ℓ_2-norm.

B. Evaluation

In our simulation, we compare our method with two state-of-the-art competing methods, i.e., Sparsity Regularized Matrix Factorization (SRMF) and Principal Component Analysis (PCA). SRMF uses a novel compressed sensing framework to predict the traffic matrix. PCA can also be used to extract the main feature components of traffic data. The principle of PCA predicting the network traffic matrix is that the principal component will not change in a short time.

In Fig. 1, we use different colors and icons to represent three different prediction methods. The red circles represent our method, the blue asterisk represents PCA, and the green pentagram represents SRMF. In Abilene, as shown in Fig. 1(a), the x-axis is the sort of OD flows from maximum mean to minimum, and the y-axis is the prediction bias. Compared with SRMF and our method, the bias of PCA has a small fluctuation range, and there is no excessive overestimation and negative estimation. When the average OD flow is large, SRMF and our method both have negative or positive estimates, however, our method has a smaller fluctuation range in comparison. As shown in Fig. 1(b), the x-axis represents the SD of the traffic flow, and the y-axis is the prediction bias. SRMF has a large bias and small SD, thus it is more suitable for tracking the flow within two-time points, that is, small time scales. PCA and our method have the characteristics of small bias and large SD, and they are generally used to predict network traffic over long time intervals. The reason for the large SD of PCA is that it changes in the principal components.

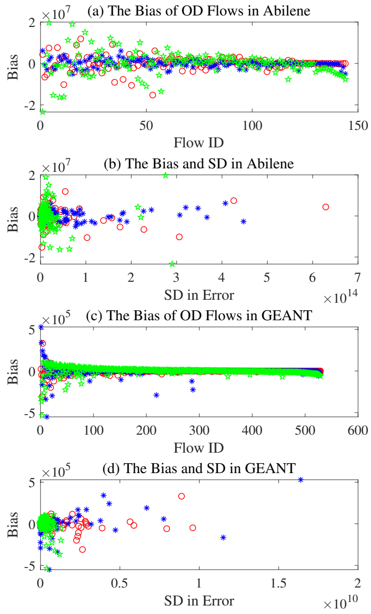

Fig. 1. The Bias of OD Flows and SD in Abilene and GÉANT.

In the GÉANT network, as shown in Figs. 1(c) and 1(d), the prediction bias of PCA, SRMF, and our method gradually decreases as the average OD flow decreases. Compared with SRMF, PCA and our method have a larger SD. The PCA method has obvious overestimation and underestimation phenomena. From our algorithm analysis, reinforcement learning is able to find out the size of the largest possible traffic at the next moment. Therefore, our method does well in discovering the overall shape of network traffic, and for urgent emergencies, the bias of some points of the traffic may be large.

Figure 2 shows the SRE and TRE of three methods in Abilene and GÉANT. Our method has lower TRE and SRE than PCA and SRMF. In GÉANT, the SRMF method has a large SRE in some OD flows. In Abilene, the averages of SREs of SRMF, our method, and PCA are 3.00, 0.67, and 0.74, respectively. The averages of TREs of SRMF, our method, and PCA are 0.31, 0.25, and 0.39, respectively. In GÉANT, the averages of TREs are 0.79, 0.48, and 0.88 in turn. To compare SRE and TRE of three algorithms more intuitively, the Cumulative Distribution Function (CDF) of SRE and TRE is given in the experimental simulation, as shown in Fig. 3. It can be seen intuitively from this figure that our method performs better at most times and on most OD flows. In Abilene, for 80% of all OD flows, the SREs of SRMF, our method, and PCA are less than 2.73, 0.77, and 1.01, respectively. For 80% of time slots, the TREs are less than 0.44, 0.28, and 0.38, respectively.

V. Conclusions

This paper studies the network traffic prediction in VANETs. We propose a network traffic prediction method based on DQN. In this paper, to increase the effectiveness

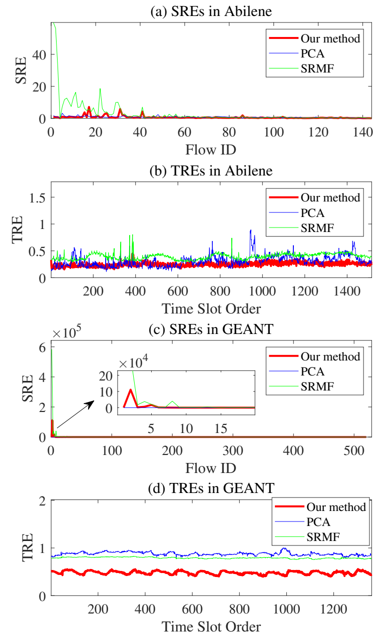

Fig. 2. The SRE and TRE in Abilene and GÉANT.

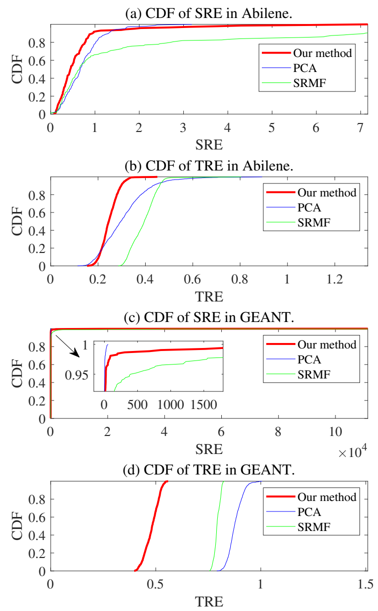

Fig. 3. The CDF of SRE and TRE in Abilene and GÉANT.

of DQN, GAN is used to build the Q network. Meanwhile, we leverage a multi-task learning mechanism to obtain excellent training in the discriminative network. We evaluate the proposed method by implementing it using two real network traffic data sets of Abilene and GÉANT. The simulation shows that the TRE and SRE of our method are lower than the other two highly competitive sophisticated methods. Moreover, it is suitable to capture the traffic flows over long time intervals. As future work, we will try to test our scheme on other network data sets and conduct more experiments.

VI. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62171378, and in part by the grant from the Macao Young Scholars Program (Project code: AM2020014).

References

[1] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. K. Kwok, “Joint Computing and Caching in 5G-Envisioned Internet of Vehicles: A Deep Reinforcement Learning-Based Traffic Control System,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5201–5212, 2021.

[2] Z. Ning, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, B. Hu, and Y. Li, “When Deep Reinforcement Learning Meets 5G-Enabled Vehicular Networks: A Distributed Offloading Framework for Traffic Big Data,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1352–1361, 2020.

[3] L. Nie, X. Wang, S. Wang, Z. Ning, M. S. Obaidat, B. Sadoun, and S. Li, “Network Traffic Prediction in Industrial Internet of Things Backbone Networks: A Multitask Learning Mechanism,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7123–7132, 2021.

[4] Y. Xie, Y. Guo, T. Zhou, Z. Mi, Y. Yang, B. Sadoun, and M. S. Obaidat, “A Strategy to Alleviate Rush Hour Traffics in Urban Areas Based on School-Trip Commute Information,” *IEEE Systems Journal*, vol. 15, no. 2, pp. 2630–2641, 2021.

[5] L. Nie, Z. Ning, M. S. Obaidat, B. Sadoun, H. Wang, S. Li, L. Guo, and G. Wang, “A Reinforcement Learning-Based Network Traffic Prediction Mechanism in Intelligent Internet of Things,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2169–2180, 2021.

[6] L. Yu, M. Li, W. Jin, Y. Guo, Q. Wang, F. Yan, and P. Li, “STEP: A Spatio-Temporal Fine-Granular User Traffic Prediction System for Cellular Networks,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 12, pp. 3453–3466, 2021.

[7] A. Yu, H. Yang, K. K. Nguyen, J. Zhang, and M. Cheriet, “Burst Traffic Scheduling for Hybrid E/O Switching DCN: An Error Feedback Spiking Neural Network Approach,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 882–893, 2021.

[8] A. Azari, F. Salehi, P. Papapetrou, and C. Cavdar, “Energy and Resource Efficiency by User Traffic Prediction and Classification in Cellular Networks,” *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2021.

[9] J. Lin, Y. Chen, H. Zheng, M. Ding, P. Cheng, and L. Hanzo, “A Data-driven Base Station Sleeping Strategy Based on Traffic Prediction,” *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.

[10] G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, “Characterization and Prediction of Mobile-App Traffic Using Markov Modeling,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 907–925, 2021.

[11] J. Bi, X. Zhang, H. Yuan, J. Zhang, and M. Zhou, “A Hybrid Prediction Method for Realistic Network Traffic With Temporal Convolutional Network and LSTM,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–11, 2021.

[12] A. H. Sodhro, M. S. Obaidat, Q. H. Abbasi, P. Pace, S. Pirbhulal, A.-U.-H. Yasar, G. Fortino, M. A. Imran, and M. Qaraqe, “Quality of Service Optimization in an IoT-Driven Intelligent Transportation System,” *IEEE Wireless Communications*, vol. 26, no. 6, pp. 10–17, 2019.

[13] A. S. Kumar, L. Zhao, and X. Fernando, “Multi-Agent Deep Reinforcement Learning-Empowered Channel Allocation in Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1726–1736, 2022.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[15] Y. Wang, L. Zhang, L. Wang, and Z. Wang, “Multitask Learning for Object Localization With Deep Reinforcement Learning,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, pp. 573–580, 2019.