Project Proposal - Predicting Critical Roads with Traffic Jams

Dario Vajda, Oliver Majer, Diego Bonaca

University of Ljubljana

1. Introduction

In this project, our goal is to train a model that predicts which road segments are likely to cause traffic jams inside of any large city network. The learning signal will come from multiple sources, the actual architecture of the given city, population density and traffic data. If such a model was shown to be reliable, it could be used in many downstream tasks, not limited to existing networks – it could be involved in the planning process and detect problems before any physical construction.

2. Data

2.1 City Networks

For the city networks data, we will use OpenStreetMap (OSM). It has an already built Python library OSMnx [1] which is made to easily download, model, analyze, and visualize street networks and other geospatial features from OpenStreetMap. Both OSM and OSMnx are publicly available. This will be the basis for our spatial network depiction of the city.

2.2 Population Density

For the population density data we will use the High Resolution Settlement Layer (HRSL) [2] developed by Meta (Facebook) in collaboration with CIESIN. It is one of the most comprehensive global population maps available to the public, with a spatial resolution of 1 arc-second (~30m×30m). Usually published as GeoTIFF raster files, the dataset can alternatively be converted to CSV (latitude, longitude, population) format. By capturing where people are expected to be located within the city, these fine-grained population estimates will help our model become more accurate.

2.3 Traffic Data

We will be using the UTD19 dataset created at ETH [3], which provides detailed traffic flow data in CSV format, gathered from more than 20,000 stationary detectors on city highways, for traffic information. It is one of the largest publicly available multi-city traffic datasets. We will train and evaluate our traffic prediction models using this dataset as the ground truth.

3. Model

Cities are large and complex networks of roads. To achieve our goal of predicting which roads are critical and are most likely to cause traffic jams, we will need a model which is expressive enough, and which scales well for large networks. Moreover, we are solving a problem in an inductive setting, where we train the model on certain networks and test on others. Therefore, our model has to be both expressive and it has to generalise well. There are two main options for us to consider:

1. Message Passing Graph Neural Network (MP-GNN) - to maximise expressivity and the model's ability to learn complex patterns, the local attention mechanism would be utilised.
2. Graph Transformers [4] - these models include global self-attention across all nodes in the graph, which increases the learning capabilities of the model, especially for long-range dependencies, but at the cost of higher memory usage and lower compute efficiency.

To choose the model which would be most suitable for our problem, we have to consider both the nature of the graphs we will be working with and the hardware limitations. Hence why we can not use the Graph Transformer, because the memory requirements scale quadratically, $O(n^2)$ with respect to the number of nodes, which is large in big cities. On the other hand, all MP-GNNs scale close-to-linearly, $O(n)$ on sparse graphs, such as road networks.

Among many options that fall into MP-GNN category (such as GCN [5], GraphSAGE [6], GIN [7], etc.), we will use a Graph Attention Network (GAT) [8], for its expressive power in comparison to using simpler message-passing methods. GATs do not assign the same importance to each neighbor, but they are weighted by learnable functions depending on the node's features and the new embedding is the weighted sum over its neighbors.

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

(1) Attention between nodes *i* and *j*

$$\vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right)$$

(2) Next layer embedding

4. Metrics

We will evaluate the model as a binary classifier on road segments, predicting whether each segment is *critical* (high congestion probability) or *non-critical*.

The primary metrics will be the precision, recall and the F_2-score ($\beta = 2$) — emphasizes recall over precision, since missing a critical road (false negative) is more costly than a false alarm [9]. Secondary metrics could involve ROC AUC [10], Recall@k% [13], Brier [11] score and Expected Calibration Error (ECE) [12] to further evaluate the quality of our classification model.

$$F_\beta = (1 + \beta^2)\frac{PR}{\beta^2 P + R}$$

(3) Formula for calculating the F_2-score

We will divide the dataset by city to evaluate the model's inductive generalization, thus, ensuring that cities used for testing are completely unseen during training. A reasonable split would have 70% data used for *training*, 10% for *validation* and the remaining 20% for *testing*. The *validation* set would be used for hyperparameter tuning and selecting the model checkpoint and the classification threshold that maximise the F_2-score. Finally, the model will be tested and evaluated on the held out *test* subset of the data which allows us to measure how well the model generalizes to new urban networks.

5. References

[1] Boeing, G. (2025). Modeling and Analyzing Urban Networks and Amenities with OSMnx. Geographical Analysis, published online ahead of print. doi:10.1111/gean.70009

[2] Facebook Connectivity Lab and Center for International Earth Science Information Network – CIESIN – Columbia University. 2016. High Resolution Settlement Layer (HRSL). Source imagery for HRSL © 2016 DigitalGlobe. Accessed 20.10.2025.

[3] Institute for Transport Planning and Systems, ETH Zurich, https://www.nature.com/articles/s41598-019-51539-5

[4] Dwivedi, Vijay Prakash, et al. "Graph Neural Networks with Learnable Structural and Positional Representations." International Conference on Learning Representations (ICLR), 2021.

[5] Kipf, Thomas N., and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks." International Conference on Learning Representations (ICLR), 2017.

[6] Hamilton, William L., Rex Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs." Advances in Neural Information Processing Systems (NeurIPS), 2017.

[7] Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. "How Powerful are Graph Neural Networks?" International Conference on Learning Representations (ICLR), 2019.

[8] Veličković, Petar, et al. "Graph Attention Networks." International Conference on Learning Representations (ICLR), 2018.

[9] Van Rijsbergen, C. J. (1979). Information Retrieval (2nd ed.). Butterworth-Heinemann.

[10] Fawcett, T. (2006). "An introduction to ROC analysis." Pattern Recognition Letters, 27(8), 861–874.

[11] Brier, G. W. (1950). "Verification of forecasts expressed in terms of probability." Monthly Weather Review, 78(1), 1–3.

[12] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). "On Calibration of Modern Neural Networks." International Conference on Machine Learning (ICML).

[13] Pullak, K. (2021). *Understanding Precision, Recall, and F-Score@K in Recommender Systems.*Medium.Retrieved.from:https://krishnapullak.medium.com/understanding-precision-recall-and-f-score-at-k-in-recommender-systems-7146a0dce68e