Learning to Optimize: Edge-Based Graph Neural Networks Trained on MILP-Optimized Routing Paths

Saurav Mittal*
Centre for Development of Advanced Computing
Pune, Maharashtra, India.
sauravmittal131@gmail.com

Abstract

In this report, we demonstrate an edge-based GNN model capable of routing traffic in a network G(23, 74) through optimized paths. Routing decisions are computed using a Mixed Integer Linear Programming (MILP) formulation that minimizes maximum link utilization (MLU) while considering the current network state. The optimal path between two nodes is used as supervised ground truth for GNN training. The study utilizes four months of real-world traffic data from the GEANT network to train the model. The proposed GNN achieves a mean absolute error of 0.0745 when predicting the MILP-optimized routing paths. This demonstrates that the model successfully learns to optimize the routing decisions in real time, offering a significant computational advantage over conventional optimization-based methods that are infeasible for runtime deployment.

1 Introduction

Modern communication networks face highly dynamic traffic conditions that demand adaptive routing strategies to minimize congestion and optimize utilization. Conventional optimization-based approaches, such as Mixed Integer Linear Programming (MILP), can compute globally optimal paths for traffic demands but are computationally expensive and impractical for real-time operation. As a result, most deployed routing methods rely on hand-crafted heuristics, which are difficult to design and often fail to generalize under changing network states. As reported in recent work by Reis et al. [2019] introducing a deep learning framework that learns routing decisions directly from MILP-generated solutions, replacing manually designed heuristics with a data-driven alternative.

In this study, we extend this framework by introducing an edge-based Graph Neural Network (GNN) that operates directly on the network topology. The GNN learns to optimize the network from MILP derived routing decisions and predicts the optimal path based on current network conditions. This approach maintains the accuracy of MILP-based optimization while providing faster inference, enabling practical use in real-time and scalability.

1.1 Dataset and Preprocessing

For this study, we used The GEANT network datasetUhlig et al. [2006] with topology (see figure:3) of 23 nodes and 74 edges. The dataset includes 11460 Traffic Matrics (TM), each corresponding to a period of 15 minutes totaling 4 months of observations. TM are aggregated traffic information or snapshot of network and some assumptions had to be made to generate a sequence of flows such that when combined they translate to the same TM. Following this we generate flows, each with source node, destination node, Bandwidth, Duration, and time of arrivalReis et al. [2019] and the procedure is then repeated for the next TM and the time of arrival distribution is shifted by 900s or 15 min.

*Corresponding author: sauravmittal131@gmail.com

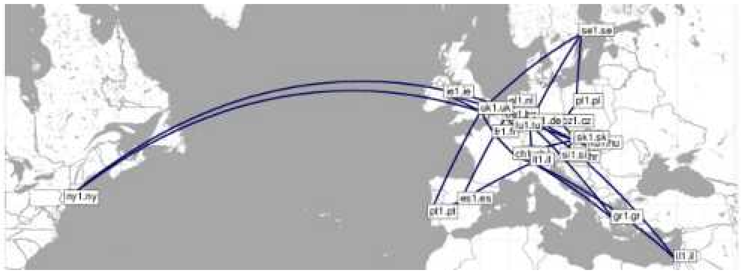
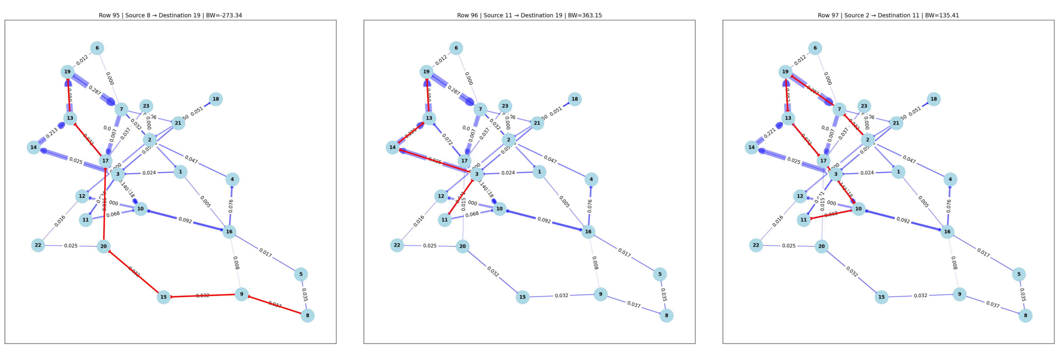


Figure 1: GEANT network topology used for MILP-based routing optimization.

1.2 Mixed Integer Linear Programming (MILP) for Dynamic Routing

Once the flow sequences are generated, we define a MILP formulation to minimize the maximum link utilization (MLU). Starting with zero edge utilization, flows are processed sequentially to generate optimized paths subject to edge capacity constraints and path continuity conditions. After each flow arrival, the utilization of all associated edges is updated, and the list of active flows is maintained. Before processing a new arrival, any flows that have reached their departure time are removed, and the utilization of their corresponding edges is adjusted. For each flow event—arrival or departure—the optimal path selected by the MILP is stored along with the instantaneous network state, forming the supervised dataset later used for GNN training.



(a) Flow Departure. (b) Flow Arrival. (c) Flow Arrival.

Figure 2: MILP-based routing in sequence. Thickness of each link indicates its current utilization.

2 Methodology and results

We define an edge-based Graph Neural Network (GNN) with a fixed network topology. Node features consist of one-hot source and destination indicators; edge features comprise the current utilization (ratio of utilization and capacity) and the requested bandwidth (bandwidth is negative for departures). The model predicts the routing path for each incoming flow as a 74-dimensional binary vector (1 = link used, 0 = not used). From our dataset we construct approximately five million flows, each labeled with the unique MILP-derived path used for training (see Figure 3). The trained model achieves a mean absolute error of ≈ 0.074 on edge-based predictions, indicating strong agreement with MILP paths and demonstrating that the GNN successfully learns the optimization policy for real-time routing.

3 Conclusion

This study presents an edge-based Graph Neural Network (GNN) that learns to replicate MILP-optimized routing decisions from real network data. Trained on four months of GEANT traffic and approximately five million MILP-optimized routing paths the model achieves a mean absolute error

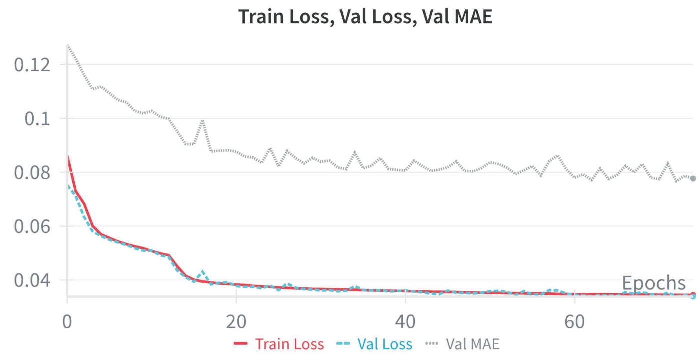


Figure 3: GEANT network topology used for MILP-based routing optimization.

of 0.074 in predicting routing paths. These results demonstrate that the proposed GNN effectively learns to optimize.

References

Joao Reis, Miguel Rocha, Truong Khoa Phan, David Griffin, Franck Le, and Miguel Rio. Deep neural networks for network routing. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019. doi: 10.1109/IJCNN.2019.8851733.

Steve Uhlig, Bruno Quoitin, Jean Lepropre, and Simon Balon. Providing public intradomain traffic matrices to the research community. *SIGCOMM Comput. Commun. Rev.*, 36(1):83–86, January 2006. ISSN 0146-4833. doi: 10.1145/1111322.1111341. URL `https://doi.org/10.1145/1111322.1111341`.