Edge classification of transportation mode in public transport network

Gioele Luca Monopoli, Pengkang Guo
Network Machine Learning, EPFL, Switzerland

***Abstract*—This study explores edge classification of transportation modes in public transport networks, analysing both traditional machine learning methods and Graph Neural Networks (GNNs) in a class imbalance context. Utilizing datasets from 25 global cities, the research delves into the complexities of network structures and transportation modes. Initial experiments on Berlin's network using Support Vector Machines and Node2Vec precede the broader application of GNNs across multiple cities. The study addresses class imbalance, by doing a comparison across models, laying the foundation for improved transportation network analysis with GNNs.**

I. Introduction

Public transport networks are essential for urban mobility, and accurately predicting transportation modes within these networks has significant implications. Traditionally, machine learning techniques have been used for this prediction task, but recent advancements in graph analysis and deep learning have introduced Graph Neural Networks (GNNs) as a powerful alternative. GNNs have demonstrated success in various domains and offer the potential to model complex relationships within graph-structured data. In this study, we compare the performance of traditional ML techniques and GNNs using a comprehensive dataset comprising public transport network data from 25 cities, including edge connectivity and transportation mode labels with class imbalance. By exploring the potential of GNNs in transportation mode classification, we aim to uncover new insights and advance the field of public transport network analysis.

We start by giving an exploratory analysis of the dataset (section II). We then present our solution (section III) to tackle this edge prediction problem, along with our results (section IV). Finally, we conclude by discussing the results and providing possible future work (section V).

II. Exploratory Analysis

A. Dataset

The public transport network dataset [1] comprises public transport networks of 25 cities across the world. As the scope of the project, we will focus only on a small part of this dataset. The data used in this paper is being merged and preprocessed from the different available files. It is composed of edge connections between nodes, along with edge attributes, respectively the distance between edges and the number of vehicles that have passed that edge during a registered period of time. The label of an edge, meaning the type of transportation mode which the edge covers, is also given. In more detail, the transportation mode of an edge could be of several types. The types in our processed dataset are five: tram (0), subway (1), rail (2), bus (3), and ferry (4). Moreover, an edge could be composed of multiple of these modes, leading to a multi-label classification problem. For example, an edge connecting two stations could have both a bus and a tram.

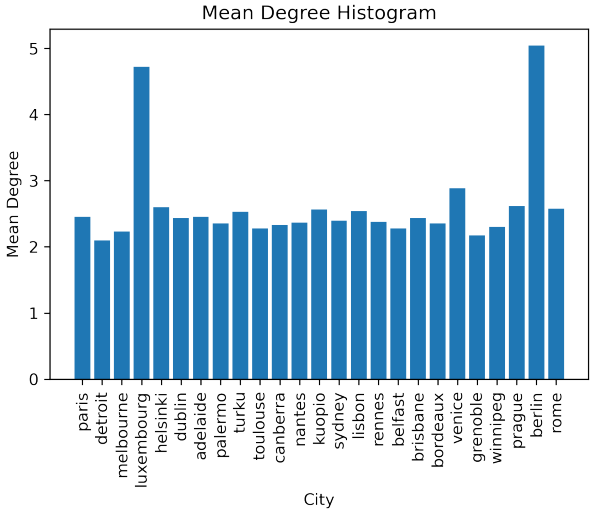

Figure 1. Average degree across all 25 cities

From the given dataset, we initially thought of trying to merge stations that are very close to each other (with a threshold, e.g, 100m of distance), as there might be multiple stations in a single big station (eg. the main station). After having calculated the distances for each station, we realised it was not making good sense, as some stations could have been really close to each other and not been part of a bigger station.

B. Graph analysis

We analyze the degree across all the 25 cities graphs. In Figure 1 we can see the mean degree, which shows us that nodes are generally with low degrees, with an average of 3. Figure 2 gives us a broader overview, from where we can also see in the long tail that there are hubs in the network with a larger number of degrees. Not being captured by these figures, we analyze these hubs in Table I, which comprises the two most important nodes (hubs) across

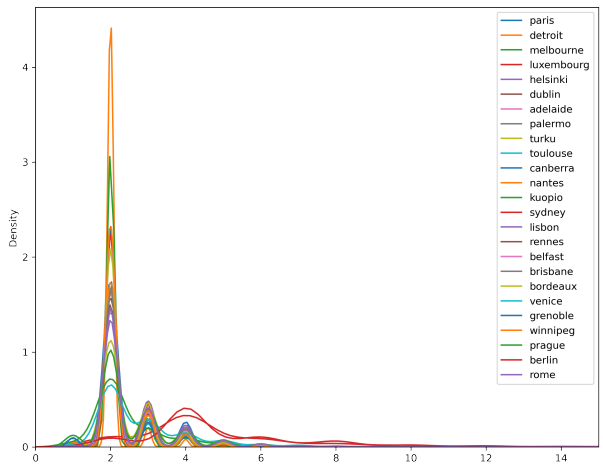

Figure 2. Degree distributions across all 25 cities

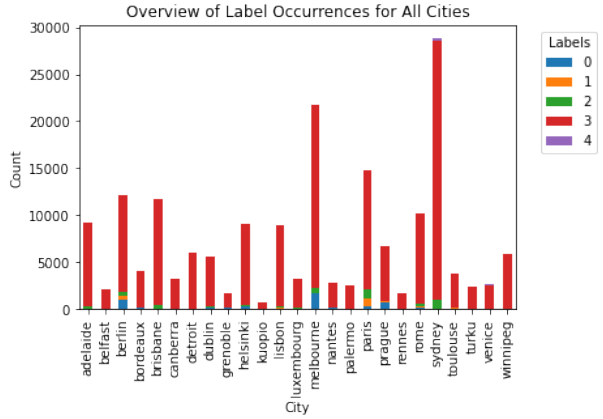

Figure 3. Label occurrences per cities

all 25 cities.

Performing edge analysis on a graph is a crucial step in understanding congestion patterns. By examining the edges, one can gain valuable insights into the flow and bottlenecks within the network. Because of this, we calculate edge betweenness centrality in the graph to find these important edges. We do this for only one city, Berlin, for computational reasons. Our findings indicate that the following edges in this network are relevant:

- S+U Gesundbrunnen Bhf (Berlin) and S+U Lichtenberg Bhf (Berlin),
- S+U Berlin Hauptbahnhof (tief) and S+U Potsdamer Platz Bhf (Berlin),
- S+U Berlin Hauptbahnhof (tief) and S+U Gesundbrunnen Bhf (Berlin).

C. Label analysis

It is important to have an understanding of the distribution of edge labels (transportation modes) in our analysis. As depicted in Figure 3, we observe a significant class imbalance across the five transportation modes. The majority of edges in the dataset correspond to buses, whereas the remaining transportation types constitute only a small percentage of entries. This issue will be addressed in detail in the following section, where we discuss our approach to handling class imbalance.

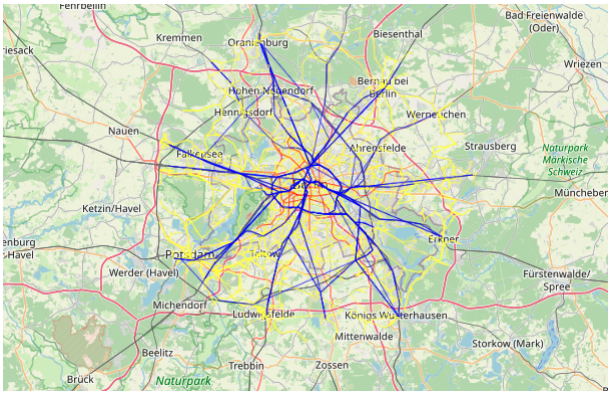

Figure 4. Network graph for the Berlin area. Blue line = rail, Yellow line = bus, Dark-red line = subway.

III. Methods

In this section, we are going to report the methods used to perform edge classification.

We start by only applying some simpler methods on a single city, to then apply more advanced methods (graph neural networks) to all cities.

A. Single City Traditional Machine Learning

In the initial phase of our edge prediction analysis, we employ traditional machine learning techniques on a single city, Berlin, which structure can be seen in Figure 4.

1) Handcrafted features with SVM

This approach involves extracting relevant graph features from the transportation network and utilizing Support Vector Machines (SVM) to model and predict edge transportation mode. We thus extracted the following node features from the graph: degree, clustering coefficient, closeness centrality, betweenness centrality, and eigenvector centrality. Once the node features were ready, we created edge features by applying an average of features between the two nodes creating the edge, along with the betweenness centrality of an edge.

2) Node2Vec with SVM

In addition to the handcrafted features approach, we also applied the Node2Vec algorithm to the transportation network of Berlin. Node2Vec is a graph embedding technique that learns continuous representations (embeddings) of nodes in a network by preserving the neighborhood structure. It allows us to capture the complex relationships and similarities between nodes in a low-dimensional space.

City	Top 1 Node	Top 2 Node
Paris	Bourg-la-Reine	JUVISY
Detroit	Moross & Mack	State Fairgrounds Transit Center
Melbourne	North Melbourne Railway Station (West Melbourne)	Sunshine Railway Station (Sunshine)
Luxembourg	Eich, Eecher Plaz	Kirchberg, John-F.-Kennedy
Helsinki	Lapinrinne	Viikki
Dublin	Dublin City South, Ashfield House	Trinity College, Shaw Street
Adelaide	Adelaide Railway Station	Stop 24 Crafers Ramp - South side
Palermo	PIAZZALE JOHN LENNON MONTE	RESUTTANA
Turku	Cygnaeuksen koulu	Brahenkatu
Toulouse	Eisenhower	Mesplé
Canberra	London Cct Commonwealth Bank	Canberra Centre City
Nantes	CIFAM	Druides
Kuopio	Tullikuningas L	Kys - Piha I
Sydney	Circular Quay, Wharf 4	Strathfield Station, Platform 6
Lisbon	Calçada Carriche (Restaurante)	Odivelas (R M Caldas Xavier) Centro Comercial
Rennes	Fac de Droit	Hôtel Dieu
Belfast	Queens Square	Connswater
Brisbane	Roma Street busway, platform 1	Cultural Centre, platform 1
Bordeaux	Quai Deschamps	Palais de Justice
Venice	Zattere SX	MESTRE CENTRO B2
Grenoble	CLAIX, POMPIDOU	GRENOBLE, LE RONDEAU
Winnipeg	Southbound Empress at Eastway	Eastbound Portage at Tylehurst (Polo Park)
Prague	Lihovar	Lihovar
Berlin	S+U Gesundbrunnen Bhf (Berlin)	S+U Zoologischer Garten Bhf (Berlin)
Rome	Roma Tiburtina	Ciampino

Table I
LIST OF SAMPLE LOCATIONS IN DIFFERENT CITIES

The node embedding learned by Node2Vec is passed again in an SVM model for edge classification.

B. Graph Neural Networks across Multiple Cities

Building upon the insights gained from the single city analysis, we expand our analysis to multiple cities by leveraging the power of Graph Neural Networks (GNNs).

To apply GNNs to our edge classification task, we extracted each city's transportation network nodes and edges, node features (such as a degree), and edge attributes (the distance of two nodes compromising an edge, i.e. its lenght) and merged them in one big disconnected graph. This is then split into training, validation, and testing.

It is well known that neural network models trained on imbalanced data may overfit the majority class very quickly. Because of this, for the training set, we applied downsampling of the majority class (3), reaching a more balanced set: 4203 entries for tram, 1265 entries for subway, 3594 entries for rail, 13748 entries for buses, and 503 entries for ferries. Note that this is only applied to the training set.

For the GNNs structure, we mainly focused on two state-of-the-art structures, GCN (Graph Convolutional Networks [2]) and GAT (Graph Attention Networks [3]).

1) GCN

Our use of GCN, based on the concept of Semi-Supervised Classification with Graph Convolutional Networks, allows us to leverage node features and the overall graph structure. By propagating node features through the graph, the GNN captures local topological information.

2) GAT

The Graph Attention Network (GAT), on the other hand, introduces the concept of attention mechanisms to graph processing. It weighs node's neighbors based on their importance, giving us a flexible model that can better understand and leverage the structure of the transportation network.

IV. RESULTS

This section presents the outcomes of the edge classification methods presented. We compare the results obtained from traditional machine learning techniques applied to a single city, with the more advanced graph neural network models across multiple cities. Our metrics focus on F1 score, which we've derived from test sets.

A. Single City Traditional ML

In the single city analysis of Berlin, the SVM model with handcrafted features provided a decent starting point with an accuracy (F1 Score) of 78.01%. In figure 5 we can see how the model predicted across the various classes. It is worth noting that no operation to tackle class imbalance

Model	Precision	Recall	F1-Score
Hand. Feat. SVM (Single City)	0.81	0.81	0.78
Node2Vec SVM (Single City)	0.90	0.88	0.89
GCN (All Cities)	0.93	**0.93**	**0.93**
GAT (All Cities)	**0.95**	0.76	0.84

Table II
PERFORMANCE OF MODELS ON TEST SETS

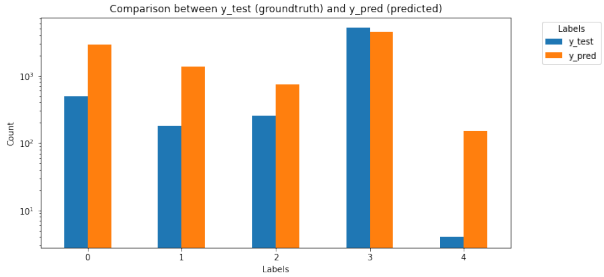

Figure 5. SVM with handcrafted features prediction against groundtruth

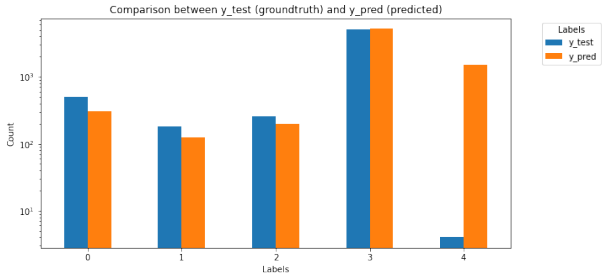

Figure 6. SVM with Node2Vec prediction against groundtruth

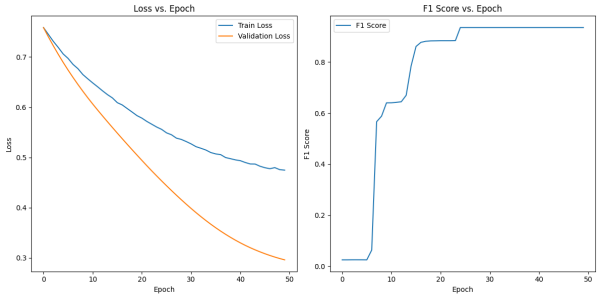

Figure 7. Plots for GCN. On the left, training and validation loss across epochs. On the right, the F1-Score on the test set

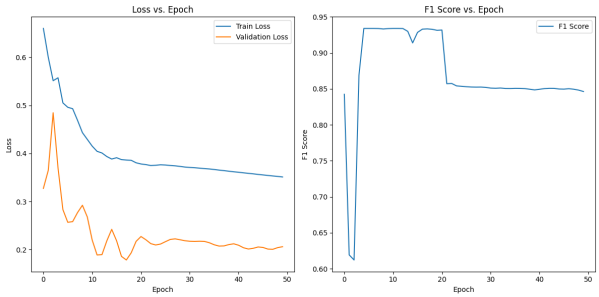

Figure 8. Plots for GAT. On the left, training and validation loss across epochs. On the right, the F1-Score on the test set

was taken here, and thus the model was able to predict fairly across classes, not being discriminative about a single class.

The Node2Vec approach, on the other hand, improved our performance, reaching an F1 Score of 88.9%. The embeddings captured the nuanced neighborhood information and led to a significant increase in the performance of our SVM model. Again, in figure 6 we can see that also Node2Vec is not discriminative against a single class despite the class imbalance.

B. Multiple Cities GNNs

For the graph neural networks applied to all cities, both GCN and GAT architectures were trained and tested. The GCN, with a hidden layer of 16 channels, was trained for 100 epochs and a learning rate of 0.002 and resulted in an accuracy of 93.2%. This could represent a considerable improvement over the single city traditional machine learning techniques, but it is to note that a comparison cannot be made as in the GNN case we are handling 25 cities instead of one. We can see the plots of the losses and F1 score across epochs in Figure 7. As we can see from Figure 9, even after appropriate downsampling to reduce class imbalance, the model is overfitting extremely the alleviated class imbalance, discriminating completely all the classes except transport mode 3 (buses).

The GAT model, which was trained for 50 epochs with a hidden dimension of 256 and a learning rate of 0.005, achieved an accuracy of 82.1%. Losses and f1 score across epochs are shown in Figure 8. The most important point from GAT is the non-discriminative behavior on the test set. In fact, as we can see from Figure 10, GAT class imbalance prediction is much better than the one from GNN, even if it is at the cost of a lower F1-score. Intriguingly, we can see from Figure 8 that initially, similar to GCN,

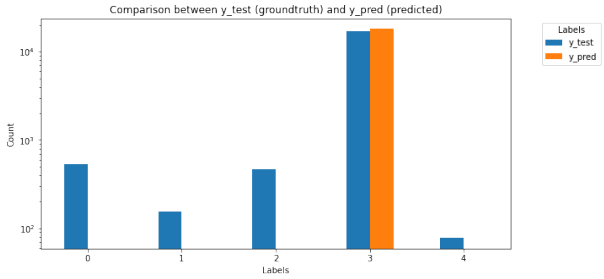

Figure 9. GNN prediction against groundtruth

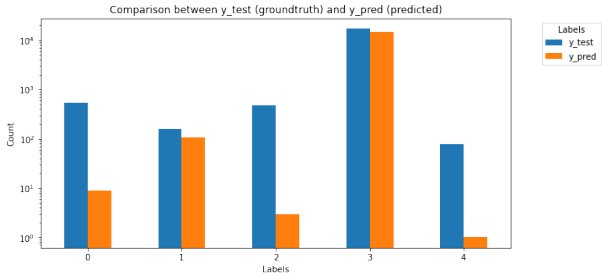

Figure 10. GAT prediction against groundtruth

the GAT model exhibits overfitting towards the majority class. However, as the epochs progress (around 22 epochs), the model gradually reduces its discriminatory power. This performance is attributed to the model's ability to weigh the neighbors of a node according to their importance, thus capturing the complex intercity transportation patterns more accurately. This model is overall a better model but this may depend on the application in which it is used.

V. Conclusions and Future Works

The paper observed two main approaches in predicting transportation modes in public transport networks. It focuses on the class imbalance problem of this kind of task and shows differences between simpler models, such as SVM with handcrafted features and node2vec, against more advanced models such as GNNs. While GNNs show promise, there's still a need to address the significant class imbalance observed in the data, which appears to affect the model's ability to accurately represent all transportation modes. Future research should focus on addressing these challenges and expanding the applicability of GNNs in transport network analysis.

In future work, it would be interesting to consider applying Markov Chains to model passenger transition probabilities between different transport modes. This could aid in identifying and understanding edges prone to congestion in the network.

References

[1] R. Kujala, C. Weckström, and R. Darst, "A collection of public transport network data sets for 25 cities," Feb. 2018, The licensing terms are documented separately for each city. [Online]. Available: https://doi.org/10.5281/zenodo.1186215

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[3] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018.