

# Challenging the generalization capabilities of Graph Neural Networks for network modeling

José Suárez-Varela<sup>\*†</sup>, Sergi Carol-Bosch<sup>\*†</sup>, Krzysztof Rusek<sup>\*‡</sup>, Paul Almasan<sup>\*†</sup>, Marta Arias<sup>†</sup>,  
Pere Barlet-Ros<sup>\*†</sup>, Albert Cabellos-Aparicio<sup>\*†</sup>  
jsuarezv@ac.upc.edu

<sup>\*</sup>Barcelona Neural Networking Center, Universitat Politècnica de Catalunya, Barcelona, Spain

<sup>†</sup>Universitat Politècnica de Catalunya, Barcelona, Spain

<sup>‡</sup>AGH University of Science and Technology, Department of Telecommunications, Krakow, Poland

## ABSTRACT

Today, network operators still lack functional network models able to make accurate predictions of end-to-end Key Performance Indicators (e.g., delay or jitter) at limited cost. Recently, a novel Graph Neural Network (GNN) model called *RouteNet* was proposed as a cost-effective alternative to estimate the per-source/destination pair mean delay and jitter in networks. Thanks to its GNN architecture that operates over graph-structured data, *RouteNet* revealed an unprecedented ability to learn and model the complex relationships among topology, routing and input traffic in networks. As a result, it was able to make performance predictions with similar accuracy than resource-hungry packet-level simulators even in network scenarios unseen during training. In this demo, we will challenge the generalization capabilities of *RouteNet* with more complex scenarios, including larger topologies.

## CCS CONCEPTS

• **Networks** → **Network performance evaluation**; • **Computing methodologies** → *Machine learning*;

## KEYWORDS

Graph Neural Networks, Network Modeling

## 1 INTRODUCTION

Network modeling is essential to build efficient network operation and optimization solutions with special attention on future self-driving networks [3]. One fundamental characteristic of network optimization tools is that they can only optimize what they can model. For example, to optimize the end-to-end delay of traffic, it is necessary a network model able to understand how delay is related to other network characteristics (e.g., topology, configuration, traffic). In this context, nowadays network operators lack efficient network models able to make accurate predictions of relevant end-to-end Key Performance Indicators (KPI) such as delay or jitter at limited cost. Analytic models (e.g., Queuing Theory) fail to achieve accurate estimation in real-world scenarios with complex configurations (e.g., real traffic distributions, multi-hop routing) [8]. Alternatively, packet-level simulators produce accurate KPI predictions at the expense of high computational cost, which makes them useless for network operation in short timescales.

In the context of Machine Learning (ML), Neural Network (NN) models seem to be promising to build lightweight network models with good accuracy. However, early ML-based attempts did not

fulfilled yet its high expectations. The main reason behind this is that computer networks are fundamentally represented as graphs, and state-of-the-art proposals [2, 4, 6, 7] use well-known NN architectures (e.g., fully-connected, Convolutional Neural Networks) that are not well suited to model information structured as graphs.

Recently, a novel Graph Neural Network (GNN) model named *RouteNet* [5] showed to be an efficient solution to produce per-source/destination estimations of per-packet mean delay and jitter with similar accuracy as packet-level network simulators. Particularly, the GNN architecture behind *RouteNet* demonstrated its ability to learn the underlying relationships over graph-structured data and, as a consequence, it was able to achieve accurate performance inference in scenarios including topologies, routing schemes and input traffic different from those seen during training.

In this demo, we go beyond the evaluation of the original paper and show how *RouteNet* can exploit its generalization capabilities to achieve accurate performance prediction in more complex scenarios including larger topologies of variable size (up to 50 nodes).

## 2 NETWORK MODELING WITH GNN

Fig. 1 shows a schematic representation of the network model built with *RouteNet* [5]. The model has as inputs the network topology, a source-destination routing scheme and a source-destination traffic matrix (traffic shared by every node pair). As output, for every source-destination pair *RouteNet* produces accurate estimates of mean per-packet delay and jitter.

Contrary to other traditional NN models with fixed-dimension input (e.g., fully-connected, Convolutional Networks), GNNs have a dynamic internal architecture that is assembled at runtime based on the input graphs. This allows them to exploit information from the graph structures (i.e., elements and connections) and model the relationships among different graph elements. As a result, they develop a unique generalization capability which enables them to make predictions in graphs not seen during training.

In a networking context, GNN models can be used to model the relationships among different network elements (e.g., forwarding

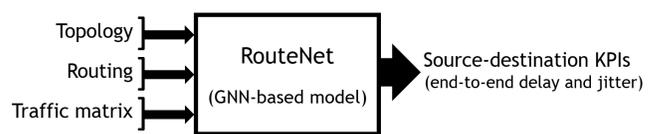


Figure 1: Schematic representation of *RouteNet*

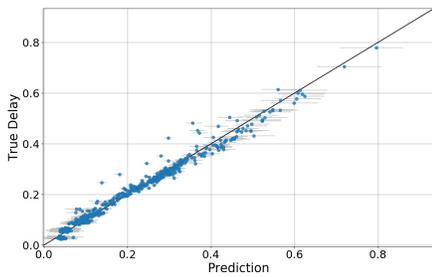


Figure 2: Regression plot in a sample scenario of Geant2

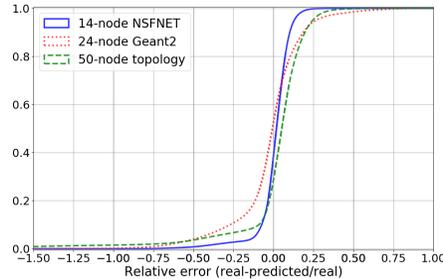


Figure 3: Cumulative Distribution Function (CDF) of the relative error

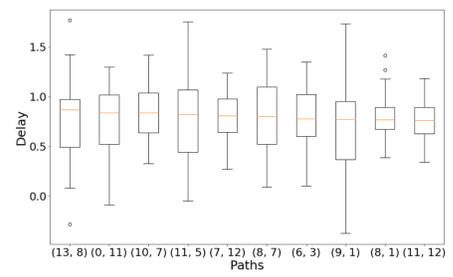


Figure 4: Screenshot of Top-10 paths with more delay

devices, links) and how they are related to different performance metrics. Thus, RouteNet leverages the generalization property of GNNs to model the complex relationships among the state of end-to-end paths and links resulting from topologies and routing schemes.

## 2.1 Training and evaluation

We use the original implementation of RouteNet and optimize a set of hyperparameters to adapt the model to scenarios with larger topologies and more complex routing schemes. Then, we train RouteNet to estimate delays on a dataset with 480,000 samples generated with a custom-built packet-level simulator in OMNeT++. This includes samples from two topologies: (i) the 14-node NSFNET topology and (ii) a 50-node synthetically-generated topology. Every topology was simulated with a wide variety of routing schemes and traffic matrices with different traffic intensity.

For the evaluation, we test the model on a different dataset with samples not present during training. Particularly, this evaluation dataset contains 120,000 unseen samples simulated in the two topologies where RouteNet was trained (14-node and 50-node). Additionally, to test the capability of RouteNet to generalize to topologies of variable size, we made a separate evaluation over a collection of 300,000 samples simulated in a third topology with 24 nodes (Geant2)<sup>1</sup>.

As a result, we observe that RouteNet produces accurate estimates even in unseen topologies. As an example, Fig. 2 shows a regression plot of the delays predicted by RouteNet in a sample scenario of the Geant2 topology. Also, in Fig. 3 we provide the Cumulative Distribution Function (CDF) of the relative error between RouteNet’s predictions and the real delay values in our evaluation datasets. Particularly, this plot includes the relative error over all the evaluation samples of the three different topologies.

## 3 DEMONSTRATION

In our demo, we will present an interactive Jupyter notebook [1] where we will show online the generalization capabilities of RouteNet to make predictions in scenarios with topologies up to 50 nodes, routings and traffic randomly generated. Also, we will show some examples leveraging the predictions of RouteNet for network visibility and planning. This includes visual figures representing the

delay on end-to-end paths and more elaborated statistics such as the Top-N paths with more delay. One example of this is the screenshot in Fig. 4.

During the conference, the Jupyter notebook will also be publicly available and will be ready to be executed directly in a cloud execution environment.

## ACKNOWLEDGEMENTS

This work was supported by the Spanish MINECO under contract TEC2017-90034-C2-1-R (ALLIANCE), the Catalan Institution for Research and Advanced Studies (ICREA) and the AGH University of Science and Technology grant, under contract no. 15.11.230.400. The research was also supported in part by PL-Grid Infrastructure.

## REFERENCES

- [1] [n. d.]. Project Jupyter. <https://jupyter.org/>. Accessed: 2019-07-28.
- [2] X. Chen, J. Guo, Z. Zhu, R. Proietti, A. Castro, and S.J.B Yoo. 2018. Deep-RMSA: A Deep-Reinforcement-Learning Routing, Modulation and Spectrum Assignment Agent for Elastic Optical Networks. In *Optical Fiber Communications Conference and Exposition (OFC)*.
- [3] Albert Mestres et al. 2017. Knowledge-defined networking. *ACM SIGCOMM Computer Communication Review* 47, 3 (2017), 2–10.
- [4] Albert Mestres, Eduard Alarcón, Yusheng Ji, and Albert Cabellos-Aparicio. 2018. Understanding the Modeling of Computer Network Delays using Neural Networks. In *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*. ACM, 46–52.
- [5] Krzysztof Rusek, José Suárez-Varela, Albert Mestres, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2019. In *Proceedings of ACM Symposium on SDN Research (SOSR)*.
- [6] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar. 2017. Learning to route. In *Proceedings of HotNets*.
- [7] Shihan Xiao, Dongdong He, and Zhibo Gong. 2018. Deep-Q: Traffic-driven QoS Inference using Deep Generative Network. In *Proceedings of the Workshop on Network Meets AI & ML*. ACM, 67–73.
- [8] Zhiyuan Xu et al. 2018. Experience-driven networking: A deep reinforcement learning based approach. In *Proceedings of IEEE INFOCOM*.

<sup>1</sup>The source code, the delay model already trained and the training/evaluation datasets used are publicly available at: <https://github.com/knowledgedefinednetworking/Papers/wiki/Challenging-the-generalization-capabilities-of-Graph-Neural-Networks-for-network-modeling>