Automatic Classification of Network Traffic Data based on Deep Learning in ONOS Platform

Jungmin Kwon*, Jungjin Lee*, Miseon Yu*† and Hyunggon Park*⋆‡
*Dept. Electronic and Electrical Engineering, Ewha Womans University, Seoul, Republic of Korea
⋆Smart Factory Multidisciplinary Program, Ewha Womans University, Seoul, Republic of Korea
†Korea Electronics Technology Institute (KETI), Seoul, Republic of Korea
‡The Alan Turing Institute, London, United Kingdom
jungmin.kwon@ewhain.net, jungjin.lee@ewhain.net, mmms543.keti@gmail.com, hyunggon.park@ewha.ac.kr

***Abstract*—Machine learning has been deployed in networks for automatically analyzing network data, proactively monitoring network dynamics, and predicting network resource availability. This becomes one of key technologies for efficient and autonomous network management in particular for software defined networks (SDN) environments. Especially, deep learning has brought recent breakthrough in machine learning algorithm as it can extract features based on artificial neural networks from data. In this paper, we study the deployment of deep neural network (DNN) for network traffic data classification, where DNN is deployed to automatically classify real network traffic data collected from ONOS (Open Network Operating System) platform. From the experiment results with simple network topologies, we conclude that DNN can be a potential approach to effective network packet classification. Moreover, it is confirmed that a deployment of DNN for a real network traffic data classification should consider not only the data packets that are intended to be delivered but also data packets required to maintain networks, as the classification performance of DNN significantly depends on the network traffic data.**

***Index Terms*—Machine learning, deep neural network, automatic network data classification, ONOS**

I. Introduction

Network automation is one of the key components for efficient management of 5G and Beyond 5G (B5G) mobile and wireless networks, where the network automation is known as the process of automating the network configuration, network management, network deployment, and network operation of physical and virtual resources and devices. The network automation can lead to several advantages such as efficient and robust network maintenance with lower operational cost. This becomes significantly important as a large number of devices are simultaneously connected and supported by a variety of services.

For network automation, machine learning has been widely deployed as it can provide an efficient approach to automatically analyzing network data, proactively monitoring network dynamics, and predicting network resource availability. This enables the networks to be efficiently and autonomously coped with in particular in SDN/NFV environment. The standardization group, ETSI ISG (industry specification group) ENI (experiential network intelligence), claims that network intelligent technology can be adopted into the infrastructure management, network operations, and service assurance [1]. The infrastructure management involves the prediction of traffic flows [2], network resource management [3]–[5], network auto-scaling [6], [7], and optimal tracking control [8]. The network operations include abnormal detection [9]–[11], quality of decision (QoD) improvement and network monitoring cost minimization [12], and VNF failure prediction [13]. The service assurance contains application identification [14], [15], VNF placement and chaining [16], [17] and content popularity prediction [18].

Recent breakthrough in machine learning algorithm is from the deep learning, which can extract features based on artificial neural networks from an abundance of data. Artificial neural networks assemble the input layer and an output layer with a hidden layer while deep learning derives outputs from multiple hidden layers. DBN (Deep Belief Network) [19], CNN (Convolutional Neural Network) [20], and LSTM (Long Short-Term Memory) [21] are different types of deep learning algorithms deployed to both supervised learning and unsupervised learning. While deep learning has shown significant improvement in machine learning techniques, it requires a massive amount of high-quality data for successful model training. For this, a large amount of data needs to be collected often in a central storage, incurring issues such as latency for data collection, network bandwidth cost for data exchange, as well as privacy. In particular, the privacy has been one of the main concerns of data owners in the deployment of machine learning algorithms.

In this paper, we focus on a key functionality for network automation, network data analytics, based on DNN with the actual data collected from an ONOS SDN controller. We show that DNN can be a potential candidate for effective network packet classification while a deployment of DNN for

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00024, Supervised Agile Machine Learning Techniques for Network Automation based on Network Data Analytics Function) and supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2020R1A2B5B01002528).

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on July 14,2022 at 15:53:27 UTC from IEEE Xplore. Restrictions apply.

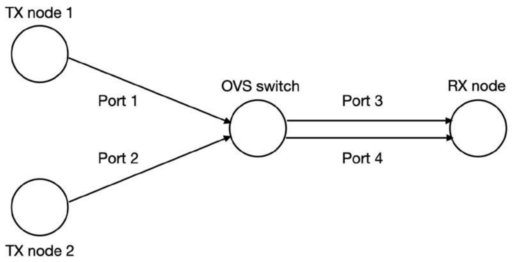

Fig. 1. Network topology (Scenario A)

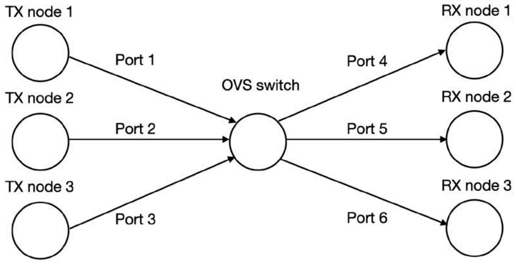

Fig. 2. Network topology (Scenario B)

TABLE I
CLASSIFICATION PERFORMANCE (SCENARIO A)

Scenario A	Accuracy (δ)
TX node 1 (Port 3)	99%
TX node 2 (Port 4)	98%

TABLE II
CLASSIFICATION PERFORMANCE (SCENARIO B)

Scenario B (with ARQ)	Accuracy (δ)
TX node 1 (Port 4)	83%
TX node 2 (Port 5)	95%
TX node 3 (Port 6)	97%
Scenario B (without ARQ)	Accuracy (δ)
TX node 1 (Port 4)	99%
TX node 2 (Port 5)	99%
TX node 3 (Po rt 6)	99%

a real network traffic data classification should consider data packets required to maintain networks, which influence the classification performance of DNN.

II. Network Traffic Data Classification in ONOS

In this paper, we consider a data classification algorithm based on DNN with a set of real network traffic data packets collected from an actual ONOS implementation. The goal is to train a DNN for packet classifier such that the sender (i.e., transmitter node) that transmits a data packet can be identified. The data used in the experiments is a set of labeled real network traffic data collected by REST API from ONOS controllers.

We consider two networks with simple topologies, where it consists of transmitter nodes, OVS switch, and receiver node. Each transmitter node generates data packets and they are delivered to the receiver node via intermediae OVS switch. The data collected at the receiver node is used to train and test DNN for network packet classification.

As a performance measure of network traffic data classification, we use the accuracy δ, defined as

$$\delta(\%) = \frac{TP + TN}{TP + FP + TN + FN} \times 100(\%) \tag{1}$$

where TP, FP, TN, and FN denote True Positive, False Positive, True Negative, and False Negative, respectively. A classification result can be determined as True if the port number is correctly identified.

III. Experiment Results

A. Experiment Setup

We consider two networks with simple topology, where it consists of transmitter nodes, OVS switch, and receiver node. The data packets are generated at the transmitter nodes in a predetermined duration and the traffic information is stored in the InfluxDB. The data stored in the InfluxDB is used for training DNN as well as testing the trained DNN at the application layer. We use a DNN for network packet classification in the experiments, where the DNN has 8 layers and 10 nodes in each layer.

In order to evaluate the proposed DNN for packet classification, we consider two different scenarios, where a network with two TX nodes (Scenario A) and a network with three TX nodes (Scenario B), as shown in Fig. 1 and Fig. 2.

In both scenarios, each TX node generates data with PING request messages in predetermined and fixed time intervals. Scenario A is designed for binary classification and its generalization to multiclass classification is considered in Scenario B. The network topologies considered in Scenario A and Scenario B are implemented in ONOS platform.

Note that the TX nodes are connected to the RX node via separate ports in our implementation, such that the performance of DNN classifier can be accurately evaluated.

B. Experiment Results

We collect data samples from the implemented network topologies in the ONOS platform, which are used for training and test. For Scenario A, the number of data samples collected from Port 1 and Port 2 are 3,320 and 775, respectively. For Scenario B, the numbers of data samples are 8,330, 3,494, and 2,803 collected from Port 4, Port 5 and Port 6, respectively. The number of data samples for network maintenance including ARQ (Automatic Repeat Request) message is 9,708. 70% of data samples from each scenario is used for model training and the rest of the data is used for test. We assume that there is no packet drop at every port, as the ONOS is implemented in wired networks.

The experiment results are shown in Table I and Table II, which classify TX nodes for network traffic data. In Scenario

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on July 14,2022 at 15:53:27 UTC from IEEE Xplore. Restrictions apply.

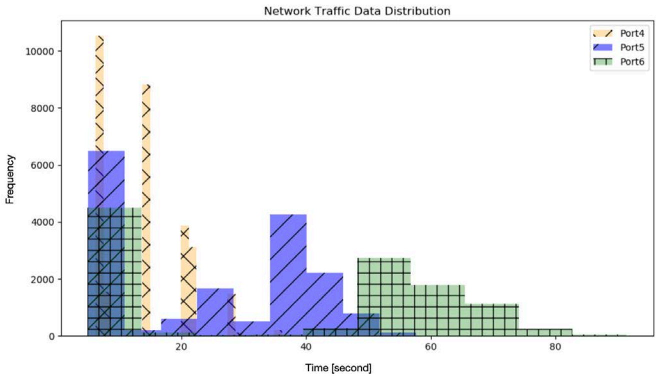

Fig. 3. Network data distribution for classification.

A, the classification performance shows 99% and 98% in Port 3 and Port 4, which corresponds to TX node 1 and TX node 2, respectively, meaning that the DNN based packet identification is effective.

It is observed on the other hand from Scenario B, which is an extension of Scenario A by including additional TX node 3, classification performance become significantly degraded, in particular Port 4 (see Scenario B (with ARQ) in Table II). This is because there exist not only the data generated by TX nodes but also the data required for network maintenance such as ARQ in a real network system. This is also confirmed from the network traffic data distribution shown in Fig. 3.

By removing the data required for network maintenance, i.e., ARQ packets, the classification performance can be significantly improved as shown in Scenario B (without ARQ) of Table II.

Therefore, it can be concluded from these experiments that DNN can be potentially efficient solution to the network packet classification. However, a practical deployment of DNN for a real network traffic data classification should consider the data packets that are intended to be delivered but also data packets required to maintain networks, as the classification performance of DNN significantly depends on the network traffic data.

IV. Conclusion

In this paper, we present preliminary results for automatic network data classification based on DNN. We confirm the potential effectiveness of the DNN to the network packet classification. Moreover, it is confirmed that a deployment of DNN for a real network traffic data classification should consider not only the data packets that are intended to be delivered but also data packets required to maintain networks, as the classification performance of DNN significantly depends on the network traffic data.

References

[1] H. Kim, M. Shin, B. Ahn, J. Lee, S. Lee, S. Lee, J. Ham, and S. Hyeon, “Network intelligence technologies,” *ETRI Insight*, 2018.

[2] W. Jiang, M. Strufe, and H. D. Schotten, “Experimental results for artificial intelligence-based self-organized 5G networks,” in *IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6.

[3] S. Sun, L. Gong, B. Rong, and K. Lu, “An intelligent SDN framework for 5G heterogeneous networks,” *IEEE Communications Magazine*, vol. 53, no. 11, pp. 142–147, 2015.

[4] A. Martin, J. Egaña, J. Flórez, J. Montalbán, I. G. Olaizola, M. Quartulli, R. Viola, and M. Zorrilla, “Network resource allocation system for QoE-aware delivery of media services in 5G networks,” *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 561–574, 2018.

[5] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, “Software-Defined networks with mobile edge computing and caching for Smart cities: A big data deep reinforcement learning approach,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 31–37, 2017.

[6] S. Rahman, T. Ahmed, M. Huynh, M. Tornatore, and B. Mukherjee, “Auto-scaling VNFs using machine learning to improve QoS and reduce cost,” in *IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[7] P. Tang, F. Li, W. Zhou, W. Hu, and L. Yang, “Efficient Auto-Scaling approach in the telco cloud using self-learning algorithm,” in *IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[8] J. Leguay, L. Maggi, M. Draief, S. Paris, and S. Chouvardas, “Admission control with online algorithms in SDN,” in *IEEE Network Operations and Management Symposium (NOMS)*, 2016, pp. 718–721.

[9] G. A. Ajaeiya, N. Adalian, I. H. Elhajj, A. Kayssi, and A. Chehab, “Flow-based intrusion detection system for SDN,” in *IEEE Symposium on Computers and Communications (ISCC)*, 2017, pp. 787–793.

[10] C. Li, Y. Wu, X. Yuan, Z. Sunand Weiming Wang, X. Li, and L. Gong, “Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN,” *Int. J. Communication Systems*, vol. 31, 2018.

[11] L. Fernández Maimó, A. L. Perales Gómez, F. J. García Clemente, M. Gil Pérez, and G. Martínez Pérez, “A self-adaptive deep learning-based system for anomaly detection in 5G networks,” *IEEE Access*, vol. 6, pp. 7700–7712, 2018.

[12] V. Sciancalepore, F. Z. Yousaf, and X. Costa Perez, “z-TORCH: An automated NFV orchestration and monitoring solution,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1292–1306, 2018.

[13] H. Huang and S. Guo, “Proactive failure recovery for NFV in distributed edge computing,” *IEEE Communications Magazine*, vol. 57, no. 5, pp. 131–137, 2019.

[14] Z. A. Qazi, J. Lee, T. Jin, G. Bellala, M. Arndt, and G. Noubir, “Application-awareness in SDN,” in *Proceedings of the Association for Computing Machineryz SIGCOMM*, vol. 43, no. 4, 2013, pp. 487– 488.

[15] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, “Machine learning in software Defined Networks: Data collection and traffic classification,” in *IEEE International Conference on Network Protocols (ICNP)*, 2016, pp. 1–5.

[16] J. Pei, P. Hong, and D. Li, “Virtual Network Function selection and chaining based on deep learning in SDN and NFV-enabled networks,” in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.

[17] X. Zhang, C. Wu, Z. Li, and F. C. M. Lau, “Proactive VNF provisioning with multi-timescale cloud resources: Fusing online learning and online optimization,” in *IEEE Conference on Computer Communications (INFOCOM)*, 2017, pp. 1–9.

[18] W. Liu, J. Zhang, Z. Liang, L. Peng, and J. Cai, “Content popularity prediction and caching for ICN: A deep learning approach with SDN,” *IEEE Access*, vol. 6, pp. 5075–5089, 2018.

[19] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief Nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[21] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on July 14,2022 at 15:53:27 UTC from IEEE Xplore. Restrictions apply.