On Failure Classification Based on Supervised Graph Classification with GCNs in IP Core Networks by NFV-Based Test Environments

Takanori Hara
Graduate School of Science and Technology
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
hara.takanori.hm8@is.naist.jp

Kentaro Fujita
Graduate School of Science and Technology
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
fujita.kentaro.fk0@is.naist.jp

***Abstract*—With the proliferation of 5G mobile networks, mobile operators have to provide stable and high-quality internet services. Once an unexpected defect in a domain on IP core networks happens, the influence of the defect will be rapidly spread all over the world because of the mutual operations among operators. Since only highly experienced operators tackle these network failures, anomaly detection leveraging artificial intelligence and machine learning is required to operate automatically and rapidly as well as to reduce their operating expenditures. In this paper, we propose the supervised graph classification with graph convolutional networks (GCNs) by regarding the failure classification as the graph classification. As an initial step toward the realization of the failure classification with the GCNs in the IP core network, we investigate the potential of the supervised graph classification with the GCNs for detecting and classifying the network status such as route information failures, single point failures, and packet loss and/or delay. Fundamental results show that (1) the GCN-based approach exhibits the higher accuracy compared with the other schemes and (2) the explicit topology embedding contributes to the performance improvement for estimating the packet loss/delay.**

***Index Terms*—Failure classification; anomaly detection; BGP injection/hijacking; supervised graph classification; Graph convolutional networks (GCNs)**

I. Introduction

With the proliferation of 5G mobile networks, many services will infiltrate various aspects of our daily life and most people will be benefited from 5G mobile networks [1], [2]. Mobile operators have to provide the stable and high-quality internet services. Once an unexpected defect in a domain on IP core networks happens, the influence of the defect will be rapidly spread all over the world because of the mutual operations among operators. Only highly experienced operators can tackle such network failures at the expense of their resources. To reduce operating expenditures, anomaly detection is required to operate automatically and rapidly by leveraging artificial intelligence and machine learning.

Each mobile operator has at least one IP core network connected to mobile core networks. The IP core network interconnects with other IP core networks of other operators through border gateway routers The border gateway routers continue to update their route information received from internal and external route information [3], [4]. Since these routers play an important role to provide high-quality internet services, it is required to detect the defects and mis-operations immediately in the IP core network.

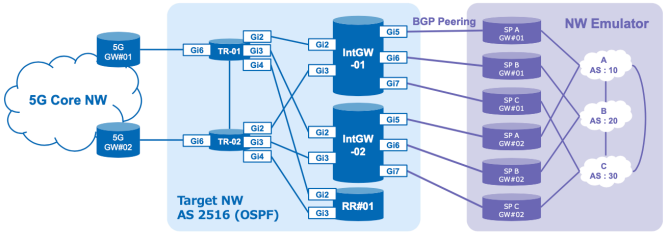

Fig. 1: Target network.

In this paper, we address the anomaly detection and failure classification in the IP core network by leveraging graph neural network (GNN) based approaches [5], [6]. We propose the supervised graph classification with graph convolutional networks (GCNs) by regarding the failure classification as the graph classification. The main motivation and contribution of this paper are as follows:

1) As an initial step toward the realization of the failure classification in IP core networks with the graph structure and its features, we investigate the potential of the supervised graph classification with the graph convolutional networks (GCNs) for detecting and classifying the network status such as route information failures, single point failures, and packet loss and/or delay.
2) We propose a simple graph transformation applying a target network topology for the GCN-based training model.
3) Fundamental results show that the GCN-based model improves the accuracy in comparison to other machine learning based models. Specifically, we confirm that the explicit topology embedding contributes to the performance improvement for estimating the packet loss/delay.

The rest of this paper is organized as follows. Section II gives the related work. After introducing the problem setting in Seciton III, we develop the proposed approach in Section IV. In Section V, we demonstrate the performance comparison among machine learning based approaches. Finally, Seciton VI gives the conclusions and future work.

II. Related Work

There are several studies for network fault analysis using machine learning. Kawasaki et al. conducted a comparative analysis of network fault classification using MLP, RF, and SVM and developed the dataset generator for NFV test environments [7]. Sauvanaud et al. proposed a random forest algorithm to detect service level agreements (SLAs) violations in NFV networks [8]. Qader et al. conducted a comparative analysis of network traffic fault using the clustering method [9]. From the viewpoint of the route information failures, BGP anomaly detection has been studied over several decades [3]. Cho et al. proposed a random forest classifier for BGP and the trained model classifier finds four categories of BGP, i.e., hijacking typos, prepending mistakes, origin changes, and forged AS paths [4].

There are several studies on applying graph neural networks (GNNs) for networking [10]–[12]. The GNN is a neural network based machine learning which enables explicit topology embedding in training model [5], [6]. Graph convolutional network (GCN) is a variant of the convolutional neural networks (CNNs) that can convolute the data on non-Euclidean space, i.e., graph structures, [13]. Geyer et al. estimated the throughput of TCP flows from the network topology and its node features by using the GNN-like approach [10]. Zheng et al. proposed the GCN-based network traffic classification by combining the traffic trace graph with the statistical features into GCN-based model [14]. Suzuki et al. proposed the semi-supervised learning for estimating communication delay between node pairs in the large-scale networks [11] Nakashima et al. proposed the deep reinforcement learning with the GCNs for channel allocation of wireless LANs [12]. In this paper, as an initial step toward the realization of failure classification in IP core networks, we investigate the potential of the supervised graph classification with the GCNs for detecting and classifying the network status such as the route information failure, single point failure, and packet loss and delay.

III. Problem Setting

A. Problem statement

The dataset at border gateway routers provided for this problem includes network status such as normal and a failure, mis-operation, and normal or abnormal labels. We have to create a model to detect and classify network status of a failure utilizing the dataset and evaluate the performance using the proposed model.

TABLE I: Failure use cases.

Scenario	Use case	Description
Route information failure	BGP injection	Inject the anomaly route from another SP
	BGP hijacking	Hijack the own origin route by another SP
Interface failure	Interface down	Cause an interface down
	Packet loss/delay	Cause the packet loss/delay on an interface
NE failure	NE reboot	Unplanned reboot of a NE

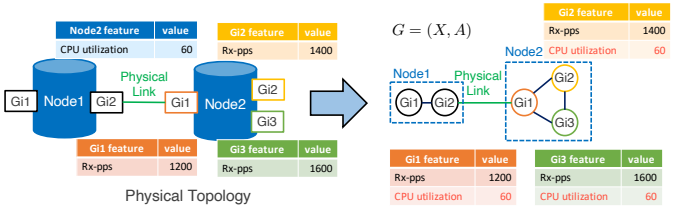

Fig. 2: Graph transformation.

B. Network topology

Fig. 1 illustrates the physical topology of the target network.[1] The IP core network consists of five network elements (NEs) where TR-01 and TR-02 are IP core node, IntGW-01 and IntGW-02 are an internet gateway router peered with other service providers (SPs), and RR-01 is a router reflector sharing route information.

C. Dataset

The dataset used in this paper contains the four types of datasets, i.e, information on failure management, on virtual infrastructure, on physical infrastructure, and on network devices, which is provided at [15]. In [7], the authors developed the dataset generator for analysis on route information failure in IP core networks by NFV-based test environment since it is hard to collect a large quantity of data in the IP core network. The dataset generator has two types of functions, i.e., a data generator and a failure generator. The data generator collects and stores the four types of datasets every minute from the target network. The failure generator intentionally causes a failure and recovery event, which occurs alternately at the interval of five minutes. Table I presents the failure use cases in this problem. The failure generator considers five use cases for failure scenarios as shown in Table I. The training data includes all possible failure use cases while the testing data includes a part of use cases randomly selected. The dataset contains not only stable data but also unstable data because of the data collection principles. Note that the dataset does not include the explicit information of the packet loss or delay, e.g., round trip time.

IV. Proposed Approach

A. Preliminaries

We consider the physical topology $\hat{G} = (\hat{V}, \hat{E})$ as in Fig 1 where $\hat{V}$ denotes a set of physical nodes and $\hat{E}$ denotes a set of links. Since it is hard to apply the physical topology to the GNN directly, we conduct the graph transformation. With

[1] https://www.ieice.org/~rising/AI-5G/download/Theme1-slide.pdf

TABLE II: Node features used for training

Feature	Denifition	Type	Scaler
cpu-util	CPU utilization	Scalar	MinMax
admin-status	Interface status	Categorical	-
network-incoming-packet-rates	Network incoming packet rates	Scalar	Standard
network-outgoing-packet-rates	Network outgoing packet rates	Scalar	Standard
tx-pps	TX packet per second	Scalar	Standard
rx-pps	RX packet per second	Scalar	Standard
prefix-activity-received-current-prefixes	Information on prefix activity	Scalar	Standard

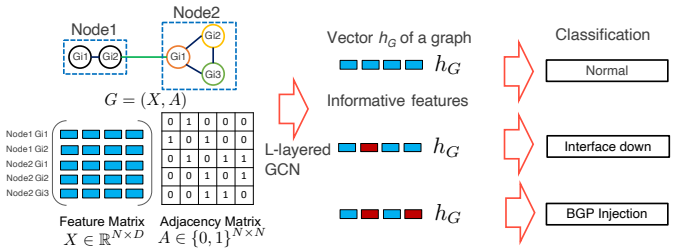

Fig. 3: Overview of the proposed model.

helps of the simply graph transformation, we can apply the transformed graph $G = (X, A)$ to the GNN where $X \in \mathbb{R}^{N \times D}$ denotes a feature matrix and $A \in \{0,1\}^{N \times N}$ denotes an adjacency matrix. Here, N denotes the number of nodes on the transformed graph and D denotes the number of features. Note that that graph transformation may not be necessary if we use a heterogeneous (relational) graph representing the physical topology, which is our future work.

Fig 2 depicts the example of the graph transformation. We first regard each interface as a node and each physical link as a link. Then, we generate links between interfaces on the physical node v as a complete graph consisting of all interfaces on the physical node v. Finally, we add the specific features of the physical node v, i.e., cpu utilization, to the features of the corresponding interfaces on the physical node v.

B. Preprocessing

As for preprocessing, we first retrieve the stable dataset from the original dataset. Recall that the dataset contains not only stable data but also unstable data because of the data collection principles. Then, we divide the stable data into training data and validation one at a ratio of nine to one. Note that the ratio of failure labels is equivalent between the training data and validation one.

We retrieve the seven types of the stable data for machine learning, i.e., the cpu utilization, the interface status, the network incoming/outgoing packet rates, the tx/rx pps, and the prefix activity, as presented in Table II. We use the minmax and the standard scaler for normalization. Note that the missing value is set to be 0 except for the prefix activity. Since the prefix activity is useful information to detect BGP-related failures, we conduct the forward filling, which propagates the last valid observation forward.

C. Model

To tackle this problem, we interpret the failure classification as the graph classification by using the GCN. We propose a supervised graph classification with the GCN for the failure

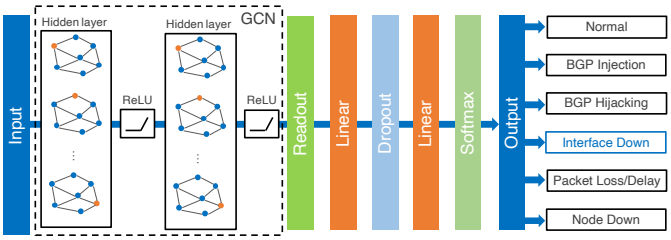

Fig. 4: Architecture of the proposed model.

classification in the IP core network to predict the failure type of an entire graph from its features. The proposed model conducts the anomaly detection and failure classification, which classifies the normal or the five types of failures, i.e., BGP injection, BGP hijacking, interface down, packet loss/delay, and NE reboot. As shown in Fig 3, our model classifies the five types of failures including the normal by leveraging the informative features generated by the L-layered GCN.

Fig 4 illustrates the architecture of the proposed model. The proposed model can be divided into two layers, which are known as the GCN-layers and the MLP-layer, respectively. The proposed model has two GCN-layers with the rectified linear unit (ReLU), which is an activation function commonly used in multilayer perceptron and considers the information on second order neighbors, which indicates the consideration of the information on physical nodes connected by the physical link on the target topology. To obtain the features of the entire graph, we use the readout as the mean aggregation. In the MLP-layer, the model classifies the failure type using the features representing the entire graph. We use the failure labels as the supervised data. Note that we have published the source code of the proposed model on Github[2].

V. Numerical Results

A. Evaluation Scenario

We use the physical topology as shown in Fig 1. As for dataset, we use the dataset provided at [15]. We implemented the proposed model in python language with Pytorch[3], dgl [16], and Networkx[4] libraries. As for experiments, we use the server with iMac (Retina 5K, 27-inch, 2019), 3.6 GHz 8-Core Intel Core i9, 64 GB 2667 MHz DDR4. For comparative purposes, we prepare four schemes, i.e., simple multilayer perceptron (MLP), XGBoost, Random forest (RF), and support vector machine (SVM).

As for the performance criteria, we use the following

[2]https://github.com/ITU-AI-ML-in-5G-Challenge/ITU-ML5G-PS-032-KDDI-naist-lsm
[3]https://pytorch.org/
[4]https://networkx.org/

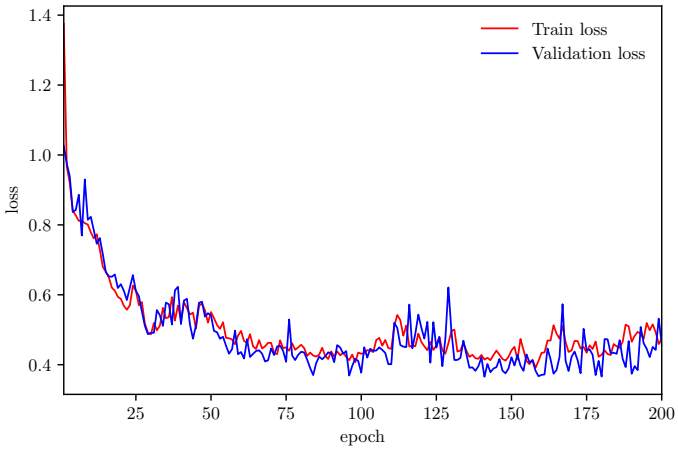

Fig. 5: Training loss.

TABLE III: Parameters used in the proposed model

Parameter	Value
Learning rate	1e-4
Number of epochs	200
Batch size	16
Dropout	0.1

metrics:

$$\text{precision} = \frac{TP}{TP+FP},$$
$$\text{recall} = \frac{TP}{TP+FN},$$
$$\text{f1-score} = \frac{2\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}},$$
$$\text{accuracy} = \frac{TP+TN}{TP+FP+TN+FN},$$

where TP, TN, FP, and FN denotes true positive, true negative, false positive, and false negative, respectively. As for the computational complexity, we also use the inference time where the trained model make a prediction.

B. Training

The proposed model is trained on the stable dataset as shown in Table II. Note that 90% (resp. 10%) of the dataset is used for training (resp. validation). We define the loss function as the cross entropy. We use the Adam [17] as the optimizer. Table III presents the parameters used in the proposed model.

Fig. 5 depicts the transition of train and validation loss. We observe that the train and validation loss drastically decrease at the beginning of training epochs and the both losses gradually decrease with increase of epochs.

C. Testing

Table IV presents the performance comparison among the five schemes, i.e., GCN (the proposed model), MLP, SVM, RF, XGBoost. We first focus on the accuracy. We observe that the GCN results in the higher accuracy compared with the other schemes. This result stems from the performance improvement for estimating the packet loss/delay. In particular, we confirm that the GCN improves the f1-score of the packet loss/delay by 0.07 in comparison with the MLP. This indicates that the explicit topology embedding contributes to the performance improvement for estimating the packet loss/delay. Recall that the stable dataset does not include the explicit information of the packet loss or delay.

However, we also confirm that all schemes except for the SVM exhibit almost the same f1-score of the BGP hijacking. As for future work, we plan to use not only the other features related to the BGP, e.g., information on as-path, but also the graph structures considering the neighbor SPs.

Next, we focus on the inference time. We confirm that the inference time with the GCN is much higher than other schemes except for SVM but the inference time with the GCN is enough small, i.e., 274 [ms].

Note that you can check the brief demonstration of the proposed model to play the video on Youtube[5].

VI. Conclusion

With the proliferation of 5G mobile networks, mobile operators should continuously provide the stable and high-quality internet services at the expense of the resource of highly experienced operators. To tackle the unexpected defect in the IP core network while reducing the operating expenditures, machine learning based network operations are required to operate automatically and rapidly.

In this paper, as an initial step toward the realization of the failure classification with the GCNs in the IP core network, we have investigated the potential of the explicit topology embedding in the training model for detecting and/or classifying the network status, i.e., the route information failures, the single point failures, and the packet loss/delay. We have proposed the supervised graph classification with GCNs by regarding the failure classification as the graph classification. Fundamental results have shown that (1) the GCN-based approach exhibits the higher accuracy compared with the other schemes, i.e., MLP, XGBoost, Random forest, and SVM, and (2) the explicit topology embedding contributes to the performance improvement for estimating the packet loss/delay.

As for future work, we plan to use the heterogeneous graph for the supervised graph classification. The heterogeneous graph can consider not only the physical topology but also the logical topology for the target network and enable us to use the GCN without the graph transformation. We also plan to find good features, e.g., as-path information, to improve the accuracy of BGP-related failures.

References

[1] R. N. Mitra and D. P. Agrawal, "5G mobile technology: A survey," *ICT Express*, vol. 1, no. 3, pp. 132–137, Dec. 2015.

[2] G. Choudhary, J. Kim, and V. Sharma, "Security of 5G-Mobile Backhaul Networks: A Survey," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 9, no. 4, pp. 41–70, Dec. 2018.

[5] https://www.youtube.com/watch?v=HqRSd6vzLb4

TABLE IV: Performance comparison

		Failure type							
scheme	criteria	normal	BGP hijacking	BGP injection	node down	interface down	packet loss/delay	accuracy	inference time [ms]
XGBoost	precision	0.93	0.68	1.00	0.92	0.88	0.75	0.89	20
	recall	0.91	0.99	0.99	1.00	0.85	0.70		
	f1-score	0.92	0.81	0.99	0.96	0.87	0.73		
RF	precision	0.90	0.70	1.00	1.00	0.91	0.75	0.87	8
	recall	0.91	0.99	0.99	1.00	0.95	0.64		
	f1-score	0.91	0.82	0.99	1.00	0.93	0.69		
SVM	precision	0.99	0.54	0.25	0.96	0.60	0.62	0.84	1319
	recall	0.82	1.00	0.95	1.00	0.96	0.86		
	f1-score	0.90	0.70	0.40	0.98	0.74	0.72		
GCN	precision	0.89	0.97	0.98	0.99	0.99	0.98	0.91	274
	recall	0.99	0.70	0.96	1.00	1.00	0.62		
	f1-score	0.94	0.82	0.97	1.00	1.00	0.76		
MLP	precision	0.90	0.97	0.97	1.00	0.96	0.66	0.87	17
	recall	0.91	0.71	0.92	0.99	0.96	0.73		
	f1-score	0.91	0.82	0.95	1.00	0.96	0.69		

[3] B. Al-Musawi, P. Branch, and G. Armitage, "BGP Anomaly Detection Techniques: A Survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 377–396, Firstquarter 2017.

[4] S. Cho, R. Fontugne, K. Cho, A. Dainotti, and P. Gill, "BGP Hijacking Classification," in *Proc. of 2019 Network Traffic Measurement and Analysis Conference (TMA)*, Jun. 2019, pp. 25–32.

[5] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph Convolutional Networks: A Comprehensive Review," *Computational Social Networks*, vol. 6, no. 1, p. 11, Nov. 2019.

[6] Z. Zhang, P. Cui, and W. Zhu, "Deep Learning on Graphs: A Survey," *arXiv:1812.04202 [cs, stat]*, Mar. 2020.

[7] J. Kawasaki, G. Mouri, and Y. Suzuki, "Comparative Analysis of Network Fault Classification Using Machine Learning," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*. Budapest, Hungary: IEEE, Apr. 2020, pp. 1–6.

[8] C. Sauvanaud, K. Lazri, M. Kaâniche, and K. Kanoun, "Anomaly Detection and Root Cause Localization in Virtual Network Functions," in *Proc. of 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, Oct. 2016, pp. 196–206.

[9] K. Qader, M. Adda, and M. Al-kasassbeh, "Comparative Analysis of Clustering Techniques in Network Traffic Faults Classification," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 4, pp. 6551–6563, 2017.

[10] F. Geyer, "Performance Evaluation of Network Topologies using Graph-Based Deep Learning," in *Proc. of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS 2017. New York, NY, USA: Association for Computing Machinery, Dec. 2017, pp. 20–27.

[11] T. Suzuki, Y. Yasuda, R. Nakamura, and H. Ohsaki, "On Estimating Communication Delays using Graph Convolutional Networks with Semi-Supervised Learning," in *Proc. of 2020 International Conference on Information Networking (ICOIN)*. Barcelona, Spain: IEEE, Jan. 2020, pp. 481–486.

[12] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura, "Deep Reinforcement Learning-Based Channel Allocation for Wireless LANs With Graph Convolutional Networks," *IEEE Access*, vol. 8, pp. 31 823–31 834, 2020.

[13] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv:1609.02907 [cs, stat]*, Feb. 2017.

[14] J. Zheng and D. Li, "GCN-TC: Combining Trace Graph with Statistical Features for Network Traffic Classification," in *Proc. of ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[15] RISING, "ITU AI/ML in 5G Challenge Global Round in Japan," https://www.ieice.org/~rising/AI-5G/, Accessed 1 Nov. 2020.

[16] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks," *arXiv preprint arXiv:1909.01315*, 2019.

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017.