International Journal of Computer Engineering in Research Trends
ISSN: 2349-7084/ https://doi.org/10.22362/ijcert.v10i7.897
Volume 10, Issue 7, July 2023

Research Paper

Routenet: Using Graph Neural Networks for SDN Network Modeling and Optimizations

[1*]P. Siva, [2]Cherukuri Sudhish, [3]Ogirala Divyanand, [4]K Sai Ananya Madhuri

[1]*Assistant Professor, Dept. of CSE, Matrusri Engineering College, Saidabad, Hyderabad, T.S.*
[2,3,4]*B.Tech Students, Dept. of CSE, Matrusri Engineering college, Saidabad, Hyderabad, T.S.*

**Corresponding Author:* sivap@matrusri.edu.in

Received: 11/05/2023, Revised: 17 /06/2023, Accepted: 21/07/2023 Published: 30/07/2023

Abstract: *This paper presents a simple and novel architecture for high-speed and low-cost processors based upon Software-Defined Networking (SDN), strictly neural networks, to solve combinatorial optimization problems within time. Software-Defined Networking (SDN) simplifies network management by separating the control plane from the data forwarding plane. However, the plane separation technology introduces many new loopholes in the SDN data plane. In order to facilitate taking proactive measures to reduce the damage degree of network security events, this paper proposes a security situation prediction method based on particle swarm optimization algorithm and long-short-term memory neural network for network security events on the SDN data plane. Network modeling is a key enabler to achieve efficient network operation in future self-driving Software-Defined Networks. However, we still lack functional network models able to produce accurate predictions of Key Performance Indicators (KPI) such as delay, jitter or loss at limited cost. In this report, we propose RouteNet, a novel network model based on Graph Neural Network (GNN) that is able to understand the complex relationship between topology, routing, and input traffic to produce accurate estimates of the per-source/destination per packet delay distribution and loss. RouteNet leverages the ability of GNNs to learn and model graph-structured information and as a result, our model is able to generalize over arbitrary topologies, routing schemes and traffic intensity. In our evaluation, we show that RouteNet is able to accurately predict the delay distribution (mean delay and jitter) and loss even in topologies, routing and traffic unseen in the training. Also, we present several use cases where we leverage the KPI predictions of our GNN model to achieve efficient routing optimization and network planning.*

Keywords- *Neural Networks, Software-Defined Networking (SDN), Key Performance Indicators (KPI), RouteNet, Graph Neural Network (GNN).*

---

1. Introduction

The positioning of the SDN controllers is the key factor in improving SDN network performance and aspects such as resiliency and re-configurability. In recent years there has been promising work on the controller placement problem in [17]. Most of this work considers physical controllers that will eventually be statically deployed in the network. Our work considers the possibility of creating a virtualized control plane to control the set of physical switches. A virtualized control plane brings with it the flexibility of reconfiguring and dynamically adapting the control plane to the needs of the network as well as the administration. Newer control plane architectures and protocols can also then be reprogrammed into the network. In this regard, this work primarily considers the deployment of a complete control plane architecture named herein as virtual control graphs.

Network modeling is a fundamental component to achieve efficient network optimization with special attention on future self-driving networks [8]. In the context of Software-Defined Networks, networking tasks are orchestrated from a centralized control plane, which may leverage a global picture of the network state in order to operate networks efficiently and dynamically adapt to changes in the network. To this end, network administrators typically define a target policy that may include some optimization objectives (e.g., minimize end-to-end latency) and constraints (e.g., security policy).

Then, SDN controllers are tasked to find some changes in the network configuration (e.g., routing) to accomplish the optimization objectives set by administrators. This is typically achieved by combining two main elements: a network model, and an optimization algorithm. In this well-known optimization architecture, the network model is tasked to predict the resulting performance (e.g, delay, packet loss) for specific configurations, and the optimization algorithm iteratively explores different configurations until it finds one that meets the optimization goals.

With the continuous expansion of the network scale in recent years, the lack of the traditional hierarchical network structure has gradually emerged. In 2009, Professor Mckeown of Stanford University proposed the concept of software-defined networking (SDN). The core feature of SDN is to separate the control logic and forwarding behavior in network forwarding as different levels, and the application plane is composed of various network services, and the forwarding is managed through the control plane. A unified and open data interface (such as OpenFlow [20]) is used to communicate between the control plane and the data plane. The control plane sends forwarding rules to the data plane switch through this interface. The switch only needs to perform forwarding according to the rules. Obviously, SDN technology effectively reduces the load of forwarding equipment, and centralized control also provides convenience for network operation and management, and also greatly improves the flexibility of the network.

One fundamental issue of network optimization solutions is that they can only optimize based on the performance metrics provided by the network model. Thus, in order to optimize Key Performance Indicators (KPI) such as delay or packet loss in networks, it is essential a network model able to understand how these performance indicators are related to the network state metrics collected from the data plane, which often can provide only timely statistics of traffic volume (e.g., traffic matrix) in real-world deployments. In this context, much effort has been devoted in the past to build network models able to predict performance metrics, however nowadays we still lack functional models providing accurate predictions of relevant KPI like delay, jitter or packet loss. Analytic models, mainly based on Queuing Theory [18], assume some non-realistic properties of networks (e.g., traffic with Poisson distribution, probabilistic routing) and, as a result, they are not accurate to produce KPI predictions in large-scale networks with realistic configurations such as multi-hop routing [7]. Conversely, packet-level network simulators showed to be very accurate for this purpose, but their high computational cost makes it unfeasible to leverage them to operate networks in short time scales.

Due to the layered idea of centralized control, the SDN structure itself also brings new security issues. Combined with the idea of SDN's own centralized management, security situation awareness as a globally coordinated and centralized management security monitoring method is considered to be an effective means of managing SDN security. The process of situational awareness is usually divided into situational element acquisition, situational assessment, and situational prediction. In the SDN environment, there is no mature method for assessing and predicting the security situation that affects the SDN architecture. This paper proposes a security situation prediction method based on particle swarm optimization (PSO) algorithm and long short-term memory (LSTM) neural network. This method uses the network security event information to evaluate and predict the security situation of the SDN data plane, so as to provide more targeted and valuable security intelligence for network managers and security analysts. To design a novel network model based on Graph Neural Networks that is able to understand the complex relationship between topology, routing, and input traffic to accurately estimate the distribution of the per-packet delay and loss ratio on every source-destination pair.

The rest of the paper is organized as follows: Section II introduces the Literature survey, Section III Using Graph Neural Networks for SDN Network Modeling and Optimizations explains. Section IV includes the experimental result analysis and finally paper concludes with Section V.

2. Literature Survey

D. Zhao and J. Liu et al., [6] Study on network security situation awareness based on particle swarm optimization algorithm. In the network security situation assessment, parallel reduction algorithms based on attribute importance matrix is proposed to reduce the attributes of the data source data. A network security situation assessment model based on a gravity search algorithm is proposed to optimize support vector machines to reduce the error between the evaluation value and the actual network security situation value. A random forest-based network security situation assessment model is proposed to make the assessment more objective and accurate. None of the above methods can effectively assess the situation based on the structural characteristics of the SDN environment.

Z. Zhan, M. Xu, and S. Xu et al., [11] Predicting Cyber Attack Rates WithExtreme Values. In the network security situation prediction, a network attack prediction model combining extreme value theory and time series theory is proposed, which is effective for both long term and short term prediction. An automated network attack prediction system that uses various public and personal data sources and uses capture technology is proposed to predict future network security events. Data from security service providers are used to analyze the correlation between security event contexts and predict security events accordingly. The above situation prediction methods are only applicable to the traditional network environment, and it is difficult to smoothly migrate to the SDN environment.

W. Zhao, H. Yang, J. Li, L. Shang, L. Hu, and Q. Fu et al., [3] Network Traffic Prediction in Network Security Based on long short-term memory (LSTM). Artificial neural networks can be widely used in the research of nonlinear systems and can predict the changing laws of network traffic. The neural network has a complex structure, so the training process will also have defects. During the iterative process, the optimization may be slower or fall into local extremes, resulting in lower prediction accuracy. PSO algorithm applied to the optimization of the LSTM neural network can avoid the above problems. The combined prediction method combines more than two methods with their respective advantages to form a new prediction model, which can show better prediction performance.

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini et al.,[19] In this paper we present RouteNet, a novel network model based on Graph Neural Networks (GNN). Our model is able to understand the complex relationship between topology, routing, and input traffic to accurately estimate the distribution of the per-packet delay and loss ratio on every source-destination pair. GNNs are tailored to achieve relational reasoning and combinatorial generalization over information structured as graphs and as a result our model is able to generalize over arbitrary topologies, routing schemes and variable traffic intensity.

F. Geyer, et al., [1] describes a Performance evaluation of network topologies using graph-based deep learning. In particular, RouteNet captures meaningfully traffic routing over network topologies. This is achieved by modeling the relationships of the links in topologies with the source-destination paths resulting from the routing schemes and the traffic flowing through them. One main contribution of the RouteNet architecture compared to other GNN based models is the representation of paths as ordered sequences of links. This makes RouteNet a new GNN architecture designed especially for computer network control and management.

K. Rusek, J. Su´arez-Varela, A. Mestres, P. Barlet-Ros, and A. Cabellos- Aparicio et al.,[2] presents the potential of graph neural networks for network modeling and optimization in SDN. An earlier version of this paper was presented. In that version, two different models were used to predict the per-path mean delay and jitter. In this paper we present an extended RouteNet model inspired by Generalized Linear Models that directly estimates the per-packet distribution of the delay on each path. This enables to use a single model to predict any metric associated to end-to-end per-packet delay (e.g., mean delay, jitter). Additionally, in this paper we adapted RouteNet to make also predictions of the per-source/destination packet loss ratio.

A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, and E. Alarcon et al., [9] Knowledge-defined networking (KDN) is an evolutionary step toward autonomous and self-driving networks. The building blocks of the KDN paradigm in achieving self-driving networks are software-defined networking (SDN), packet-level network telemetry, and machine learning (ML). The KDN paradigm intends to integrate intelligence to manage and control networks automatically. In this study, we first introduce the disadvantages of current network technologies. Then, the KDN and associated technologies are explored with three possible KDN architectures for heterogeneous wireless networks. Furthermore, a thorough investigation of recent survey studies on different wireless network applications was conducted. The aim is to identify and review suitable ML-based studies for KDN-based wireless cellular networks. These applications are categorized as resource management, network management, mobility management, and localization. Resource management applications can be further classified as spectrum allocation, power management, quality-of-service (QoS), base station (BS) switching, cache, and backhaul management.

Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang et al., [5] Modern communication networks have become very complicated and highly dynamic, which makes them hard to model, predict and control. In this paper, we develop a novel experience-driven approach that can learn to control a communication network from its own experience rather than an accurate mathematical model, just as a human learns a new skill (such as driving, swimming, etc). Specifically, we, for the first time, propose to leverage emerging Deep Reinforcement Learning (DRL) for enabling model-free control in communication networks; and present a novel and highly effective DRL-based control framework, DRL-TE, for a fundamental networking problem: Traffic Engineering (TE). The proposed framework maximizes a widely-used utility function by jointly learning network environment and its dynamics, and making decisions under the guidance of powerful Deep Neural Networks (DNNs).

B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani et al., [10] Recent years, Software Defined Routers (SDRs) (programmable routers) have emerged as a viable solution to provide a cost-effective packet processing platform with easy extensibility and programmability. Multi-core platforms significantly promote SDRs parallel computing capacities, enabling them to adopt artificial intelligent techniques, i.e., deep learning, to manage routing paths. In this paper, we explore new opportunities in packet processing with deep learning to inexpensively shift the computing needs from rule-based route computation to deep

learning-based route estimation for high-throughput packet processing. Even though deep learning techniques have been extensively exploited in various computing areas, researchers have, to date, not been able to effectively utilize deep learning-based route computation for high-speed core networks.

K. Rusek and P. Chołda et al., [4] Network modeling is a key enabler to achieve efficient network operation in future self-driving Software-Defined Networks. However, we still lack functional network models able to produce accurate predictions of Key Performance Indicators (KPI) such as delay, jitter or loss at limited cost. In this paper we propose RouteNet, a novel network model based on Graph Neural Network (GNN) that is able to understand the complex relationship between topology, routing, and input traffic to produce accurate estimates of the per-source/destination per packet delay distribution and loss.

3. Routenet: Using Graph Neural Networks for SDN Network Modeling and Optimizations

In this work, Using Graph Neural Networks for SDN Network Modelling and Optimization is presented. The Fig. 1 shows the architecture of presented model.

The input data to RouteNet consists of the network topology, traffic demand, and various network parameters such as link capacity, delay, and bandwidth. Network modeling enables the control plane to further exploit the potential of SDN to perform fine-grained management. This permits the evaluation of the resulting performance of what-if scenarios without the necessity to modify the state of the data plane. It may be profitable for a number of network control and management applications such as optimization, planning or fast failure recovery.

Technically speaking, SDN is made possible by separating the control plane from the data plane. "Plane" is a networking term that refers to an abstract conception of where networking processes take place. The control plane refers to networking processes that direct network traffic, while the data plane is the actual data traversing the network. The control plane does this by establishing network routes and communicating which protocols should be used.

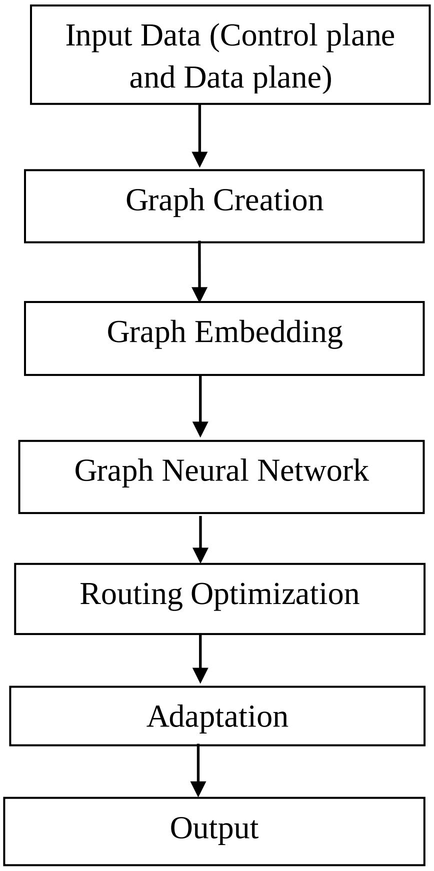

Fig. 1 The architecture of presented model.

The input data is used to create a graph representation of the network topology, where nodes represent switches or routers and edges represent links between them. Can we use Data Flow Diagram (DFD) to implement an SDN Network Management System. The developed system handles topology modeling and storage (as a Graph in a graph DB), load balancing, security, acquiring traffic statistics, and routing. All these functions are performed as a cooperation between the graph database and the SDN controller. The results are projected as services in the SDN Application Plane. Data Flow Diagrams can represent any kind of processing, since it show how data moves between systems or processes or storage. DFD allow also to decompose systems/processes into sub-systems/sub-processes to see the details of a system's internal flows, up to the desired level of details.

RouteNet uses a graph embedding algorithm to learn a low-dimensional representation of the graph, which captures the underlying structure and topology of the network. Graph embedding can lead to better quantitative understanding and control of complex networks, but traditional methods suffer from high computational cost and excessive memory requirements associated with the high-dimensionality and heterogeneous characteristics of industrial size networks. Graph embedding techniques can be effective in converting high-dimensional sparse graphs into low-dimensional, dense and continuous vector spaces, preserving maximally the graph structure properties. Another type of emerging graph embedding employs Graph Neural Network (GNN) graph embedding with important uncertainty estimation. The main goal of graph embedding methods is to pack every node's properties into a vector with a smaller dimension, hence, node similarity in the original complex irregular spaces can be easily quantified in the embedded vector spaces using standard metrics.

The embedded graph is fed into a neural network, typically a graph convolutional neural network (GCN), which is trained to optimize the routing paths for traffic based on various criteria, such as minimizing latency or maximizing throughput. Graph neural network is an artificial neural architecture designed for graph-structured data, where the nodes, edges, and the whole graph can have associated feature vectors. The most important property of GNN is that it preserves the basic topological relations between node adjacencies (graph isomorphism), so it is well suited to be used for different topologies without retraining.

Once the neural network is trained, it can be used to generate optimized routing paths for traffic in the network. In particular, RouteNet captures meaningfully traffic routing over network topologies. This is achieved by modeling the relationships of the links in topologies with the source-destination paths resulting from the routing schemes and the traffic flowing through them. One main contribution of the RouteNet architecture compared to other GNN based models is the representation of paths as ordered sequences of links. This makes RouteNet a new GNN architecture designed especially for computer network control and management. Main contributions of RouteNet are: More diverse and larger network topologies, Probabilistic modeling inspired by Generalized Linear Models, Adaptation to predictions of the per-source/destination packet loss ratio, Computation cost improvement, Additional input features (support for arbitrary link capacities), New network optimization use cases incorporating packet loss requirements, Graphs that show the accuracy of the predictions compared to the actual data.

RouteNet, the GNN based model proposed in this paper, is able to propagate any routing scheme throughout a network topology and abstract meaningful information of the current network state to produce relevant performance estimates. More in detail, RouteNet takes as input a given topology, a source destination routing scheme (i.e., list of end-to-end paths) and a traffic matrix (defined as the bandwidth between each node pair in the network), and produces as output performance metrics according to the current network state (per-path mean delay, jitter, and packet loss).

RouteNet can be continually updated and adapted to changing network conditions through feedback mechanisms, such as monitoring network performance and updating the neural network's parameters accordingly. Graph neural networks (GNNs) is widely used to learn a powerful representation of graph-structured data. Recent work demonstrates that transferring knowledge from self-supervised tasks to downstream tasks could further improve graph representation. However, there is an inherent gap between self-supervised tasks and downstream tasks in terms of optimization objective and training data. Conventional pre-training methods may be not effective enough on knowledge transfer since they do not make any adaptation for downstream tasks. To solve such problems, we propose a new transfer learning paradigm on GNNs which could effectively leverage self-supervised tasks as auxiliary tasks to help the target task. Our methods would adaptively select and combine different auxiliary tasks with the target task in the fine-tuning stage. We design an adaptive auxiliary loss weighting model to learn the weights of auxiliary tasks by quantifying the consistency between auxiliary tasks and the target task. In addition, we learn the weighting model through meta-learning. Our methods can be applied to various transfer learning approaches, it performs well not only in multi-task learning but also in pre-training and fine-tuning. Comprehensive experiments on multiple downstream tasks demonstrate that the proposed methods can effectively combine auxiliary tasks with the target task and significantly improve the performance compared to state-of-the-art methods.

The output of RouteNet is a set of optimized routing paths that can be used to route traffic in the network, with the goal of improving network performance and reducing congestion. GNNs are tailored to achieve relational reasoning and combinatorial generalization over information structured as graphs and as a result our model is able to generalize over arbitrary topologies, routing schemes and variable traffic intensity. In particular, RouteNet captures meaningfully traffic routing over network topologies.

4. Result Analysis

In this analysis, Using Graph Neural Networks For SDN Network Modelling And Optimization is presented. Statistics like GNN provide a good picture of the general accuracy of the model. However, there are more elaborated methods that offer a more detailed description of the model behavior. However, in our simulation datasets there are many cases with zero drops, and these cases are also accurately predicted by the RouteNet model. To test this one can compute the correlation coefficient between predictions and true loss ratio (including cases without loss where the value is 0). The performance analysis of proposed method evaluated with parameters, Accuracy, Delay, speed and Loss.

Accuracy: The Accuracy is used to evaluate the effectiveness of the classifier as shown in below equation (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \dots (1)$$

Delay: Network Delay refers to the round-trip measure of time it takes for data to reach its destination across a network. Delay is strongly linked to network connection speed and network bandwidth. Delay is measured in milliseconds (ms). A lower number of milliseconds means that the delay is low, the network is performing more efficiently and therefore, the user experience is better. Delay is strongly linked to network connection speed and network bandwidth.

Loss: Loss or Network Packet Loss refers to the number of data packets that were successfully sent out from one point in a network, but were dropped during data transmission and never reached their destination. When large amounts of Packet Loss start plaguing the network, it’s a clear indicator that the network isn’t performing as it should be. Incomplete or delayed data transmission can impact network and application performance and affect the user experience. Loss is measured in percentage (%).

Loss = Number of lost packets / Number of received packets

Table 1. Performance of the proposed model

Parameters	LSTM	GNN
Accuracy	80.45	97.45
Loss(%)	200	30
Delay (ns)	8.74	3.32
Speed	150	230

The Fig. 2 shows the Accuracy comparison between standard LSTM based approach and GNN approach.

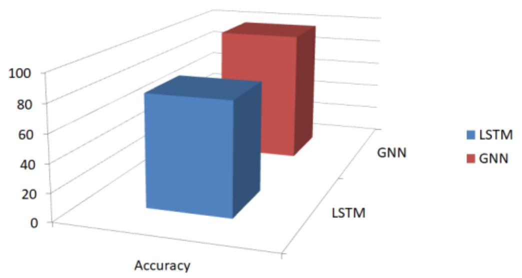

Fig.2 Accuracy Comparison Graph

The Fig. 3 shows the Loss comparison between standard LSTM based approach and GNN approach.

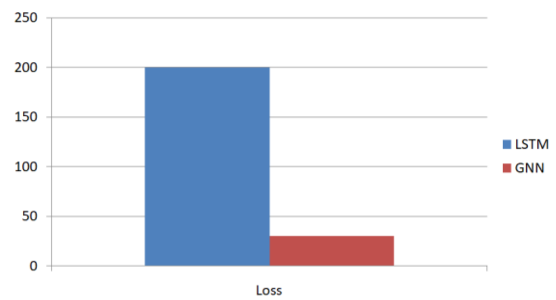

Fig.3 Loss Comparison Graph

The Fig. 4 shows the Delay comparison between standard LSTM based approach and GNN approach.

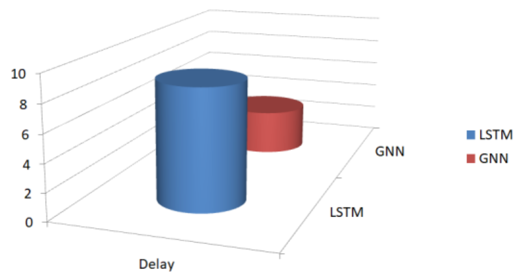

Fig.4 Delay Comparison Graph

The Fig. 5 shows the Speed comparison between standard LSTM based approach and GNN approach.

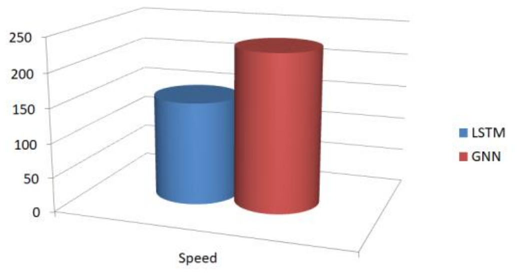

Fig.5 Speed Comparison Graph

5. Conclusion

In this paper, Using Graph Neural Networks for SDN Network Modelling and Optimization is presented. Software-Defined Networks offer an unprecedented degree of flexibility in network control and management that, combined with timely network measurements collected from the data plane, open the possibility to achieve efficient online network optimization. As a result, current optimization approaches are limited to improving a global performance metric, such as network utilization or planning the network based on the worst-case estimates of the latencies obtained from network calculus. We designed and implemented an extended RouteNet model based on Generalized Linear Models that predicts the distribution of the per-source/destination per-packet delay and loss in networks. From these output distributions we evaluate the accuracy of the mean per-packet delay, the speed, and the mean packet loss predicted. Our evaluation results show that RouteNet is able to generalize to other network topologies, routing configurations and traffic matrices not seen in the training. Also, the modular architecture of RouteNet simplifies transfer learning, which consists of reusing neural network models trained for a particular task and retraining them to address other problems in similar domains. Compared to other based models, the presented GNN model has better performance in terms of delay, speed, loss, accuracy.

References

[1] F. Geyer, “DeepComNet: Performance evaluation of network topologies using graph-based deep learning,” Performance Evaluation, vol. 130, pp. 1–16, apr 2019.

[2] K. Rusek, J. Su´arez-Varela, A. Mestres, P. Barlet-Ros, and A. Cabellos- Aparicio, “Unveiling the potential of graph neural networks for network modeling and optimization in SDN,” in Proceedings of the ACM Symposium on SDN Research (SOSR), 2019, pp. 140–151.

[3]W. Zhao, H. Yang, J. Li, L. Shang, L. Hu, and Q. Fu, "Network TrafficPrediction in Network Security Based on EMD and LSTM," inProceedings of the 9th International Conference on ComputerEngineering and Networks(CENet), 2019, pp. 461-469.

[4] K. Rusek and P. Chołda, “Message-passing neural networks learn little’s law,”IEEE Communications Letters, vol. 23, no. 2, pp. 274–277, 2019.

[5] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, “Experience-driven networking: A deep reinforcement learning based approach,” arXiv preprint arXiv:1801.05757, 2018.

[6]D. Zhao and J. Liu, "Study on network security situation awareness based on particle swarm optimization algorithm," Computers & Industrial Engineering, vol. 125, pp. 764-775, 2018.

[7] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, “Experience-driven networking: A deep reinforcement learning based approach,” arXiv preprint arXiv:1801.05757, 2018.

[8] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, and E. Alarc´on, , “Knowledge-defined networking,” ACM SIGCOMM Comput. Commun. Rev., vol. 47, no. 3, pp. 2–10, Sep. 2017.

[9] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, and E. Alarcon, ´ “Knowledge-defined networking,” ACM SIGCOMM Comput. Commun. Rev., vol. 47, no. 3, pp. 2–10, Sep. 2017.

[10] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, “Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning,” IEEE Transactions on Computers, vol. 66, no. 11, pp. 1946–1960, 2017.

[11]Z. Zhan, M. Xu, and S. Xu, "Predicting Cyber Attack Rates WithExtreme Values," IEEE Transactions on Information Forensics andSecurity, vol. 10, no. 8, pp. 1666-1677, 08/01 2015.

[12] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, p. 436, 2015.

[13] A. Sallahi and M. St-Hilaire, “Optimal model for the controller placement problem in software defined networks,” Communications Letters, IEEE, vol. 19, no. 1, pp. 30–33, 2015.

[14] L. F. Muller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspary, and M. P. Barcellos, “Survivor: an enhanced controller placement strategy for improving sdn survivability,” in Global Communications Conference (GLOBECOM), 2014 IEEE. IEEE, 2014, pp. 1909–1915.

[15] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran- Gia, “Pareto-optimal resilient controller placement in sdn-based core networks,” in Teletraffic Congress (ITC), 2013 25th International. IEEE, 2013, pp. 1–9.

[16] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, and R. Boutaba, “Dynamic controller provisioning in software defined networks,” in Network and Service Management (CNSM), 2013 9th International Conference on. IEEE, 2013, pp. 18–25.

[17] B. Heller, R. Sherwood, and N. McKeown, “The controller placement problem,” in Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012, pp. 7–12.

[18] F. Ciucu and J. Schmitt, “Perspectives on network calculus: no free lunch, but still good value,” ACM SIGCOMM Comput. Commun. Rev., vol. 42, no. 4, pp. 311–322, 2012.

[19] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, 2009.

[20]N. McKeown et al., "OpenFlow: enabling innovation in campusnetworks," ACM SIGCOMM Computer Communication Review, vol.38, no. 2, pp. 69-74, 2008.