Forecasting crowding pattern evolution at subway stations using opportunistic data

A thesis presented in part fulfilment of the requirements of the Degree of Master of Science in Transportation Systems at the Department of Mobility System Engineering, TUM School of Engineering and Design, Technical University of Munich.

Supervisor Dr. Qinglong Lu
M.Sc. Ningkang Yang
Univ.-Prof. Dr. Constantinos Antoniou
Chair of Transportation Systems Engineering

Submitted by Wanrong Hu

Submitted on Munich, 30.09.2024

Declaration

I hereby confirm that the presented thesis work has been done independently and using only the sources and resources as are listed. This thesis has not previously been submitted elsewhere for purposes of assessment.

Munich, 30.09.2024, Wanrong Hu

Munich, 30.09.2024, Signature

Abstract

Understanding crowdedness patterns at public transport stations is crucial for enhancing operational efficiency, providing better service, and promoting the use of public transit. However, forecasting crowdedness accurately, especially in non-regular scenarios such as holidays or special events, presents significant challenges. Moreover, data availability and accessibility are often limited, further complicating the problem.

In this research, we analyze crowdedness patterns at urban public transport stations and develop a spatio-temporal dataset construction pipeline that integrates various data sources to capture both spatial and temporal characteristics of the transit environment. Specifically, we utilize Google Popular Times (GPT) data as the crowdedness data, which is open-source and easy to access. Subsequently, we evaluate the performance of various prediction models, including traditional statistical models, time series models, and our proposed spatial-temporal model with a graph attention module enhanced by a position embedding mechanism (APT-GCN). This GNN-based model is designed to effectively evaluate crowdedness patterns and measure shifts in crowdedness within the network. Furthermore, the attention weights from the attention layer are further utilized to model the crowdedness shift.

Our experimental design includes evaluating model performance under both regular and special event scenarios to assess robustness. Results show that the proposed APT-GCN model outperforms all baseline models, achieving superior performance in both regular and special event scenarios. Additionally, the crowdedness shift modeling reveals unique patterns of passenger flow transfers across different cities and metro lines. Our findings contribute to a deeper understanding of crowdedness at public transport stations and provide valuable insights for urban planners and transport operators to optimize station management and enhance the overall public transport experience, particularly under challenging, non-regular conditions.

Acknowledgement

I would like to express my sincere gratitude to the Chair of Transport Systems Engineering at TUM and Univ.-Prof. Dr. Constantinos Antoniou, for providing me the opportunity to work on this engaging topic as my master's thesis. Your guidance and support have been invaluable, and I am truly thankful.

A special thanks go to my supervisor, Dr. Qinglong Lu and Ms. C. Ningkang Yang, for their relentless support and patience. Your insightful advice, detailed feedback, and endless encouragement have been crucial in the completion of this thesis. I have learned so much from you both, not just in terms of academic knowledge but also personal growth and professional conduct.

I would also like to express my appreciation to TUM, the city of Munich, the places I visit, and the person I was two years ago who decided to come here to study. One thing I've learned over these two years is that you only meet extraordinary people in extraordinary places, and Munich has been such a place for me. To my friends, both here and elsewhere, thank you for creating wonderful memories together. Mia San Mia! Lastly, I want to thank my parents and my cat, for your unwavering support from afar. This accomplishment would not have been possible without your love and support.

Contents

List of Figures

List of Tables

Glossary

A

ADSTGCN - Adaptive Deeper Spatio-Temporal Graph Convolutional Network 17

AFC - Automatic Fare Collection 20, 23, 94

AIC - Akaike Information Criterion 58

APT-GCN - Spatial-Temporal Model with Graph Attention Module Enhanced by Position Embedding Mechanism 5–7, 25, 43, 44, 48, 57, 59, 62, 85–89, 94, 95

ARIMA - Auto-Regressive Integrated Moving Average 6, 7, 11, 15, 16, 19, 26, 39–41, 57–59, 61, 77–80, 83, 88

ARIMAX - Autoregressive Integrated Moving Average with Exogenous Variables 41

B

BATF - Bayesian Augmented Tensor Factorization 56

BPMF - Bayesian Probabilistic Matrix Factorization 56

C

CNN - Convolutional Neural Network 43

D

DCRNN - Diffusion Convolutional Recurrent Neural Network 15, 17

DTW - Dynamic Time Warping 31, 32, 34

G

GBR - Gradient Boosting Regression 35, 36, 75, 76

GCN - Graph Convolution Network 43, 44, 46

GNN - Graph Neural Networks 5, 11, 12, 15–20, 22, 25, 43, 85, 94

GPR - Gaussian Process Regression 35, 36, 75

GPT - Google Popular Time 54, 55, 57, 64, 94

GRU - Gated Recurrent Unit 27, 43, 45, 46

I

ITS - Intelligent Transport Systems 11, 14

K

KNN - K-Nearest Neighbors 7, 15, 35, 36, 57, 58, 75–77, 88

L

LSTM - Long Short-Term Memory 5, 7, 15, 16, 26, 27, 39–43, 45, 46, 57, 59–62, 83, 84, 88

M

MAE - Mean Absolute Error 89

MAPE - Mean Absolute Percentage Error 63, 75, 77, 88

MDB-HDNN - Multivariate Disturbance-based Hybrid Deep Neural Network 19

MLP - Multi-Layer Perception 5, 35, 38, 39, 57, 59, 61, 75, 76, 88

MSE - Mean Squared Error 59, 62

N

NLP - Natural Language Processing 17

O

OD - Origin-Destination 48, 49

P

POI - Point of Interest 30, 54

R

R^2 - Coefficient of Determination 63, 77, 78, 88, 89

RBF - Radial Basis Function 26, 36

ReLU - Rectified Linear Unit 59, 61

RMSE - Root Mean Squared Error 62, 75, 77, 88

RNN - Recurrent Neural Network 16, 17, 41–43, 45

S

SARIMA - Seasonal Auto-Regressive Integrated Moving Average 7, 15, 40, 77, 78, 81, 83, 89

SARIMAX - Seasonal Autoregressive Integrated Moving Average with Exogenous Variables 7, 61, 82, 83

STGCN - Spatio-Temporal Graph Convolutional Network 16, 17

SVM - Support Vector Machines 15

SVR - Support Vector Regression 35, 37, 57, 58, 61, 75, 76

T

TGCN - Temporal Graph Convolutional Network 43

TRMF - Temporal Regularized Matrix Factorization 56, 57

V

Var - Explained Variance Score 63, 75, 77, 88

1. Introduction

1.1. Background

Public transportation systems have become an essential element of modern urban mobility, significantly contributing to environmental sustainability, reducing road congestion, and improving the quality of life in densely populated areas. Their importance continues growing as cities face increasing levels of urbanization, along with the associated challenges of pollution, traffic, and limited space for road expansions (Balcombe et al., 2004). Efficient public transportation networks enable citizens to access services, employment, and education while simultaneously alleviating the reliance on private vehicles, thus reducing carbon footprints and energy consumption (Mugion et al., 2018).

However, public transportation systems face operational challenges, particularly with regard to the crowdedness of stations. Crowdedness refers to the degree to which a space is occupied by people and can have a direct impact on service quality, including increased waiting times, reduced comfort levels, and potential safety concerns due to overcrowding (dell'Olio et al., 2011). Crowdedness at public transport stations is a key factor that influences passengers' perceptions of service quality. For instance, longer waiting times or overly crowded environments can lead to dissatisfaction, potentially driving passengers to opt for private vehicles, which counteracts efforts to reduce traffic congestion and promote sustainability (Millonig et al., 2012). In this context, analyzing and understanding crowdedness patterns becomes crucial, not only to improve the operational efficiency of public transportation systems but also to enhance the overall passenger experience.

Furthermore, addressing crowdedness is not merely about improving the passenger experience; it also impacts the long-term sustainability and attractiveness of public transportation systems. By improving service levels and reducing waiting times, public transit systems can become more competitive with private transport modes, thus encouraging more citizens to opt for sustainable transportation solutions. In turn, this helps cities achieve their goals related to reducing emissions and promoting eco-friendly urban mobility (Mugion et al., 2018). Analyzing crowdedness patterns in urban transit systems is critical for optimizing operational strategies and ensuring efficient public transportation management. This can be particularly useful during peak hours or special events, where proactive planning based on predictive models can help mitigate congestion and improve service delivery.

In the context of public transport stations, crowdedness at stations is particularly dynamic, driven by factors such as time of day (Vlahogianni et al., 2014), weather conditions (Pelletier et al., 2011b), and special events (Villiers et al., 2019). Large-scale gatherings, such as sports matches and concerts, can lead to significant surges in passenger numbers, resulting in operational challenges at transit hubs (Carvajal and Garcia-Colon, 2003; Goodwill and Joslin, 2006). Studies on crowdedness at public transport stations have explored how such factors affect passenger flow and system efficiency. For instance, research has examined how video data and crowd-counting algorithms can be used to estimate and forecast station congestion levels, helping transport authorities manage flows effectively (Thilakasiri et al., 2021). Other studies have used sensor-based and ticketing data to model crowdedness, providing insights into peak traffic times and helping to improve station operations (Niu et al., 2017).

In modern urban environments, crowding pattern analysis is particularly relevant within the context of ITS (Intelligent Transport Systems), which leverage real-time data and advanced analytics to optimize transit operations (Nuzzolo and Comi, 2016). Accurately predicting passenger flow and crowdedness at transit stations is a complex task, especially in dynamic urban settings where passenger numbers can fluctuate. Traditional forecasting models, such as ARIMA, are often used to predict traffic flows but may struggle with capturing nonlinear relationships and volatile fluctuations in urban transit systems (Li et al., 2017). Recent advances in machine learning and deep learning models, particularly GNNs, have shown promise in addressing these limitations by leveraging spatial and temporal dependencies within transit networks while more adaptive across diverse scenarios than traditional models (Wu et al., 2021). GNNs enable the modeling of interactions between stations, capturing the interdependencies of different network transit nodes and improving forecasting accuracy. With the capabilities of GNNs, we can better predict the crowdedness levels at individual stations and identify potential hotspots. Beyond merely predicting crowdedness at each station, understanding the shifts in crowdedness between stations is also crucial. Modeling the transfer of crowdedness across the network can provide deeper insights into the evolution of crowding patterns and support more effective management strategies and decision-making processes for urban transit systems.

In conclusion, accurately modeling and forecasting the evolution of crowding patterns in urban transit systems is essential for effective management and strategic planning. The integration of advanced GNN techniques and other mathematical techniques enables a comprehensive understanding of these patterns by capturing both spatial and temporal dependencies. Such an approach supports the implementation of dynamic operational strategies to alleviate congestion, particularly during peak hours or special events. By leveraging these techniques, public transportation authorities can optimize network performance, ensuring sustainable and efficient operations. Ultimately, enhancing the management of crowded public transit stations contributes to improved urban mobility and quality of life for city residents, aligning with the goals of modern cities striving for sustainable growth and better service quality.

1.2. Research Objectives

This thesis focuses on forecasting crowdedness patterns at public transport stations, paying particular attention to variations caused by special events, and studying crowdedness shifts within transit networks. The primary objectives of the research are as follows:

- To integrate various data sources to construct a robust pipeline for spatio-temporal dataset development tailored to this type of analysis.
- To propose and develop a robust forecasting model based on a GNN framework to accurately predict crowdedness patterns across an entire urban transit network, considering both spatial and temporal dependencies.
- To thoroughly investigate the impact of special event scenarios on crowdedness patterns in public transportation systems. This involves assessing how these events affect the performance of forecasting models and determining the specific challenges they pose to urban transit management.
- To analyze the temporal and spatial dynamics of crowdedness shifts within the public transport network, providing insights into how crowdedness propagates through the system.

1.3. Thesis Structure

This thesis is organized into several chapters, each addressing a distinct aspect of the research:

Chapter 2 provides a comprehensive literature review, discussing previous research on traffic forecasting and crowdedness pattern analysis in public transport systems. Particular emphasis is placed on the forecasting methods employed, ranging from traditional statistical models to advanced machine learning techniques, including GNN. Additionally, a separate section is dedicated to discussing the influence of special events on crowdedness and their impact on traffic prediction models.

Chapter 3 outlines the data development process, detailing the collection, preprocessing, and construction of the dataset used for the models. The dataset consists of three key components: spatial data representing the public transport network, station-related temporal data that reflects or can be processed to represent real-time crowdedness at the station, and special event data.

Chapter 4 describes the methodology used in building the forecasting models, including statistical models, time-series models, and the proposed GNN model. This chapter provides a detailed discussion of the model architectures and the specific approaches taken to ensure robust predictions.

Chapter 5 details the experimental setup, including the study areas, preprocessing steps applied to the data, model training configurations, and the evaluation metrics employed to assess the performance of the models.

Chapter 6 presents and discusses the results obtained from the models, offering an overview of the crowdedness patterns observed, the prediction results under different scenarios using various models, and a proposed method for analyzing crowdedness shifts within the transit network.

Chapter 7 concludes the thesis by summarizing its key contributions and offering insights into potential future research directions. Special emphasis is placed on furthering research in the area of crowdedness prediction in urban transit systems, particularly in the context of special event scenarios.

The thesis overview structure is shown in Figure 1.

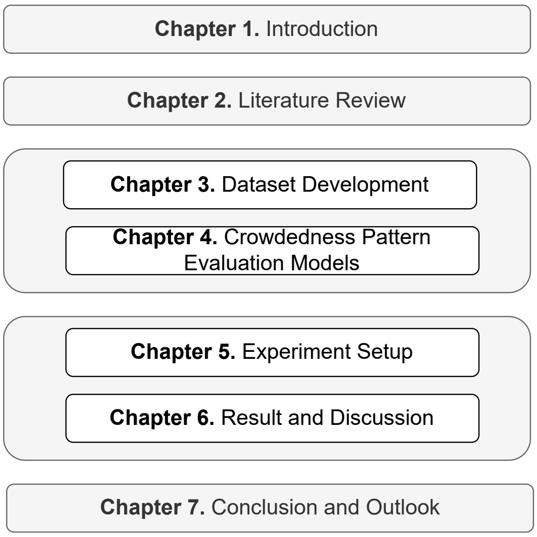

Chapter 1. Introduction
Chapter 2. Literature Review
Chapter 3. Dataset Development
Chapter 4. Crowdedness Pattern Evaluation Models
Chapter 5. Experiment Setup
Chapter 6. Result and Discussion
Chapter 7. Conclusion and Outlook

Figure 1 Structure of thesis

2. Literature Review

2.1. Traffic Forecasting

Traffic forecasting is an essential tool in urban planning and the efficient operation of transportation systems. As cities continue to urbanize, the growing demand for transportation systems, particularly public transport, has made the accurate prediction of traffic flows increasingly important (Balcombe et al., 2004). Traffic forecasting involves predicting not only road traffic but also flows in public transit, rail systems, and pedestrian networks. This variety in traffic types makes forecasting a challenging task, requiring models capable of handling both spatial and temporal dependencies. The dynamic nature of traffic, influenced by factors like time of day, special events, and weather conditions, has led to the evolution of more advanced traffic prediction methodologies (Ermagun and Levinson, 2018).

Traditional traffic forecasting models have predominantly focused on road traffic, with the main goal being congestion reduction and vehicle flow improvement (Vlahogianni et al., 2014). These models typically employ statistical methods, which use historical traffic data to make short-term predictions. However, public transportation has only recently become a significant focus in forecasting studies. Models addressing public transport aim to predict passenger flows, evaluate operational efficiency, and handle demand fluctuations due to events or seasonal changes (Vlahogianni et al., 2014). The inclusion of these variables adds another layer of complexity, necessitating a broader approach to traffic forecasting.

Accurate traffic predictions rely heavily on the diversity and quality of data sources. Traditionally, fixed-position sensors, such as inductive loops, magnetic sensors, and video processors, have been used in ITS to monitor vehicle counts and classify traffic at specific points. However, these sensors offer only localized insights and are spatially limited to particular intersections or road segments (Ermagun and Levinson, 2018). To address this limitation, mobile sensors, primarily using GPS-enabled devices in vehicles or smartphones, provide a more dynamic and widespread view of traffic patterns. These mobile sensors contribute to real-time predictions by tracking vehicle trajectories, speeds, and congestion levels (Jin et al., 2016). GPS-based data, coupled with vehicle-to-infrastructure communication systems, can significantly enhance the granularity of traffic forecasting models.

In public transport systems, smart card data from automatic fare collection systems offers detailed insights into passenger behaviors, such as boarding times, locations, and travel routes (Pelletier et al., 2011b). However, the availability of smart card data is not universal. In some cities, for instance, most cities in Germany, automatic fare collection systems are not yet widely implemented, making it difficult to obtain accurate travel patterns for research or operational purposes (Transperth, 2018). In cities where this data is available, it has proven highly effective for estimating ridership demand and evaluating operational efficiency, although the lack of alighting information often requires the use of trip-chaining algorithms to reconstruct complete journeys (Pelletier et al., 2011b).

Crowd-sourced data from platforms such as Waze and Moovit, which gather real-time traffic information from users, have recently emerged as significant contributors to traffic prediction models. These platforms allow for real-time traffic monitoring by relying on user-contributed data to update conditions dynamically, providing an additional layer of insights beyond traditional fixed sensors (Lau and Sabri Ismail, 2015). Furthermore, social media platforms like Twitter can provide real-time information on events, disruptions, or even public sentiment regarding traffic conditions, which can be integrated into forecasting models (Vlahogianni et al., 2014). Additionally, environmental data such as weather conditions, holidays, and special events need to be considered when predicting irregular traffic patterns (Ermagun and Levinson, 2018). The integration of these various data sources is critical for developing robust traffic forecasting models capable of adapting to both regular and irregular traffic conditions.

The methodologies used in traffic forecasting have evolved significantly, transitioning from simple statistical models to advanced machine learning and deep learning techniques. Early methods like ARIMA have been widely applied for short-term traffic prediction. However, ARIMA and its seasonal variant SARIMA are limited in handling non-linear relationships and spatial dependencies in traffic data (Ermagun and Levinson, 2018). To address this, non-parametric models such as KNN have been introduced. KNN is capable of handling non-linearities by comparing new traffic instances to similar historical cases. However, its computational expense limits its practicality for large-scale datasets (Vlahogianni et al., 2004).

Machine learning techniques, including SVM (Support Vector Machines) and decision trees, provide more flexibility in handling non-linear patterns. These methods have shown promise in capturing more complex relationships in traffic data, though they remain sensitive to hyperparameters and require careful tuning to generalize across various traffic conditions. Despite their advantages, these methods often struggle with large-scale urban traffic data due to their computational costs and sensitivity to parameter tuning (Ermagun and Levinson, 2018).

The advent of deep learning has marked a significant shift in traffic forecasting. Models like LSTM networks are particularly effective at capturing temporal dependencies, making them well-suited for forecasting dynamic traffic patterns (Ma et al., 2015). LSTM networks, by preserving long-term dependencies, outperform traditional time-series models in scenarios where traffic patterns exhibit significant temporal variations. More recently, GNN have emerged as a leading approach for capturing spatial dependencies in traffic data. GNN model traffic systems as graphs, with intersections, stations, or road segments represented as nodes and traffic flow as edges (Wu et al., 2021). Hybrid models that combine GNNs with recurrent neural networks, such as the DCRNN (Diffusion Convolutional Recurrent Neural Network), have demonstrated success in capturing both spatial and temporal dependencies, offering improved performance over traditional models (Li et al., 2018).

In summary, the combination of increasingly diverse data sources and advanced modeling techniques—ranging from traditional statistical models like ARIMA to more complex models like LSTM and GNN, has significantly improved the accuracy of traffic forecasting. However, challenges remain, particularly in integrating real-time data sources and improving model interpretability for practical applications in urban traffic management.

2.2. GNN for Traffic Forecasting

GNN have emerged as a powerful tool for traffic forecasting, particularly when dealing with graph-structured data, such as transportation networks. In these systems, traffic nodes, such as intersections or public transport stations, are modeled as graph nodes, and the flow of traffic between them is represented by edges. This graph-based approach is especially well-suited to capturing the complex spatial dependencies inherent in urban traffic networks (Ye et al., 2022). Unlike traditional neural networks, which struggle to represent non-Euclidean data, GNNs are specifically designed to operate on graphs, making them ideal for modeling transportation systems. Figure 2 provides a general overview of GNN architecture.

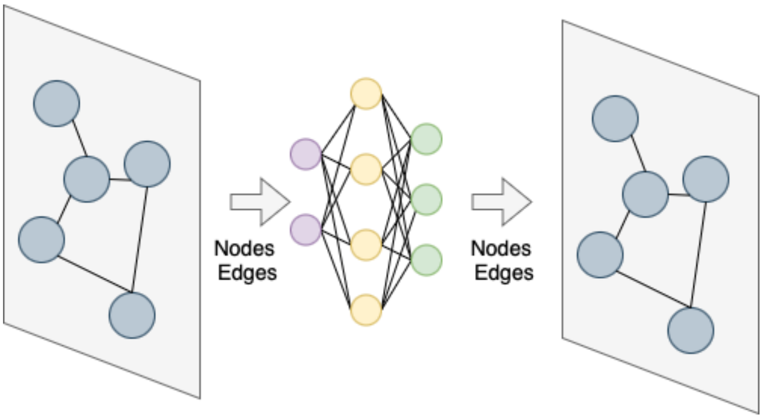


Figure 2 A genereal overview of GNN architecture

The earliest versions of GNNs were extensions of RNN (Recurrent Neural Network)s, specifically tailored to handle graph-structured data (Scarselli et al., 2009a). However, the introduction of convolutional GNNs marked a significant advancement, enabling more efficient learning from graph data through spectral and spatial convolutions. These approaches utilize either spectral methods, which operate in the Fourier domain, or spatial methods, which aggregate information from neighboring nodes, to learn effective node representations (Wu et al., 2021). This capability has made GNNs particularly useful in traffic forecasting, where both spatial and temporal dependencies must be modeled to capture the dynamic nature of traffic flows.

A notable example of a successful GNN-based model is the STGCN (Spatio-Temporal Graph

Convolutional Network) (Yu et al., 2018). STGCN combines GNNs with RNNs to model spatial dependencies between traffic nodes (e.g., stations, intersections) and temporal dependencies across different time intervals. By leveraging this hybrid approach, STGCN can capture both spatial relationships within the transportation network and temporal trends in traffic flow. These hybrid models have demonstrated remarkable success in improving the accuracy of traffic forecasting, especially in complex, dynamic urban environments. Li et al. (2018) introduced the Diffusion Convolutional Recurrent Neural Network (DCRNN), which extends this concept by employing a diffusion process to capture spatial dependencies while using recurrent neural networks to model temporal dependencies. DCRNN has shown significant improvements in handling irregular traffic flows and predicting traffic conditions in urban settings. Another recent example is the ADSTGCN (Adaptive Deeper Spatio-Temporal Graph Convolutional Network), which adapts both the graph structure and hidden layer connections dynamically to improve multi-step traffic forecasting, as demonstrated by Cui et al. (2023).

In recent years, attention mechanisms have been integrated into GNN architectures, further enhancing their predictive power. The attention mechanism is a technique that allows models to focus selectively on the most relevant parts of the input, dynamically adjusting the weight or importance given to each input feature. Originally inspired by human cognitive processes, attention mechanisms have been widely used in NLP (Natural Language Processing) and computer vision to improve model interpretability and performance by identifying the most salient parts of the data (Vaswani et al., 2017).

When applied to GNNs, attention mechanisms allow the model to dynamically prioritize certain nodes or edges in the graph based on their relevance to the task at hand. For traffic forecasting, this means that nodes or edges with higher levels of influence on overall traffic flow or congestion are given more weight in the model's predictions. For instance, during rush hour or special events, certain intersections or stations may become more critical, and the model can focus its attention on these areas, leading to more accurate predictions of traffic surges and congestion (Niu et al., 2021). Attention mechanisms thus help improve the ability of GNNs to make contextually aware predictions by weighting the relative importance of different nodes and edges (Zhang et al., 2022).

The use of attention mechanisms in GNNs has shown great promise, particularly in the context of dynamic traffic systems. These mechanisms allow models to adapt to sudden changes in traffic patterns, such as those caused by special events or accidents. By selectively focusing on the most relevant parts of the graph, attention-enhanced GNNs can improve their robustness and accuracy in handling real-time traffic forecasting scenarios (Chen and Liu, 2022). For example, Niu et al. (2021) highlights how attention mechanisms assign different weights to nodes and edges in transportation networks, allowing the model to adapt to fluctuating traffic conditions and make more refined predictions.

In conclusion, the integration of GNNs with attention mechanisms represents a significant leap forward in traffic forecasting. While GNNs are already well-suited for capturing spatial dependencies in traffic data, attention mechanisms enhance the model's ability to focus on critical nodes and edges, thereby improving both accuracy and interpretability. This combined approach is particularly useful for handling disruptions in traffic flow, such as those caused by special events, and is expected to play a vital role in the future development of real-time traffic forecasting systems.

2.3. Special Event for Traffic Forecasting

Special events, such as concerts, sports games, festivals, and public celebrations, introduce unique challenges for traffic forecasting due to the sudden and substantial influx of large crowds, which create irregular and complex patterns in transportation systems. These events can significantly disrupt regular traffic flows, leading to increased congestion, delays, and the need for rapid and effective traffic management strategies (Fernando, 2019). The complexity lies in the temporal and spatial concentration of attendees, which often leads to surges in transportation demand that cannot be captured by traditional forecasting models based solely on historical data (Villiers et al., 2019).

Special events can be categorized in several ways depending on their nature and scale. One common method is to distinguish between planned and unplanned events. Planned events, such as concerts, festivals, and sporting events, are typically scheduled well in advance, allowing authorities to anticipate and mitigate traffic impacts. In contrast, unplanned events, such as accidents or spontaneous gatherings, are unpredictable and thus more difficult to manage effectively (Villiers et al., 2019). Another useful categorization differentiates between short-term and long-term events. Short-term events typically last only a few hours or a day, while long-term events may span multiple days or even weeks, as seen with major festivals or conventions (Goodwill and Joslin, 2006).

Additionally, special events can be categorized based on their location and size. For instance, large-scale urban events like marathons or citywide festivals require different traffic management strategies compared to localized events, such as a sports game at a stadium (Noursalehi et al., 2018). Events held in central urban areas generally pose more significant traffic challenges than those in suburban or rural locations, as they interact with the already dense traffic patterns of city centers (Tempelmeier et al., 2020).

However, special events, particularly large-scale planned events, exert significant influence on traffic patterns. The influx of attendees creates a surge in transportation demand, often exceeding the capacity of the existing infrastructure (Fernando, 2019). Traffic congestion around event venues typically occurs in two waves: first, as attendees arrive at the venue, and second, as they leave. Both inbound and outbound traffic surges require efficient traffic management strategies to avoid severe delays and gridlock. This surge in demand poses unique challenges for traffic forecasting and management. Traditional traffic forecasting models, which typically rely on historical data and assume stable traffic patterns, struggle to capture the nonlinear and stochastic nature of traffic surges during special events. For example, models like ARIMA or simple regression approaches fail to capture the complex, nonlinear effects that large-scale events can impose on urban transport systems (Carvajal and Garcia-Colon, 2003). The limitations of these models have driven researchers to explore more advanced forecasting techniques that incorporate real-time data and external factors.

Recent advances in data collection technologies, such as GPS-enabled devices, smart cards, and social media platforms, have provided valuable real-time data that can be integrated into traffic forecasting models to capture the dynamic nature of traffic surges during special events. For instance, social media platforms such as Twitter and Facebook provide real-time updates on event attendance, road closures, and public sentiment, which can be leveraged to improve the accuracy of traffic predictions (Lau and Sabri Ismail, 2015; Zhang et al., 2022). In public transport systems, smart card data has become a vital source of information, offering detailed insights into passenger flows, boarding times, and locations, particularly during large-scale events. However, one of the main challenges in utilizing smart card data is its limited availability in certain regions or systems that do not use automated fare collection (Pelletier et al., 2011b).

Recent developments in deep learning have further improved the accuracy of traffic predictions for special events. For instance, Xue et al. (2022) proposed a MDB-HDNN (Multivariate Disturbance-based Hybrid Deep Neural Network) that has demonstrated superior performance in modeling the spatio-temporal dependencies inherent in traffic data during special events. Yu et al. (2018) generated a GNN-based model that is particularly adept at capturing the intricate relationships between traffic nodes, such as stations or intersections, and excels at dynamically adjusting predictions based on real-time updates from the surrounding event environment. These kinds of deep learning models are capable of incorporating real-time data and external factors such as weather, public sentiment, and event schedules, providing a more holistic view of traffic conditions. These models not only improve predictive accuracy but also help transit agencies design better traffic management strategies. By incorporating real-time data and external factors like weather and social media trends, these models provide a more comprehensive understanding of how special events will impact traffic, enabling authorities to make informed decisions about congestion mitigation measures (Goodwill and Joslin, 2006).

2.4. Literature Gaps

Despite the substantial progress made in traffic prediction and forecasting in recent years, several critical gaps remain in the literature that need further attention. This review has identified the following shortcomings in the existing body of work, specifically in relation to public transport networks, the use of diverse data types, and considerations of special event scenarios.

Firstly, there is a limited focus on public transport forecasting in the existing research on traffic prediction. In a comprehensive review of 146 traffic prediction studies utilizing GNN (Jiang and Luo, 2022), only a small subset of research, amounting to just 6% of the collected studies, concentrated on urban transit networks and public transportation systems. Specifically, out of 118 journal papers and 30 conference papers published in 2022, only 9 studies were dedicated to subway flow prediction, and most of these relied on data from AFC (Automatic Fare Collection) systems, such as smart card data. This reveals a significant imbalance in research priorities, with the majority of efforts concentrated on road traffic prediction. While road traffic studies are undeniably important, this disproportionate focus fails to address the growing need for accurate forecasting in public transportation, especially in rapidly urbanizing cities where efficient transit management is crucial.

Secondly, while there is a growing body of literature on crowdedness patterns and traffic forecasting models, many of these studies primarily rely on publicly available datasets, such as smart card data or automatic passenger counting data. For instance, studies such as (Chen et al., 2020) and (Li et al., 2017) leverage transit smart card data for special event forecasting. However, the availability of such data varies significantly across regions. For instance, in Germany, the adoption of AFC systems is not uniform across all cities and public transport lines. While cities like Bonn have implemented modern fare collection systems that rely on contactless payment technologies, other areas still lack comprehensive smart card systems (GmbH, 2020). This inconsistency in data availability presents a significant challenge, while few have explored how heuristic data, such as social media trends or ad-hoc survey data.

Furthermore, much of the existing work on both road traffic and public transport forecasting tends to focus on predictions at specific points or individual stations rather than considering the entire network or interrelationships between multiple nodes across the system. For instance, studies like (Noursalehi et al., 2018) explore station-level forecasting but do not fully integrate the broader network effects, particularly the dependencies between stations or nodes in complex urban transit systems. Network-based forecasting models, especially those incorporating graph structures, remain underexplored, limiting the models' capacity to capture intricate spatial and temporal dependencies.

Finally, the impact of special events on traffic forecasting has been widely overlooked in the literature. Special events, including planned events such as concerts, sports games, and festivals, as well as unplanned disruptions, can significantly alter traffic and passenger flow dynamics. Only a few studies, such as (Villiers et al., 2019) and (Kumar and Khani, 2021), explicitly focus on special event scenarios, and these often pertain to road traffic. The dearth of research addressing how special events impact public transportation systems, particularly in an urban transit network context, leaves a critical gap in forecasting models' ability to perform reliably in real-world situations.

In summary, the key gaps identified from the existing literature are as follows:

- Limited researches focus on public transport forecasting, with most studies prioritizing road traffic systems.
- A reliance on publicly available datasets without incorporating heuristic data, which could enhance forecasting accuracy, especially under dynamic and irregular conditions.
- Most forecasting models are point-based, lacking comprehensive network-level analyses that account for interrelationships between different nodes or stations.
- Few studies consider the impact of special events on traffic and public transport systems despite the frequent occurrence and potential significant disruption of such events.

These gaps highlight the need for further research to improve the robustness and applicability of traffic forecasting models, particularly in the context of public transport systems and the challenges posed by special events.

3. Dataset Development

In this section, we present the preparation of the dataset used in the models. In our research, to effectively model crowdedness at public transport stations under both regular conditions and special event scenarios, we utilize three key data subsets: special event data, public transport network data, and station-related temporal data. The following sections will detail the contents and collection methods of each subset.

3.1. Public Transport Network Spatial Data

Unlike many previous studies that focus solely on isolated stations, our research emphasizes the flow of passengers throughout the transport network and the interconnected influence between stations when predicting passenger flow. Hence, obtaining the entire public transport network as spatial data is necessary. There are two kinds of spatial data desired. One is network topology data, and the other is coordinates of stations. After defining the spatial scope of our study, we collected the corresponding urban rail transit network map. This data captures the structure of the public transport network, including the relationships between stations and transit lines. The network topology data will be utilized in various ways in subsequent stages of the research. For instance, it will be used to construct the adjacency matrix for GNN models or to analyze passenger flow transfers across different stations within the network. By incorporating the spatial structure of the network, our methodology goes beyond single-station predictions, enabling more comprehensive modeling of how passenger flow propagates through the entire transit system. This holistic approach allows a more accurate representation of real-world dynamics and enhances the model's ability to capture complex interactions within the public transport network.

With a list of stations in the research scope, we utilize the geocoding (Open Street Map, 2024) API of OpenStreetMap to obtain the latitude and longitude of each station to build the coordinate dataset. In later procedures, this dataset will be used to calculate the geometric distance between stations to construct features.

3.2. Special Event Data

Special events, such as concerts, sports events, and fairs, can lead to a significant increase in traffic demand, exerting substantial influence on public transportation usage and passenger flow. These events can create crowding patterns that differ from regular conditions, posing challenges for accurate passenger flow forecasting. Therefore, it is essential to collect data on special events occurring within the spatio-temporal scope of our study to construct a dedicated special event dataset. The further usage of this dataset enables the model to adapt to and capture these unique patterns, improving performance under special event scenarios. It will also be used to test the robustness of the models by evaluating their predictions during both regular and special event conditions.

The dataset includes several key components. For each event corresponding to one or more public transport stations, we collect the event's start and end times along with details of the associated stations. Table 1 outlines the data's attributes and descriptions. The definition of 'station(s) affected by the event' can vary based on different criteria. For instance, a specific distance, such as Euclidean distance, Manhattan Distance, or walking distance catchments, could be used to define the affected area. Additionally, official public transport information provided by the event organizers may serve as a more precise means of defining affected stations. Our research adopts the latter approach, relying on official transport information to delineate the relevant stations.

Attribute	Description
event	Name of the special event.
start_time	The start time of the event. In 'YYYYMMDD-HHMM' format.
end_time	The end time of the event. In 'YYYYMMDD-HHMM' format.
city	City where the event takes place.
line	Public transportation line(s) corresponding to the event location.
station	Public transportation station(s) affected by the event.

Table 1 Attributes of the special event dataset

3.3. Station-Related Temporal Data

Station-related temporal data refers to time-series data collected at public transport stations that capture dynamic changes in passenger flow or crowdedness over time. This data can be gathered through various sources, such as sensors, smart traffic card systems, or crowd-sourced data from mobile applications and social media platforms. Examples from prior research include AFC data from smart card systems, which provide detailed information on passenger entry and exit times, and data from GPS-enabled mobile apps that track passenger movement patterns. Heuristic data derived from social media check-ins and posts have also been utilized to estimate public transport usage during specific time intervals (Cheng et al., 2021a,b; Pelletier et al., 2011a).

This data is typically collected at regular intervals to create a continuous time series that represents the real-time or near-real-time crowdedness at stations. Once gathered, the raw data can be processed and transformed into meaningful features that characterize or relate to the crowdedness of the station. These features, in turn, are used to model and predict crowdedness at different stations under both regular and special event scenarios.

We have now completed the introduction of the dataset construction methodology. This method allows us to capture passenger flow characteristics in the public transportation system from both spatial and temporal dimensions, enabling the analysis of crowding patterns under various spatiotemporal conditions. This approach provides a viable framework for obtaining datasets in studies of this nature.

4. Crowdedness Pattern Evaluation Models

This chapter presents the methodology employed to evaluate the crowdedness patterns within the public transportation system. First, we provide an overview of the framework used to approach the evaluation and define the tasks. Following this, we introduce the models designed to analyze and predict crowdedness using the previously constructed dataset. A range of models, categorized into statistical and deep learning approaches, are implemented to achieve this objective. Notably, this research proposes a neural network model based on GNN, APT-GCN. Finally, the modeling of the crowdedness shift will be discussed.

4.1. Framework and Definition

The methodology is structured into two main components within the pipeline. Using the dataset collected and processed in previous stages, which captures spatial and temporal crowdedness of public transportation system dynamics, the first task involves forecasting crowdedness through various prediction models. The second component builds upon the outputs from the first, utilizing the attention weights from the APT-GCN model to measure the crowdedness shift along transit lines. Figure 3 shows the general framework of the methodology.

Following this, we define the problem for both subtasks. Table 2 provides a summary of the variables used in this chapter.

Variable	Description
	Basic definition variables
G	Graph to describe public transport network.
V	Set of stations, $V = \{v_1, v_2, \ldots, v_N\}$.
N	Station set size, number of stations.
E	Edge of graph represent the connection between stations.
A	Adjacency matrix.
$\tilde{A}$	$\tilde{A} = A + I$, representing adjacency matrix with seld-loop.
X_t	Attribute matrix. X_t denotes the observed data for all stations at time t.

Table 2 – *Conti.*

Variable	**Description**
M	Number of attributes.
y_t	Popularity at time t.
T	Data collection interval in munites.
s_{t+i}	Time seires. Time t is the start point, and the length of this vector is L.
L	Length of time series.
Prediction model related variables	
$f(\cdot)$	Prediction function mapping inputs data to popularity y.
p	The order of autoregression (AR) part in ARIMA model.
q	The order of moving average (MA) part in ARIMA model.
d	The order of differencing.
ϵ_t	The error at time t in regression models.
β	Coefficients for the feature X.
k	Number of nearest stations in the K-Nearest Neighbors model.
σ^2	Variance of the Gaussian Process model.
l	Length scale parameter in Gaussian Process with RBF (Radial Basis Function) kernel.
W	Weights for layers or units in neural networks.
b	Bias items.
$h^{(l)}$	Output of layer l in neural networks.
i_t	Input gate of LSTM unit.
f_t	Forget gate of LSTM unit.

Table 2 – *Conti.*

Variable	**Description**
o_t	Output gate of LSTM unit.
$\tilde{C}_t$	Candidate cell content at time step t.
C_t	Updated cell state at time t.
h_t	Hidden state at time t.
z_t	Update gate in GRU (Gated Recurrent Unit) unit at time t.
r_t	Reset gate in GRU unit at time t.
$\tilde{h}_t$	Candidate hidden state at time t.
$\tilde{D}$	Degree matrix, $\tilde{D} = \sum_j \tilde{A}_{ij}$.
$\sigma(\cdot)$	Activation function.
$\theta^{(l)}$	Parameter of layer $l + 1$.
p_v	Position embedding for node v.
$L(\cdot)$	Loss function of unsupervised embedding model.
Crowdedness shift modeling related variables	
e_{vu}	Attention coefficient.
α_{vu}^k	Normalized attention coefficient of k-th attention head.
u_t	Crowdedness shift feature.
P	Crowdedness shift matrix.
v_i	Weighted in-degree centrality.
v_o	Weighted out-degree centrality.
v_e	Weighted eigenvector centrality.
v_p	PageRank centrality.

Table 2 – *Conti.*

Variable	**Description**

Table 2 Summary of variable notations

Definition 1 Public Transport Network G. A public transport network is described as an unweighted, undirected graph $G = (V, E)$, representing the inherent typologies of public transport lines. Here, the graph's node set $V = \{v_1, v_2, \ldots, v_N\}$ represents the set of stations, where N is the number of stations. The set of edges E represents the connections between stations, with an edge existing between each pair of adjacent stations.

Definition 2 Adjacency Matrix $A^{N \times N}$. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ is a binary matrix that represents the connections between stations in the public transport network G. Each element a_{ij} in the matrix is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if there is an edge between station } v_i \text{ and station } v_j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

Definition 3 Attribute Matrix $X^{N \times M}$. The features of stations during observation time are represented by the feature matrix $X_t \in \mathbb{R}^{N \times M}$, where M denotes the number of attributes, which include the popularity y along with other features. At time t, attribute matrix $X_t \in \mathbb{R}^{N \times M}$ represents the status at each node.

Definition 4 Time Series. Time series $s_{t+i}^{L \times N \times M}$ is a vector consisting of a list of attribute matrices, where L is the length of time series. i is the number of the intervals. Thus the length of this vector L equals to $i + 1$. s_{t+L} can be denoted by:

$$s_{t+i} = [X_t, X_{t+1}, \ldots, X_{t+i}] \tag{4.2}$$

Defination 5 Multi-Step Popularity Forecasting. The first task is framed as a multi-step forecasting task that focuses on short-term traffic prediction. The goal is to forecast the popularity at future intervals, treating popularity as the target variable. The aim is to learn a function f that maps the public transport network G and a time series s_{t-i} of length $i+1$ to future n-step predictions of popularity y for each station. This forecasting model can be formally described as follows:

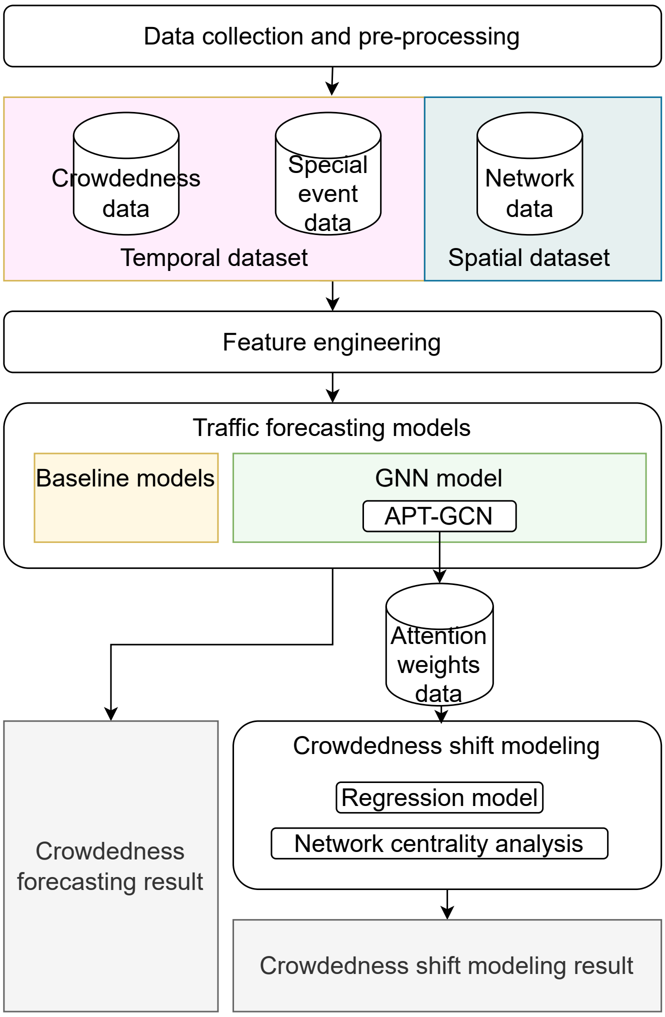

Data collection and pre-processing
Crowdedness data
Special event data
Temporal dataset
Network data
Spatial dataset
Feature engineering
Traffic forecasting models
Baseline models
GNN model
APT-GCN
Attention weights data
Crowdedness forecasting result
Crowdedness shift modeling
Regression model
Network centrality analysis
Crowdedness shift modeling result

Figure 3 General framework of forecasting crowding pattern evolution

$$(y_{t+1}, \ldots, y_{t+n\times T}) = f\left(G, s_{t-i}\right) \tag{4.3}$$

Where s_{t-i} represents the time series containing the attribute metrics in the past i interval, and y_t is the popularity at the timestamp t. When $n = 1$, this multi-step forecasting model degenerates into a single-step popularity forecasting model.

Defination 6 Crowdedness Shift Features $u_t \in \mathbb{R}^{N \cdot L}$. The crowdedness shift feature is designed to represent the contributions to the popularity at each station from all other stations in the network across different time steps. For a given station i, the features contain the influence of each other station $j \neq i$ in the network, along with the historical time-series data across L previous time steps.

4.2. Feature Engineering for Station-Related Temporal Data

In this section, we discuss the process of feature engineering. Beyond the current crowdedness index, which serves as the dependent variable, we have constructed eleven groups of features for the station-related temporal dataset. We begin by introducing the property of the station-related-temporal data, i.e., the crowdedness data. Following this, calculation methods for these features will be introduced. A comprehensive list of the generated features, along with their detailed descriptions, is provided in Table 4.

4.2.1. Crowdedness Data

To describe the level of crowdedness at specific transit stations during specific times, we utilize the term popularity as an indicator of station crowdedness. Regardless of the data source, the chosen crowdedness data should include the following attributes: current popularity, which directly represents the level of crowdedness at the station at a given time, and historical popularity, which provides an overview of typical busy periods at the station based on past observations. In our dataset, two additional attributes—visit duration and wait time estimation—are included to capture passenger behavior more comprehensively (Google, n.d.). However, these attributes are not mandatory for crowdedness estimation at this feature engineering step. All of these attributes are listed in Table 3 and serve as features for modeling or as raw data for constructing additional features in the analysis.

Attribute	**Description**
Current Popularity	Crowd level at a given POI at the present time.
Historical Popularity	Average popularity over the past several weeks, providing a value for each hour of each day of the week.
Visit Duration	Average time customers typically spend at a specific POI, estimated from visit patterns over recent weeks.
Wait Time Estimates	An estimate of the wait time a customer would experience before receiving service at different times of the day.

Table 3 Crowdedness data attributes

4.2.2. Special Event

This feature group includes the attribute ’event’, which serves as a special event label for the popularity data. This label is intended to capture the distinct passenger flow patterns associated with special events. For each identified special event, the start and end times are recorded, as detailed in Table 1. Utilizing this information, we introduce a new feature, ’event’, to reflect these unique temporal dynamics. The specific indicator is expressed in 4.4, designating a buffer time before the start and after the end of the event. The value of ’event’ for the corresponding station j will be given according to the distance from the special event in the time domain.

$$event_{ij} = \begin{cases} 10 - \left\lfloor \frac{|d|}{T} \right\rfloor & \text{if} \quad |d| \leq t_b + \frac{\Delta t_{event}}{2} \\ 0 & \text{Otherwise.} \end{cases} \tag{4.4}$$

In which $|d|$ denotes the minimum distance between the timestamp and the event start or end time, defined as:

$$|d| = \min(|t_{ij} - t_{start}|, |t_{ij} - t_{end}|) \tag{4.5}$$

Where t_{ij} is the timestamp i at station j, $event_{ij}$ is the event feature at t_{ij} for station j, t_{start} and t_{end} represent the start time and end time of event. d denotes the event duration in minutes while Δt_{event} denotes the duration of the event in minutes, t_b is the buffer time around the event in minutes, i.e., the time before and after the event where the feature still has non-zero values. T is the data collection interval in minutes. $\lfloor \cdot \rfloor$ is the floor function that rounds the value down to the nearest integer.

4.2.3. Station Similarity

This feature group includes statistical values of popularity from k-most similar stations in the same network. Specifically, two types of distances are employed to describe the similarity of stations: DTW (Dynamic Time Warping) distance to capture the similarity of time series and geometric distance to represent the physical distance in the real world. The DTW distance can be computed as,

$$d_{DTW}(s_{p+n}, s_{q+m}) = \min \left(\sqrt{\sum_{(i,j)\in\pi} (s_{p+n,i} - s_{q+m,j})^2} \right) \tag{4.6}$$

Where $d_{DTW}(s_{p+n}, s_{q+m})$ denotes the DTW distance between two time series starting at indices p and q, with length $n+1$ and $m+1$ respectively, representing the similarity of crowdedness pattern. π is a warping path.

In this study, we calculate the geometric distance between stations as follows:

$$d_{geo}(i,j) = \sqrt{(lat_i - lat_j)^2 + (lon_i - lon_j)^2} \tag{4.7}$$

where $d_{geo}(i,j)$ represents the geometric distance between two station i and j. lat_i, $latj$ are the latitudes while lon_i, lon_j are the longitudes from spatial dataset 3.1.

In total, four features are derived from this group: the average value and standard deviation of popularity from the k-nearest stations based on both DTW distance and geometric distance.

4.2.4. Station Clustering

Finally, to further capture station features with similar crowdedness patterns, we apply K-means clustering to the stations within the same network based on the obtained features. K-means is an unsupervised clustering algorithm that aims to minimize intra-cluster variance to form groups. The algorithm requires specifying the number of clusters. Given n clusters, the K-means algorithm seeks to choose centroids μ_j that minimize the following objective:

$$J = \sum_{j=1}^{n} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2 \tag{4.8}$$

where J represents the sum of squared distances between each data point x_i and the centroid μ_j of the cluster C_j, and n is the number of clusters.

Since the optimal number of clusters is not predetermined in our case, we utilize silhouette analysis (Rousseeuw, 1987) to determine the best number of K-means clusters. By calculating and maximizing the silhouette scores, this method ensures that each point in a cluster is as far away as possible from neighboring clusters while being close to its own cluster's centroid. The silhouette score is given by:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4.9}$$

Where $s(i)$ is the silhouette score for data point i, $a(i)$ is the average intra-cluster distance (i.e., the average distance between i and all other points in the same cluster), and $b(i)$ is the average nearest-cluster distance (i.e., the average distance between i and points in the nearest neighboring cluster).

Through the clustering algorithm, we obtain classification information for each station. At each time stamp, we calculate the mean and standard deviation of the popularity values for stations in the same cluster, which serve as the final two features for stations within the cluster. With this, the construction of station-related temporal data is complete.

Category	**Feature**	**Description**
Dependency value	y	Current popularity.
Time related feature	week_year	Week number of the year.
	day_week	Day number of the week, Monday = 1, Tuesday = 2, and so on.
	no_interval	Number of intervals in a day.
	interval_weekly	Number of intervals in a week.
	interval_of_year	Unique interval of the year.
Special event	event	Event index, equals 1 if during an event, otherwise 0.
Rating	rating	Overall user rating or satisfaction level for the station.
	rating_n	Number of reviews contributing to the overall rating.
Station type	station_type_X	Dummy features to define the type of subway station. Equals 1 if the station belongs to type X, otherwise 0. X = {regional railway, subway, tram}.
Network scale	network_index	Number of rail transit lines in the current city.

Table 4 – *Conti.*

Category	Feature	Description
City	city_X	Dummy variable for city X, equals 1 if the station is in city X, otherwise 0. X $\in \{Set\ of\ cities\}$.
Holiday	holiday	Holiday label, equals 1 during a national holiday, otherwise 0.
Weather	temp	Temperature.
	rhum	Humidity.
	prcp	Precipitation.
	wspd	Wind speed.
Historical popularity	h_p	Popularity at this time on this day in history.
	ha_p	Popularity at this time on the previous day in history.
	hb_p	Popularity at this time on the next day in history.
	h_w	Waiting time data (in minutes) for this time on this day in history.
	ha_w	Waiting time data (in minutes) for this time on the previous day in history.
	hb_w	Waiting time data (in minutes) for this time on the next day in history.
	y_1, y_2, y_3	Popularity from the last three intervals.
Nearest station	knn_y_i_mean, knn_y_i_std	Average and standard deviation of popularity from the k-nearest stations using DTW distance ($i \in (1, k)$).

Table 4 – *Conti.*

Category	**Feature**	**Description**
Nearest station	geo_knn_y_i_mean, geo_knn_y_i_std	Average and standard deviation of popularity from the k-nearest stations using geometric distance ($i \in (1, k)$).
Station cluster	cluster_mean, cluster_std	Average and standard deviation of popularity data from stations in the same cluster.

Table 4 Station feature data description

4.3. Statistical Models

Statistical models refer to techniques that assume specific probabilistic structures or patterns within the data to make predictions. Below, we introduce both traditional regression models and machine learning models used in this research, including Linear Regression, GPR (Gaussian Process Regression), KNN, GBR (Gradient Boosting Regression), SVR (Support Vector Regression), and MLP Model.

4.3.1. Linear Regression

Linear regression is one of the most basic and widely applied regression models, especially useful for straightforward datasets. Given the attribute matrix X_t, we can directly apply linear regression to predict the popularity at time t. This method is simple but efficient and effective, Although it may struggle to capture complex relationships beyond the provided features. The linear regression model is given as:

$$y_t = \beta_0 + \sum_{j=1}^{M} \beta_j X_{t,j} + \epsilon_t \tag{4.10}$$

Where:

- y_t is the predicted popularity at time t.
- β_0 is the intercept.
- β_j are the coefficients for the features $X_{t,j}$, where j is the index for the feature in the

attribute matrix.

- ϵ_t is the error term.

4.3.2. Gaussian Process Regression

Unlike linear regression, a GPR offers a probabilistic approach, allowing the prediction to be accompanied by empirical confidence intervals. Furthermore, it is flexible with respect to different kernels as well. Using the RBF kernel as an example, a GPR can be formulated as:

$$y_t \sim \mathcal{GP}(m(X_t), k(X_t, X_{t'})) \tag{4.11}$$

$$k(X_t, X_{t'}) = \exp\left(-\frac{\|X_t - X_{t'}\|^2}{2\ell^2}\right) \tag{4.12}$$

Where:

- $m(X_t)$ is the mean function of the process, often assumed to be zero.
- $k(X_t, X_{t'})$ is the covariance function (kernel) that defines the relationship between the features of different stations.
- ℓ is the length scale parameter that controls the smoothness of the predictions.

4.3.3. K-Nearest Neighbors

KNN assumes that stations with similar attributes will have similar popularity predictions. The predicted popularity at a given station v_i can be formulated as the average popularity of its k-nearest stations:

$$y_{t,i} = \frac{1}{k}\sum_{j=1}^{k} y_{t,j} \tag{4.13}$$

Where:

- $y_{t,i}$ is the predicted popularity for station v_i.
- $y_{t,j}$ is the observed popularity at the j-th nearest station.
- k is the number of nearest neighbors considered.

4.3.4. Gradient Boosting Regression

GBR is an ensemble learning method that builds models sequentially, where each model corrects the errors of the previous one. The GBR model can be expressed as:

$$y_t = \sum_{m=1}^{M} \alpha_m h_m(X_t) \tag{4.14}$$

Where:

- y_t is the predicted popularity.
- $h_m(X_t)$ is the m-th weak learner, often a decision tree.
- α_m are the weights assigned to each weak learner.

4.3.5. Support Vector Regression

SVR aims to identify the most significant features for prediction by solving an optimization problem. Given the feature matrix $X \in \mathbb{R}^{N\times(M-1)}$, SVR solves the following primal problem:

$$\min_{w,\epsilon} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \epsilon_i \tag{4.15}$$

$$subject\ to \quad y_i(w^T X_i + b) \geq 1 - \epsilon_i,\ \epsilon_i \geq 0 \tag{4.16}$$

Where:

- w are the model weights.
- ϵ_i are the slack variables for errors.
- C is the penalty parameter controlling the trade-off between margin size and misclassification error.

The dual problem is given by:

$$\min_{\alpha} \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j k(X_i, X_j) - \sum_{i=1}^{N} \alpha_i y_i \tag{4.17}$$

$$subject\ to \quad \sum_{i=1}^{N} \alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C \tag{4.18}$$

Where α_i are the Lagrange multipliers, and $k(X_i, X_j)$ is the kernel function. Finally, the prediction is made as:

$$y_t = \sum_{i=1}^{N} \alpha_i k(X_t, X_i) + b \tag{4.19}$$

Where b is the bias term, and $k(X_t, X_i)$ is the kernel function that measures the similarity between station features.

4.3.6. Multi-Layer Perception

The MLP is a widely adopted neural network architecture, particularly effective for regression tasks. Unlike traditional linear models, MLP is capable of capturing complex, non-linear relationships between the input features and the target variable. As shown in Figure 4, the MLP consists of an input layer, one or more hidden layers, and an output layer, all of which are connected through weighted directed connections. Once the inputs pass through the network, they traverse these directed connections between layers and eventually reach the output layer. Thus, the MLP operates as a forward network, also known as a feedforward neural network.

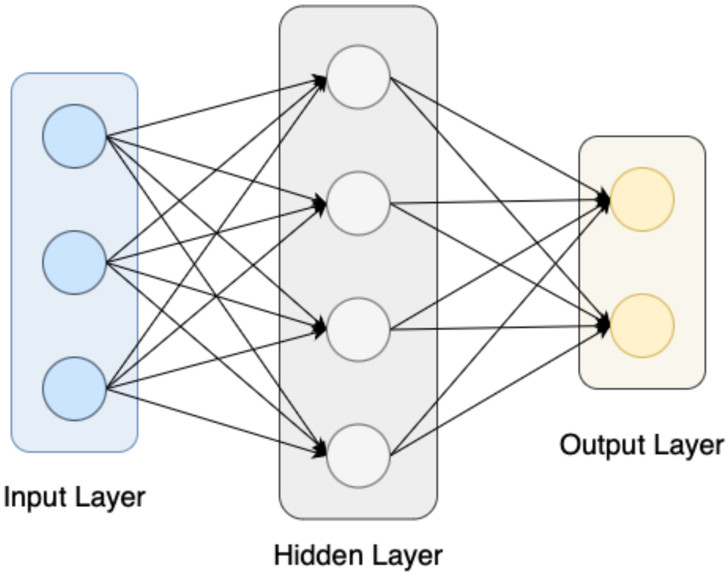


Figure 4 An example of MLP network

The computation at each layer and the final prediction can be expressed mathematically as follows:

$$h^{(l)} = \sigma\left(W^{(l)} h^{(l-1)} + b^{(l)}\right), \quad l = 1, 2, \ldots, L-1 \tag{4.20}$$

$$\hat{y} = \sigma_o\left(W^{(L)} h^{(L-1)} + b^{(L)}\right) \tag{4.21}$$

Where $h^{(l)}$ represents the output of the l-th layer (hidden or output layer), $W^{(l)}$ and $b^{(l)}$ are the weight matrix and bias vector at the l-th layer, $\sigma(\cdot)$ is the activation function applied at each hidden layer, such as ReLU or sigmoid, $\sigma_o(\cdot)$ is the activation function applied at the output layer, typically a softmax function for classification or a linear function for regression, L is the total number of layers.

During training, the weights are updated through a learning process. The most well-known learning algorithm for MLPs is backpropagation, where the error is propagated backward from the output layer to the input layers to adjust the weights. This is mathematically expressed as:

$$\Delta W^{(l)} = -\eta \frac{\partial \mathcal{L}}{\partial W^{(l)}}, \quad l = L, L-1, \ldots, 1 \tag{4.22}$$

Where: η is the learning rate, controlling the step size for weight updates, $\mathcal{L}$ is the loss function, typically cross-entropy for classification or mean squared error for regression, $\frac{\partial \mathcal{L}}{\partial W^{(l)}}$ is the gradient of the loss function with respect to the weights of the l-th layer.

Unlike traditional regression or statistical models, the performance of deep learning models like MLPs depends not only on the data and the choice of structural parameters, such as the number of hidden layers and nodes but also on training parameters like the learning rate and number of iterations. These factors significantly influence how well the model generalizes to unseen data.

4.4. Time Series Models

In this section, we will discuss two time series models: ARIMA and LSTM. Unlike the regression models mentioned previously, time series models do not treat the attribute matrix X_t at a specific time t for a station as an independent input. Instead, these models consider the impact of the temporal ordering and position of these values within the sequence. By utilizing time series s_{t+i} as input, time series models can better capture the evolution of patterns over time, which is crucial for making accurate predictions in dynamic systems such as urban transport networks.

The main advantage of time series models lies in their ability to capture both short-term fluctuations and long-term trends in data. For example, ARIMA models are designed to handle linear relationships in time series data, leveraging auto-regression and moving averages to predict future values based on past observations (Box et al., 2015). In contrast, LSTM models are particularly well-suited for modeling non-linear temporal relationships and longer-term dependencies due to their ability to retain memory over long sequences of data (Hochreiter and Schmidhuber, 1997). This memory capability makes LSTM models ideal for capturing the complex, multi-step temporal dynamics often observed in real-world transport systems (Ma et al., 2015).

4.4.1. ARIMA

The ARIMA model is a classic time series forecasting model that has been widely applied in transportation prediction, with numerous variations such as SARIMA. The fundamental idea behind ARIMA is that future values generally follow long-term historical trends but fluctuate around these trends due to short-term random events. An ARIMA model is composed of AR and MA components and can be represented as:

$$\text{ARIMA}(p, d, q) : \phi_p(B)(1-B)^d y_t = \theta_q(B)\epsilon_t \tag{4.23}$$

Where:

- y_t is the predicted popularity (the target variable).
- c is a constant.
- $\phi_p(B)$ is the autoregressive (AR) polynomial of order p.
- $\theta_q(B)$ is the moving average (MA) polynomial of order q.
- d is the order of differencing.
- ϵ_t is the error term at time t.

Here, p and q are the orders of the AR and MA models, determining the abstraction level of information extraction. A higher order leads to the loss of some of the original information but allows the model to better focus on underlying trends and patterns in the data.

Additionally, as a variant of ARIMA, the SARIMA model accounts for seasonal variations in the data by adding a seasonal component. This seasonal term helps mitigate the effects of periodic fluctuations that are characteristic of time series data, such as daily, weekly, or yearly cycles. The general form of the SARIMA model is expressed as:

$$\text{SARIMA}(p, d, q)(P, D, Q, s) : \phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D y_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t \tag{4.24}$$

Where:

- $\phi_p(B)$ is the autoregressive (AR) polynomial of order p.
- $\theta_q(B)$ is the moving average (MA) polynomial of order q.
- $\Phi_P(B^s)$ is the seasonal AR polynomial of order P.
- $\Theta_Q(B^s)$ is the seasonal MA polynomial of order Q.
- $(1 - B)^d$ is the non-seasonal differencing term.
- $(1 - B^s)^D$ is the seasonal differencing term.
- s represents the length of the seasonal cycle.
- ϵ_t is the white noise error term at time t.
- B is the backshift operator such that $By_t = y_{t-1}$.

ARIMAX (Autoregressive Integrated Moving Average with Exogenous Variables) is another extension of ARIMA model. In this model, exogenous regressors are incorporated to provide more information from the external factors when making predictions. This model could be specified as:

$$\text{ARIMAX}(p, d, q) : \phi_p(B)(1 - B)^d y_t = \theta_q(B)\epsilon_t + \sum_{k=1}^{K} \beta_k x_{t-k} \tag{4.25}$$

Where:

- y_t is the predicted popularity (the target variable).
- $\phi_p(B)$ is the autoregressive (AR) polynomial of order p.
- $\theta_q(B)$ is the moving average (MA) polynomial of order q.
- d is the order of differencing.
- ϵ_t is the error term at time t.
- x_{t-k} represents the exogenous variables at lag k.
- β_k are the coefficients corresponding to the exogenous variables.

4.4.2. LSTM

LSTM is a variant of RNNs. Unlike Feedforward Neural Networks (FNNs), which process inputs in a fixed-length manner and lack memory of prior inputs, RNNs are designed to handle sequence data and capture dependencies across time. This makes RNNs particularly effective in tasks involving sequential data, which is favorable to our time series forecasting problem.

RNNs have been extensively studied due to their ability to model temporal sequences. However, despite their potential, traditional RNNs suffer from certain limitations, particularly the problem of vanishing gradients, which makes it difficult for them to capture long-term dependencies over extended sequences. This issue often results in poor performance when modeling long-range temporal dependencies.

To address these limitations, Hochreiter and Schmidhuber (1997) introduced the LSTM network, which modifies the RNN architecture by adding a more sophisticated gating mechanism. LSTMs incorporate input, forget, and output gates, enabling the network to control the flow of information more effectively and maintain long-term dependencies. Figure 5 illustrates the structure of an LSTM unit.

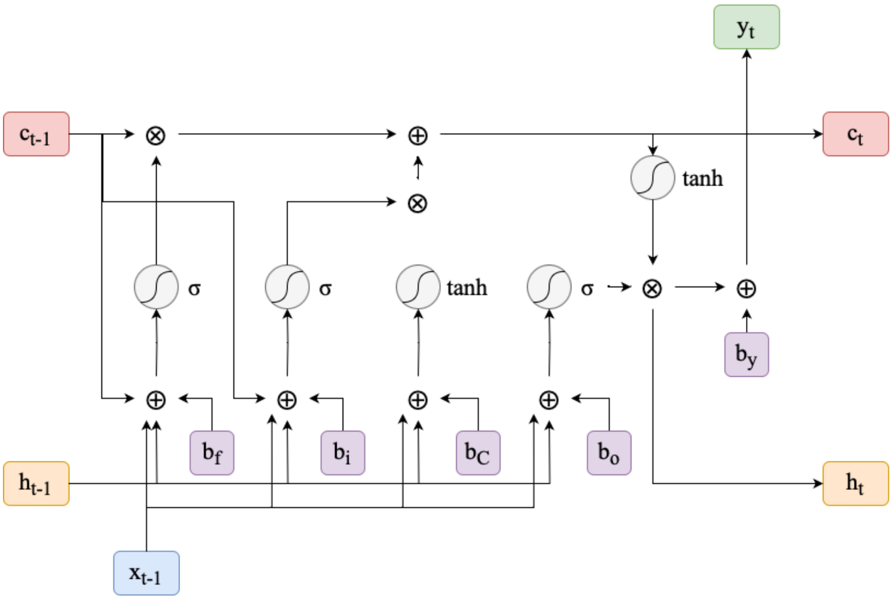


Figure 5 The unit structure of LSTM

The expressions governing the operations within an LSTM cell are given as follows:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4.26}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{4.27}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{4.28}$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{4.29}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4.30}$$

$$h_t = o_t \odot \tanh(C_t) \tag{4.31}$$

Where i_t is the input gate, controlling how much new information is written to the cell state, f_t is the forget gate, determining how much of the previous cell state C_{t-1} is retained, o_t

is the output gate, deciding the amount of cell state to pass through to the hidden state. The candidate cell state $\tilde{C}_t$ represents the new cell content at time step t, while C_t is the updated cell state at time step t. h_t is the hidden state at time step t and W_i, W_f, W_o, W_C are the respective weight matrices, and b_i, b_f, b_o, b_C are bias vectors. $\odot$ is the element-wise product.

The LSTM network's architecture overcomes the vanishing gradient problem faced by traditional RNNs (Cho et al., 2014b), allowing it to capture long-term dependencies in sequence data.

Unlike the GRU, which has a simpler structure with only update and reset gates, the LSTM uses three gates—input, forget, and output—along with a cell state that is updated iteratively. This allows LSTM to balance how much information from previous time steps is remembered or forgotten and how much new information is introduced at each step. Specifically, the reset gate R_t in GRU allows the model to forget parts of the past selectively, while LSTMs achieve finer control with their three gates and separate cell state.

4.5. APT-GCN Model

4.5.1. Overview

In this study, we proposed a spatial-temporal model with a graph attention module enhanced by a position embedding mechanism (APT-GCN). Fig 6 illustrates the structure of the APT-GCN model. In the first part, the method to capture temporal-spatial dependencies is based on the TGCN (Temporal Graph Convolutional Network) proposed by (Zhao et al., 2020). It begins by taking time series data s for each node, and the adjacency matrix A as input. A GCN (Graph Convolution Network) is then used to capture spatial features from this data. The output of the GCN is subsequently fed into a GRU, which allows for the temporal flow of information across different time snapshots. Following the T-GCN module, the output is further processed by an attention mechanism designed to integrate historical information from non-adjacent stations. At the same time, the attention mechanism evaluates the contribution of other stations at time t to our secondary objective: the modeling of crowdedness shift. This is accomplished through position embedding, which enhances the attention module's ability to incorporate information from distant nodes. Finally, the results from the attention module are aggregated along the temporal dimension to produce predictions. This model effectively combines short-term historical data for each node with information from other nodes in the graph, enabling more accurate predictions of future popularity trends.

4.5.2. Spatio-Temporal Module

Gathering neighbors' influence via graph convolution layer

GCNs are a type of semi-supervised model introduced by Kipf and Welling in 2016 (Scarselli et al., 2009b). GCNs enhance traditional CNN (Convolutional Neural Network)s within the domain of GNN. While CNNs can capture local spatial features in Euclidean space, such

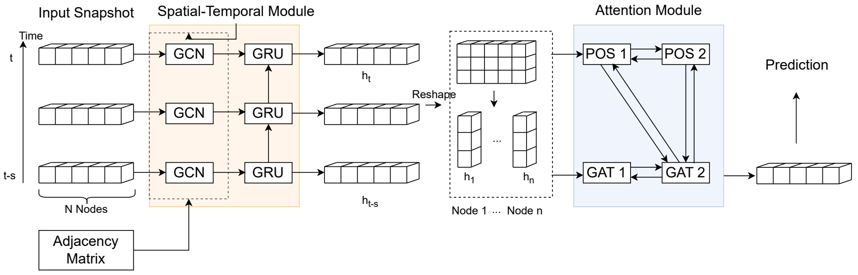


Figure 6 An overview of the APT-GCN network

as those in images, they fall short in domains like transportation, where spatial dependencies arise from network topological structures. GCNs address this limitation by performing convolution operations on non-Euclidean data.

The fundamental idea of GCNs is to aggregate features from neighboring nodes and then transform these features. By stacking k layers, a GCN can capture the features of k-order neighbors. Given the adjacency matrix A, which represents non-Euclidean graph data, a GCN typically performs two operations: propagation and transformation. Propagation involves using filters in the Fourier space to capture and aggregate features from first-order neighbors. The aggregated spatial information is then transformed between layers through linear transformations or activation functions. Given the feature matrix X and the matrix $\tilde{A}$ representing the network structure, the output of GCN layer $l+1$ is computed as follows:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}\theta^{(l)}\right) \tag{4.32}$$

where $H^{(l)}$ is the output of layer l, $\tilde{A} = A + I$ denotes adjacency matrix adding self-loop, $I \in \mathbb{R}^{N\times N}$ is an identity matrix. $\tilde{D}$ is the degree matrix computed by $\tilde{D} = \sum_j \tilde{A}_{ij}$. $\theta^{(l)}$ denote the parameter of layer $(l+1)$. $\sigma(\cdot)$ represents the activation function. $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is a pre-process of adjacency matrix.

Thus, a graph convolutional network can be defined as:

$$H^{(l)} = g(H, \tilde{A}) = \sigma(\tilde{A}H^{(l-1)}W^{(l)}) \tag{4.33}$$

$$Z = g^{(n)}(X, \tilde{A}) = \underbrace{g\left(g\left(\ldots g(X, \hat{A})\ldots\right)\right)}_{n \text{ layers}} \tag{4.34}$$

Where n is the number of layers to capture spatial dependencies from n-order neighbors. In our model, n is set to 2. X is input feature matrix and $Z \in \mathbb{R}^{N\times C}$ is the final output of GCN network.

Gated recurrent units for temporal domain

When addressing transportation issues—whether they pertain to road traffic, rail transit, or a combination of public transportation types—participants invariably move through a specific network along the time dimension. In the short term, historical data significantly influences the prediction of traffic related variables at a given moment. This relationship has been highlighted in various studies, which emphasize the importance of incorporating historical data to enhance the accuracy of traffic forecasts (Żochowska and Pamuła, 2024). Therefore, one of the primary objectives in traffic forecasting is to capture temporal dependencies effectively. This goal can be achieved using RNNs, which are designed to handle sequence data and capture dependencies across time. RNNs have been extensively studied for their ability to model temporal sequences and their applications in time-series forecasting.

RNNs come in several variants, with Gated Recurrent Units (GRUs) and LSTM units previously mentioned in (4.4.2) being among the most notable. Introduced by Cho et al. (2014a), GRUs simplify this structure by combining the input and forget gates into a single update gate, making them computationally less demanding (Cho et al., 2014a). In the context of our research, The GRU cell processes the output from the graph convolutional network by:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \tag{4.35}$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \tag{4.36}$$

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h) \tag{4.37}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{4.38}$$

Where z_t is the update gate controlling how much of the past hidden state h_{t-1} is retained in the current hidden state. r_t is the reset gate, which determines how much of the past hidden state is ignored for the candidate hidden state $\tilde{h}_t$. $\tilde{h}_t$ is the candidate hidden state, computed based on the current input x_t and the previous hidden state h_{t-1}, modulated by the reset gate r_t. h_t denotes the final hidden state at time step t, as a combination of the previous hidden state and the candidate hidden state, weighted by the update gate. W_z, W_r, W_h are the weight matrices for the update, reset, and candidate hidden state, respectively, while b_z, b_r, b_h are the corresponding bias vectors. $\odot$ represents element-wise multiplication. In GRU, the activation function σ is the sigmoid activation function, which outputs values between 0 and 1, and $\tanh$ is the hyperbolic tangent activation function, which outputs values between -1 and 1.

While both models utilize gating mechanisms to control the flow of information, LSTMs generally involve more complex operations, which can lead to longer training times (Chung et al., 2014). In our study, we evaluated the performance of GRUs and LSTMs on sample data, as well as the combination of these units with single-layer models. Experimental results indicated that while LSTM and the combined models did not significantly improve accuracy compared to GRU under the same conditions, they considerably increased the training time. The GRU layer delivered comparable accuracy to LSTM, with only a marginal improvement observed when using LSTM. However, this slight accuracy gain came at the cost of a notable increase in computational time. Therefore, considering the trade-offs between accuracy and computational efficiency, we ultimately chose GRUs as the recurrent units to capture temporal information in our model.

4.5.3. Graph Attention Module

As a subset of general traffic prediction, crowdedness forecasting in public transport systems inherits several characteristics from traffic forecasting while presenting distinct challenges. At each station, crowdedness is influenced by various factors, including historical passenger flow, the schedule of arriving and departing trains, inter-station transfers, and new demands, which all contribute to the dynamic passenger load at any given moment. Public transport operates on fixed service schedules, leading to periodic contributions from other stations in the network to the crowdedness at any specific station. Capturing these inter-station relationships in forecasting is crucial for improving predictive accuracy, especially during special events where sudden surges in passenger volume can impact multiple subsequent time steps and nodes within the network (Caroleo et al., 2024). Additionally, modeling the shift in crowdedness of these connections, which is another objective of our research, provides valuable insights into the operation and management of public transport.

The introduction of the graph attention module further enhances the model's efficiency in addressing these challenges. In previous modules, GCNs were employed to capture spatial features across the network. While GCN layers can access information from n-order neighbors by stacking multiple layers, this approach becomes inefficient in rail public transport scenarios where passenger flow transfers rapidly. For instance, the total trip duration on Munich's U-Bahn Line 6, which comprises 27 stations, is approximately 51 minutes (MVV open data, 2024). This implies that, within a single time interval in the data, passengers could transfer to most of the stations along the line, considering the data collection interval is relatively long (e.g., 30 minutes), which is often the case given the heuristic data collection methods and associated resource constraints. To capture the features of distant neighboring stations, an excessive number of GCN layers would need to be stacked, resulting in an inefficient and computationally expensive model.

To overcome these limitations and achieve our goals, we introduce a graph attention module to capture non-local dependencies effectively. The graph attention mechanism allows the model to weigh the importance of each node differently, thus providing a more nuanced representation of the network. This module is further enhanced with a position embedding mechanism proposed by Ma et al. (2021), which helps in integrating historical information from non-topologically adjacent stations into the current node's prediction. This mechanism is particularly effective for non-homophilic graphs, enabling us to capture features from stations that are not geographically or topologically proximate but share similar characteristics. This significantly enhances the accuracy of popularity forecasting, especially under special event conditions.

Our model comprises two graph attention layers and two positional embedding layers that interact with each other, enhancing the model's ability to capture complex spatial relationships within the data. Taking the output from the last section and reshaping it into vectors of node features, the attention coefficients are computed by

$$e_{vu}^{k} = a(W_k x_v + U_k p_v \parallel W_k x_u + U_k p_u) \tag{4.39}$$

where x_v, x_u is the vectors of node features, W_k, U_k and a_k are the weights in the k-th attention head. $\|$ is the concatenation operation. p_v is the positional embedding for node v computed by,

$$p_v^l = \sigma(W_{emb}^l p_v^{l-1}) \tag{4.40}$$

$$\mathbf{L}(p_{v v\in N}, G) = \sum_{v\in N}\sum_{u\in\mathcal{N}(v)} (-log\sigma(p_v^T p_u) - Q \cdot \mathbb{E}_{u' \ P_n(v)} log(\sigma(-p_v^T p_{u'}))) \tag{4.41}$$

where l is the l-th position embeddings layer, and $\mathbf{W}_{emb}^l$ is the learned weight matrix of position embedding layer l. Equation 4.41 describes the loss function of the unsupervised embedding mechanism layers.

The attention coefficients $\mathbf{e}_{vu}$ are normalized by softmax function:

$$\alpha_{vu}^{k} = softmax_u(e_{vu}) = \frac{exp(e_{vu})}{\sum_{j\in\mathbb{N}_v} exp(e_{vj})} \tag{4.42}$$

Then, a linear transform is applied to gain the output $x_v^{'}$ by combining neighbors with normalized coefficients,

$$\mathbf{x}_v^{'} = \sigma(\frac{1}{K}\sum_{k=1}^{K}\sum_{u\in\mathbb{N}_v} \alpha_{vu}^{k} W^k \mathbf{x}_u) \tag{4.43}$$

In conclusion, the APT-GCN model effectively captures the spatial-temporal dependencies in public traffic networks and efficiently predicts future popularity. The first module captures the spatial information of topologically adjacent nodes on the public transit lines, as well as the historical information of the time series. The second module utilizes a graph attention network with position embedding mechanisms to obtain information on semantically adjacent stations in crowdedness patterns while providing the attention coefficients for further crowdedness shift modeling.

4.6. Crowdedness Shift Modeling

4.6.1. Regression Model for Crowdedness Shift Matrix

For public transport systems without an automatic fare collection system that lacks smart card data, traditional methods such as sample data for calculating the OD (Origin-Destination) matrix are not feasible (Bagchi and White, 2005). However, OD information is crucial at various stages of traffic engineering, including planning, operation, and management. Generating the OD matrix requires an awareness of the passenger flow dynamic between stations within the network. In this section, we leverage the graph attention coefficients derived earlier to construct time-based station weight features and build a regression model to estimate the passenger flow volume between stations. It is important to note that in this research, instead of measuring crowdedness based on real-world concepts like the number of people transferred per unit of time, we define passenger flow volume as the change in popularity over a specific time interval.

The approach is as follows: In the previous objective, namely crowdedness forecasting, we obtained attention coefficients $\alpha_{ij}^k \in \mathbb{R}^L$ from K attention heads through the APT-GCN model, which represents the contribution of station j to station i's popularity at each time step. We reshape this data into a crowdedness shift feature matrix $U \in \mathbb{R}^{N\times(N\cdot L)}$ as follows:

$$\alpha_{ij} = \frac{1}{K}\sum_{k=1}^{K}\alpha_{ij}^k \tag{4.44}$$

Where α_{ij} can be denoted as:

$$\alpha_{ij} = [a_{i,j,1}, \cdots, a_{i,j,L}] \tag{4.45}$$

(4.46)

Then, reshape the matrix as,

$$u_i = [a_{i,1,1}, \cdots, a_{i,N,L}] \tag{4.47}$$

$$u_t = [u_{1,1}, \cdots, u_{N,1}, \cdots, u_{N,L}] \tag{4.48}$$

$$\tag{4.49}$$

Next, we use the crowdedness shift features u_t as the input and the popularity y_t as the dependent variable to train a Lasso regression model. The passenger flow volume is then calculated as follows:

$$y_t = \beta_0 + \sum_{j=1}^{M} \beta_{j,t-nT} u_{t-nT,j} + \epsilon_t \tag{4.50}$$

$$p_{t-nT} = y_t \times \beta_{j,t-nT} \tag{4.51}$$

Where y_t is the predicted popularity at time t, M is the number of crowdedness shift features, $\beta_{j,t-nT}$ represents the coefficients corresponding to the features $u_{t-nT,j}$, and j is j^{th} station on line, and $t - nT$ represent previous n^{th} interval from current time. Finally, p_t denotes the estimated passenger flow volume matrix at time t.

The objective function for Lasso regression to minimize is,

$$\min_{\beta_0,\beta} \left(\frac{1}{2n_samples} \sum_{i=1}^{n_{samples}} \left(y_i - \beta_0 - \sum_{j=1}^{N...L} \beta_j u_{ij} \right)^2 + \alpha \sum_{j=1}^{M} |\beta_j| \right) \tag{4.52}$$

Where $n_{samples}$ is the number of samples, M is the number of crowdedness shift features, y_i is the predicted popularity, and α is the regularization parameter that controls the strength of the penalty applied to the coefficients.

Since the features are composed of the attributes of station j at each previous interval n, the coefficient $\beta_{j,t-nT}$ captures the contribution of the corresponding station to the crowdedness of the current station. If any coefficient $\beta_{j,t-nT}$ is negative, indicating reverse flow, we adjust it by taking the absolute value and adding it to station j's coefficient $\beta_{k,t-nT}$, while setting $\beta_{j,t-nT}$ to zero. By summing the estimated passenger flow volumes from all previous intervals, we construct the crowdedness shift matrix P for the entire network. A crowdedness shift matrix is analogous to an OD matrix in $\mathbb{R}^{N \times N}$. However, instead of representing the actual number of passengers, each value in the matrix denotes the passenger flow volume defined by the crowdedness index, the popularity.

4.6.2. Network Centrality Analysis

Building on the previously derived crowdedness shift matrix, we now apply network centrality analysis to further investigate the relationships and significance of stations within the urban rail transport network. This analysis aims to identify how stations contribute to or absorb passenger flows and determine their overall importance within the network. Based on the nature of our study and the characteristics of the collected data, the following centralities are selected to quantify station influence and connectivity: weighted out-degree centrality, weighted in-degree centrality, weighted eigenvector centrality, and PageRank centrality.

The purpose of using these centralities is to capture different aspects of station influence: Weighted out-degree and in-degree centralities are employed to directly express each station's role in either contributing to or absorbing crowdedness within the network. Weighted eigenvector centrality and PageRank centrality are used to reflect the significance of a station based on the overall crowdedness shifts in the surrounding area, emphasizing its importance within the broader network structure.

The following equations define each centrality:

- Weighted In-degree Centrality: This measures the total incoming crowdedness at a station i, indicating the extent to which the station absorbs passengers.

  $$v_i = \sum_{j=1}^{N} P_{ji} \tag{4.53}$$

  Where P_{ji} is the passenger flow from station j to station i in the crowdedness shift matrix P, and v_i is the in-degree centrality of station i.

- Weighted Out-degree Centrality: This represents the total outgoing crowdedness from station i, indicating the extent to which the station contributes to the overall crowdedness in the network.

  $$v_o = \sum_{j=1}^{N} P_{ij} \tag{4.54}$$

  Where P_{ij} is the passenger flow from station i to station j in the crowdedness shift matrix P, and v_o is the out-degree centrality of station i.

- Weighted Eigenvector Centrality: This centrality assigns higher scores to stations that are connected to other well-connected stations, emphasizing the influence of stations based on the connectivity of their neighbors.

  $$v_e = \frac{1}{\lambda} \sum_{j=1}^{N} P_{ij} v_{e,j} \tag{4.55}$$

  Where v_e is the eigenvector centrality of station i, λ is the eigenvalue, and $v_{e,j}$ represents the eigenvector centrality of station j.

- PageRank Centrality: This centrality takes into account both the quantity and quality of connections, giving higher importance to stations that are connected to other influential stations. The PageRank centrality is defined as:

$$v_p(i) = \frac{1-d}{N} + d\sum_{j=1}^{N} \frac{P_{ji}}{\sum_{k=1}^{N} P_{jk}} v_p(j) \tag{4.56}$$

  Where $v_p(i)$ is the PageRank centrality of station i, d is the damping factor (typically set to 0.85), and P_{ji} represents the crowdedness shift from station j to station i.

In this analysis, the weight of each edge in the graph is defined by the crowdedness volume from the crowdedness shift matrix rather than using the actual adjacency matrix A of the physical transport network. This choice is justified because, within the context of this study, during the observed time intervals (spanning at least one data collection period), stations not directly adjacent to the physical network may still be considered reachable due to passenger flow shifts in such a period.

5. Experiment Setup

This chapter outlines the preparations and configurations for our research. First, the study area and data preprocessing are introduced, followed by model setup. Then, the evaluation metrics for the prediction model performance assessment are presented.

5.1. Study Area

Our research focuses on twelve urban transit lines as the primary subjects of analysis. These lines span eight European cities of varying sizes, collectively encompassing a total of 428 stations, which reflects the considerable scale and diversity of the dataset employed in this study. Each transit line is treated as an individual public transport network, which is then analyzed through the models. The popularity data for these stations were collected over a period of nearly four months, from February 7, 2024, to June 4, 2024.

Special event scenarios are particularly taken into consideration. We have chosen football matches as a representative example. Football, being one of the most globally popular sports, frequently draws large crowds, enriching public life but simultaneously imposing substantial pressure on urban transit systems. In our study, we focus on 12 football clubs from top European leagues, along with their associated stadiums, which are related to one or more stations within our study area. The matches held at these stadiums, including both league and cup games, are classified as special events in our research.

For the purposes of this study, the start time of each match is considered the beginning of the event. Given the typical 90-minute duration of a football match, along with halftime and possible added time, we estimate the event duration to be two hours. Thus the end time is set to two hours after the start. During the data collection period, a total of 122 football matches took place.

Table 5 provides an overview of the study area, while Table 6 illustrates an example of the special event data used in this research.

City	Line ID	Stops	Football Club
Munich	U6	Fröttmanning	Bayern Munich
Berlin	S3	Berlin-Köpenick	1.FC Union Berlin
Madrid	Line 10	Santiago Bernabéu	Real Madrid
Madrid	Line 7	Estadio Metropolitano	Atletico Madrid
Madrid	Line 12	Los Espartales	Getafe
Madrid	Line 1	Portazgo	Rayo Vallecano
London	Piccadilly	Arsenal	Arsenal
London	District	Fulham Broadway	Chelsea
Newcastle	Yellow	St James	Newcastle United
Marseille	M2	Sainte-Marguerite Dromel	Olympique de Marseille
Copenhagen	M3	Trianglen St.	F.C. Copenhagen
Lisbon	Blue	Colégio Militar/Luz	Benfica

Table 5 Overview of study area

City	Line	Event	Start Time	End Time
Copenhagen	M3	FC Copenhagen-Manchester City	13.02.2024, 21:00	13.02.2024, 22:55
Lisbon	Blue	Benfica-Toulouse	15.02.2024, 21:00	15.02.2024, 22:58
Madrid	Line 7	Atletico Madrid-Las Palmas	17.02.2024, 14:00	17.02.2024, 15:51
London	Yellow	Newcastle-Bournemouth	17.02.2024, 16:00	17.02.2024, 17:58
Lisbon	Blue	Benfica-Vizela	18.02.2024, 19:00	18.02.2024, 20:58
Marseille	M2	Marseille-Shakhtar	22.02.2024, 21:00	22.02.2024, 22:59

Table 6 Example of special event data

5.2. Data Preprocessing

In this study, we utilize GPT (Google Popular Time) data (Google, n.d.) as our primary source to quantify crowdedness levels at public transport stations. GPT provides an index 'popularity' ranging from 0 to 100, where 0 represents the least crowded state, and 100 indicates the highest level of congestion. Each public transport station is treated as a Point of Interest (POI), and GPT data is collected periodically at predefined time intervals for these stations. The raw GPT data includes several components, such as 'current popularity', which indicates the crowd level at a given POI at the current moment, and 'historical popularity', which represents the average crowd level for each hour of each day of the week based on data collected over the past several weeks, thus providing 168 (7 days × 24 hours) data points that illustrate how crowded the location typically is during different times of the day. In addition, the attribute 'visit duration' estimates the typical amount of time that customers spend at the POI based on historical visitation patterns, while 'wait time' estimation provides an estimate of how long a customer might have to wait for service at various times of the day (Google, n.d.). Leveraging GPT data helps to address potential gaps in data availability for specific regions or research subjects and supports the construction of a robust dataset to analyze crowdedness patterns effectively.

Given the nature of GPT data collection and the constraints encountered during the data acquisition process, it is necessary to perform a series of preprocessing steps to ensure data consistency and quality. One significant challenge is that GPT data is not always recorded at uniform time intervals. To mitigate this issue, we applied a rounding procedure to standardize the time stamps to 30-minute intervals, thereby harmonizing the data and enabling further analysis. After standardizing the time intervals, we conducted a comprehensive data quality analysis to identify and address any missing or anomalous values. This preprocessing step ensures that the data is suitable for subsequent modeling and analysis, thereby improving the reliability and accuracy of our results. As detailed in Chapter 3, the preprocessed station-related temporal data serves as the basis for constructing the feature set used in our predictive models.

5.2.1. Data Quality Analysis and Assumption

During the data collection period, the GPT data was not consistently gathered for all stations and times, despite the best efforts to ensure comprehensive coverage. Data collection failures may result from network instability, inherent unavailability from Google, or other unforeseen issues. Given these potential gaps, it is critical to assess the quality of the collected data. This evaluation is performed at both the urban transit line and station levels and consists of two main components.

First, the amount of data obtained from each station along the urban public transit line is assessed to ensure sufficient coverage across all stations. Second, for each station, a statistical analysis of the GPT current popularity values is conducted. Stations with a limited number of data points or stations exhibiting abnormal statistical distributions of the "current popularity" (e.g., an average value close to zero) are considered invalid and are subsequently deleted from the dataset.

Using Madrid Line 1 as an example, Figure 7 presents the statistics of popularity values at various stations over the data collection period, while Figure 8 illustrates the distribution of data points collected over a typical week. Given that data was collected from each station every 30 minutes, there should ideally be 336 data points per week. Stations with abnormal, poor-quality data or significant gaps in data—such as Atocha station in this case—were removed from the dataset.

The exclusion of invalid stations relies on two key assumptions. Specifically, we assume that stations with insufficient or anomalous data do not exhibit unique or distinct passenger flow patterns that would impact the overall analysis. In other words, the stations we retain are sufficient for capturing the typical patterns in the network, and the removal of these stations will not introduce new, uncaptured patterns. Therefore, the remaining data is adequate for the models to effectively learn and make accurate predictions. Additionally, we assume that the popularity, representing passenger traffic flow, will move fluidly along the transit line, including any new or increased demand at the deleted stations. In this way, the flow of passengers

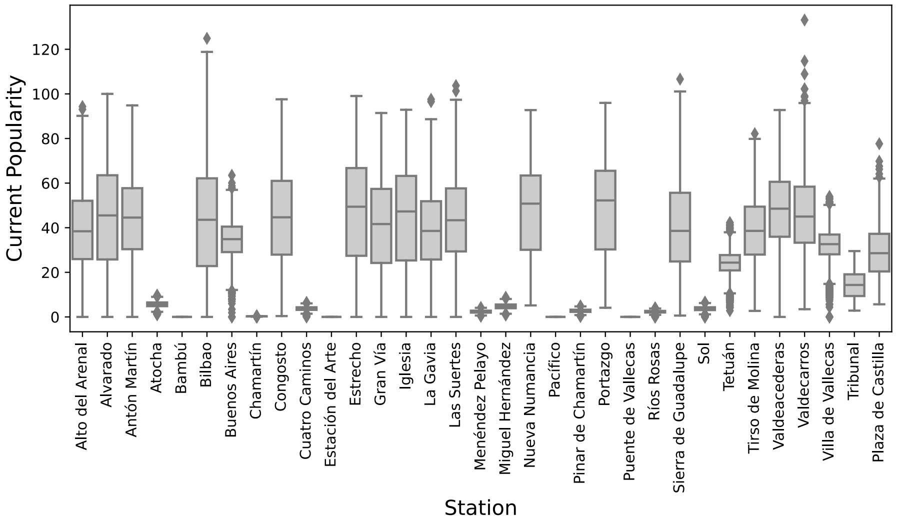


Figure 7 Popularity value distribution of stations in Madrid Line 1

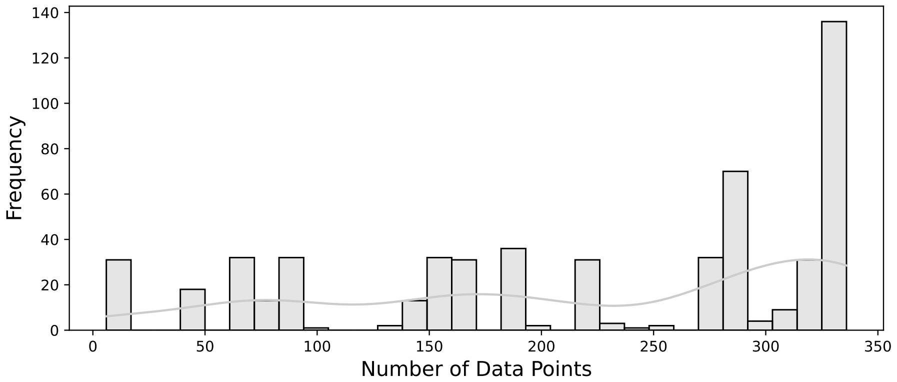


Figure 8 The distribution of popularity data collected for stations in Madrid Line 1 shows the distribution of data amounts collected each week at various stations along Madrid Line 1. The x-axis represents the number of data points collected for a given week at a given station, while the y-axis indicates the count of station-week observations corresponding to the number of data points collected

at adjacent or nearby stations will naturally absorb and reflect the patterns that would have occurred at the excluded stations. Thus, the retained stations provide a comprehensive and accurate view of the overall passenger flow, ensuring that the models can still capture essential patterns and make reliable predictions without introducing significant bias.

5.2.2. Data Imputation

For stations where the number of collected data points exceeds the threshold, data imputation is necessary to handle the missing values. Given the missing data patterns in our dataset, the problem is defined as a random missing spatial-temporal data problem, where each sensor loses observation completely at random. To address this issue, several algorithms can be applied, such as BPMF (Bayesian Probabilistic Matrix Factorization) (Salakhutdinov and Mnih, 2008), TRMF (Temporal Regularized Matrix Factorization) (Yu et al., 2016), BATF (Bayesian Augmented Tensor Factorization) (Chen et al., 2019), among others. In this study, we utilize TRMF. This model incorporates temporal dependencies into the matrix factorization process.

Temporal dependencies are represented by x_i,

$$x_t \approx \sum_{l \in \mathcal{L}} \theta_l \odot x_{t-l} \tag{5.1}$$

Then,

$$\mathcal{R}_{\mathsf{AR}}(X \mid \mathcal{L}, \Theta, \eta) = \frac{1}{2} \sum_{t=l_d+1}^{f} \left(x_t - \sum_{l \in \mathcal{L}} \theta_l \odot x_{t-l} \right)^{\top} \left(x_t - \sum_{l \in \mathcal{L}} \theta_l \odot x_{t-l} \right) + \frac{\eta}{2} \sum_{t=1}^{f} x_t^{\top} x_t \tag{5.2}$$

where $\mathcal{L} = \{l_1, l_2, \ldots, l_d\}$ is a lag set, in our task is setting to $\mathcal{L} = \{1, 2, 48\}$, and $\theta_l \in \mathbb{R}^r, \forall l$ is weight to decide autoregressive (AR). $\odot$ is the element-wise product. Thus the TRMF is to solve,

$$\min_{W,X,\Theta} \frac{1}{2} \sum_{(i,t) \in \Omega} \left(y_{it} - w_i^{\top} x_t \right)^2 + \lambda_w \mathcal{R}_w(W) + \lambda_x \mathcal{R}_{\mathsf{AR}}(X \mid \mathcal{L}, \Theta, \eta) + \lambda_\theta \mathcal{R}_\theta(\Theta) \tag{5.3}$$

where $\mathcal{R}_w(W) = \frac{1}{2} \sum_{i=1}^{m} w_i^{\top} w_i$ and $\mathcal{R}_\theta(\Theta) = \frac{1}{2} \sum_{l \in \mathcal{L}} \theta_l^{\top} \theta_l$ are regularization terms.

Our research applied the TRMF algorithm to address the missing data issue. After applying the imputation algorithms to complete the GPT data, we obtained a full set of temporal data for the revised station set, which contains 4,617 data points per station recorded at 30-minute intervals over the study period. These imputed temporal data are further calculated to popularity-related features, combined with other constructed features, to form the final station-related temporal dataset. The attribute matrix, combined with the adjacency matrix A representing the network topology and the special event data, constitutes the dataset used in our study.

5.3. Model Setup

In this experiment, all models are implemented using Python with specific libraries for different models. The ARIMA model is implemented with the statsmodels API (Seabold and Perktold, 2010), while the MLP, along with other regression models such as Linear Regression, KNN, and SVR, are constructed using scikit-learn (Pedregosa et al., 2011). For the deep learning models such as the LSTM and our proposed APT-GCN model, we use TensorFlow 2.16.1 (Abadi et al., 2015). All experiments were conducted on a MacBook Pro with an Apple M2 Max chip, featuring 32 GB of memory and 12 cores.

For the statistical models employed in this study, we conducted extensive performance testing to determine the most suitable kernels and parameters. This involved testing on a small sample of data, specifically data from Line 1 and Line 7. We observed no significant performance differences among the variations tested, including Lasso, Ridge, and Bayesian Linear Regression. As a result of this rigorous testing, we selected the standard Linear Regression model for its simplicity and efficacy. In the case of the Gaussian Process model, we found that the linear kernel outperformed the other options tested, providing reassurance about the reliability of our model selection. For the SVR model, we chose the linear kernel based on its superior performance in the preliminary tests. When configuring the KNN model, we used cross-validation techniques to select the optimal number of neighbors for each line, as shown in Table 7. All the settings for the models used in this study are summarized in Table 9.

Line	Number of neighbors
Line 1, Madrid	14
Line 7, Madrid	10
Line 10, Madrid	9
Line 12, Madrid	9
Line Blue, Lisbon	9
Line District, London	10
Line Piccadilly, London	11
Line M2, Marseilles	8
Line M3, Copenhagen	9
Line S3, Berlin	9
Line U6, Munich	10
Line Yellow, Newcastle	9

Table 7 Number of neighbors selected for KNN models across different lines

For the ARIMA model, different combinations of (p, d, q) and (P, D, Q, s) were tested on a small sample of data (data from line 7) to determine the optimal parameters. The model performance was evaluated using the AIC (Akaike Information Criterion) as the performance metric. The results are summarized in Table 8. It was observed that the performance differences among various parameter combinations were not very significant. Ultimately, for the ARIMA model, we selected the combination (3, 1, 1) for the non-seasonal parameters and (2, 1, 1, 8) for the seasonal parameters. Here, 8 represents the length L of the seasonal cycle, which will be consistently applied across all models that use time series as input.

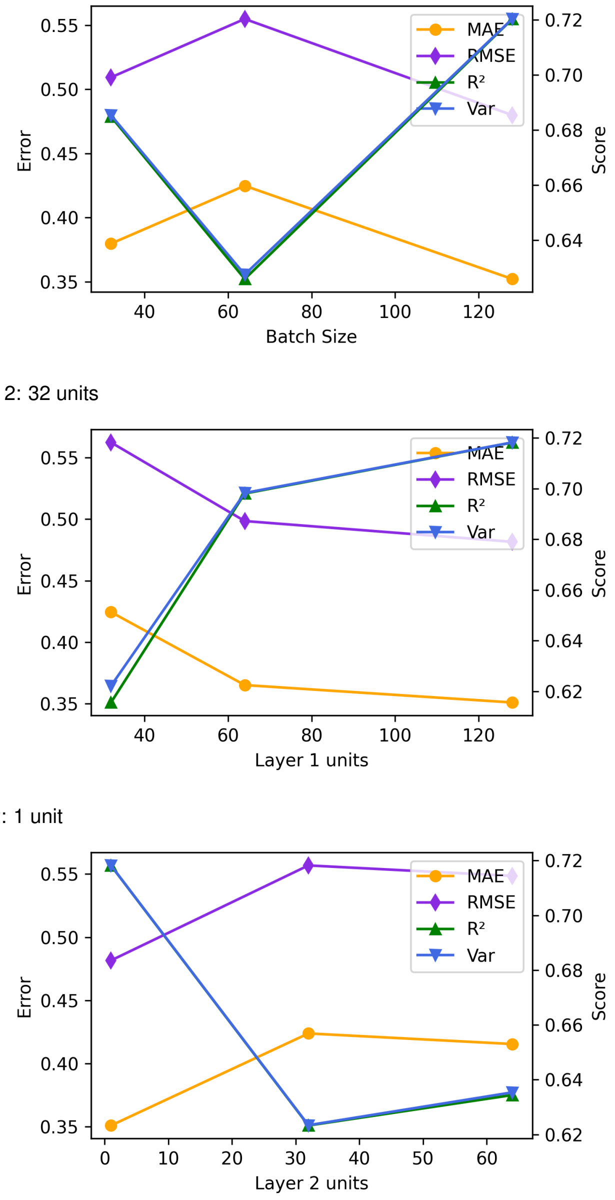
(p, d, q)	(P, D, Q, s)	**AIC**
(2, 1, 1)	(2, 1, 1, 0)	824.9142
(2, 2, 1)	(2, 1, 1, 0)	843.2799
(3, 1, 1)	(2, 1, 1, 0)	824.7306
(3, 2, 1)	(2, 1, 1, 0)	841.9540
(4, 1, 1)	(2, 1, 1, 0)	823.2314
(4, 2, 1)	(2, 1, 1, 0)	836.5101

Table 8 Performance comparison of ARIMA models with different parameters

The MLP model was implemented using scikit-learn's MLPRegressor. The architecture consists of two hidden layers, each containing 64 units, and the ReLU (Rectified Linear Unit) activation function was applied to introduce non-linearity. The model was trained with the Adam optimizer and a learning rate of 0.001. The number of epochs was set to 500, and the batch size to 64. Early stopping was also applied with a patience of 30 to prevent overfitting. The model outputs continuous predictions for the station crowdedness based on the input features.

The LSTM model, implemented using TensorFlow 2.16.1, was designed with two LSTM layers. After conducting parameter searches for the optimal number of units in each layer and the batch size on a small set of example data, the final configurations were determined based on the evaluation results presented in Figure 9. Since we employed the early stopping technique to prevent overfitting, allowing the model to stop training at the optimal epoch, we did not exhaustively test the number of epochs. The results indicated that the best configuration includes 128 units in the first layer, 32 units in the second layer, and a batch size of 64. A dropout rate of 0.2 was applied to mitigate overfitting, and the Adam optimizer with a learning rate of 0.001 was used. The model was trained for up to 1000 epochs, with early stopping triggered by a patience of 50 epochs, monitored by the validation loss to enhance training efficiency.

For our proposed APT-GCN model, the training process spans 1000 epochs, utilizing a batch size of 128. The dataset is split into training, validation, and test sets in a 7:1:2 ratio. The Adam optimizer is used with a learning rate of 0.001. During training, the MSE (Mean Squared Error) loss function is applied, and early stopping is implemented to enhance ef-



(a) Layer 1: 128 units, layer 2: 32 units

(b) Batch size: 128, Layer 2: 1 unit

(c) Batch size: 128, Layer 1: 128 unit

Figure 9 The influence of hyper-parameter selection for LSTM model

ficiency, with validation loss as the monitor and a patience value of 50. The model's hidden layers are configured with 64 units. The historical time series length L is set to 8, and the model performs one-step predictions, with the prediction length being 1.

Table 9 summarizes the settings for all models used in the experiment.

Model	**Parameter**	**Settings**
Linear Regression	Kernel	Standard
Gaussian Process	Kernel	Linear
SVR	Kernel	Linear
glaknn	Number of neighbors	Automatically decide by cross-validation
ARIMA models	(p, d, q)	(3, 1, 1)
	(P, D, Q, s)	(2, 1, 1, 8)
	Maximum iteration	200
	Exogenous features for SARIMAX	Popularity from the last three intervals^4, event index^4
MLP	Hidden layers	2
	Units	64
	Activation	ReLU
	Optimizer	SGD
	Learning rate	0.001
	Epochs	500
	Batch size	64
LSTM	Units of first layer	128
	Units of second layer	32

Table 9 – *Conti.*

Model	**Parameter**	**Settings**
LSTM	Optimizer	Adam
	Learning rate	0.001
	Epochs	1000, with early stopping
	Batch size	64
	Early stopping patience	50
APT-GCN	Units	64
	Optimizer	Adam
	Learning rate	0.001
	Epochs	1000
	Batch size	128
	Early stopping patience	50
	Loss Function	MSE
	Exogenous features	Popularity from the last three intervals [4], event index [4]

Table 9 Model settings for experiments in this research

5.4. Evaluation Metrics

To comprehensively evaluate the performance of the crowdedness prediction models, we employ several standard metrics. These metrics are used to compare the predicted values against the actual values to assess the model's accuracy and consistency. The following evaluation metrics are introduced:

- RMSE (Root Mean Squared Error): RMSE measures the square root of the average of the squared differences between predicted and actual values. It gives higher weight to larger errors, making it useful for assessing models where large errors are undesirable.

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2} \tag{5.4}$$

- MAPE (Mean Absolute Percentage Error): MAPE computes the average percentage difference between the predicted and actual values, making it useful for comparing models across datasets with different scales.

$$\text{MAPE} = \frac{100}{M}\sum_{i=1}^{M}\left|\frac{\mathbf{y}_i - \hat{\mathbf{y}}_i}{\mathbf{y}_i}\right| \tag{5.5}$$

- R^2 (Coefficient of Determination): R^2 provides a measure of how well the predicted values explain the variance in the actual values. An R^2 value of 1 indicates a perfect predictions, while a value of 0 indicates that the model does no better than a simple mean prediction.

$$R^2 = 1 - \frac{\sum_{i=1}^{M}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2}{\sum_{i=1}^{M}(\mathbf{y}_i - \bar{\mathbf{y}})^2} \tag{5.6}$$

- Var (Explained Variance Score): This metric measures the proportion of the variance in the actual data that is explained by the model. It ranges from 0 to 1, with 1 indicating perfect explanation of variance.

$$\text{Var} = 1 - \frac{\text{Var}(\mathbf{Y} - \hat{\mathbf{Y}})}{\text{Var}(\mathbf{Y})} \tag{5.7}$$

Here, M is the number of samples, $\mathbf{y}_i$, $\hat{\mathbf{y}}_i$, and $\bar{\mathbf{y}}_i$ represent the actual, predicted, and average popularity values for the i-th sample, respectively. $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ represent the full set of actual and predicted values, while $||\cdot||$ denotes the Frobenius norm.

6. Result and Discussion

6.1. Crowdedness Pattern Overview

This section provides an overview of the crowdedness patterns at urban rail transit stations, represented using the popularity data from GPT, as introduced earlier.

6.1.1. Weekday and Weekend Comparison

Pattern 1: Similar waveforms with higher weekday popularity

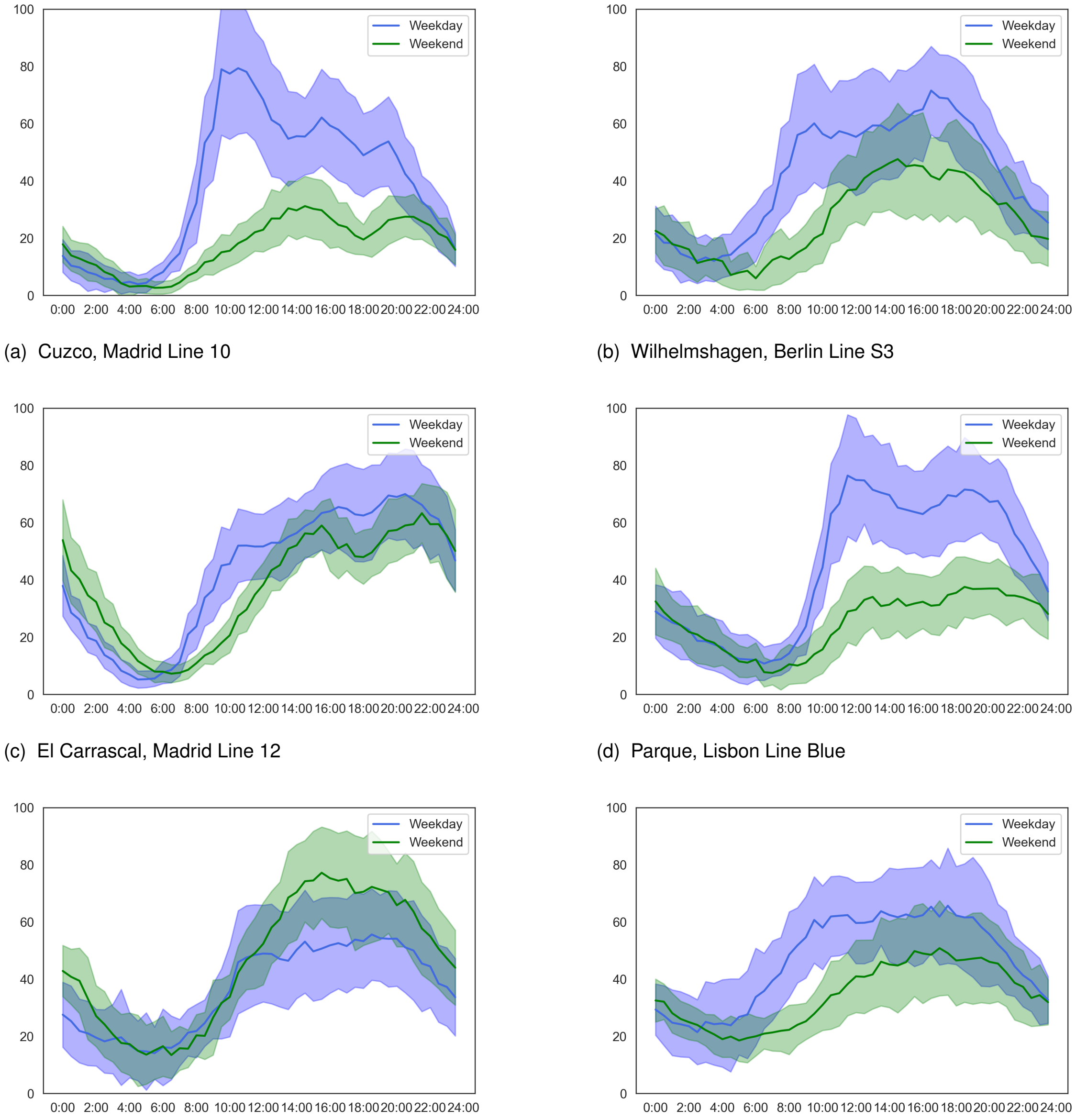
This is the most common station crowdedness pattern observed. The peaks and troughs in the popularity curve occur at similar times on both weekdays and weekends, which is shown in Figure 10. Within this pattern, we can further distinguish between two sub-patterns: single-peak and double-peak curves. In the single-peak pattern, the station's crowdedness peak typically occurs during the evening rush hour, which varies between 16:00 and 22:00 depending on the city and metro line. The double-peak pattern, in addition to the evening peak, exhibits a midday peak, usually between 10:00 and 14:00. Notably, for the double-peak pattern, the weekend peak is often less pronounced than the weekday peak, or the two peaks may even merge altogether(Figure 10d, 10b). Furthermore, the midday peak on weekends tends to occur later compared to weekdays (Figure 10a, 10c).

There are, however, exceptions. For instance, in stations along Madrid Metro Line 12, the midday peak on weekends can be observed more significantly than on weekdays (Fig: 10c). Another intriguing phenomenon observed is that, despite the near-parallel nature of the weekday and weekend curves, many stations exhibit a higher average popularity during midnight hours (typically 0:00–4:00) on weekends compared to weekdays.

Although the waveforms are similar under both scenarios, there are notable differences in the values of crowdedness. In some cases, the curves remain close during non-peak hours, while the differences between weekday and weekend crowdedness grow significantly during peak periods. While at some stations, the curves are nearly parallel throughout the day, indicating stable crowdedness fluctuations across the week (Figure 10f).

Pattern 2: Diverging peaks between weekday and weekend (weekday double-peak, weekend single-peak)

This pattern is almost exclusive to stations in London. In this pattern, the weekday curve exhibits two peaks, while the weekend curve shows only one, i.e., the evening peak. However, this weekend peak reaches or surpasses the weekday peak at the same time of day. Similar but rare occurrences of this pattern are also found in a few stations on Copenhagen's M3 and Marseilles's M2 lines (Figure 11a).



(a) Cuzco, Madrid Line 10

(b) Wilhelmshagen, Berlin Line S3

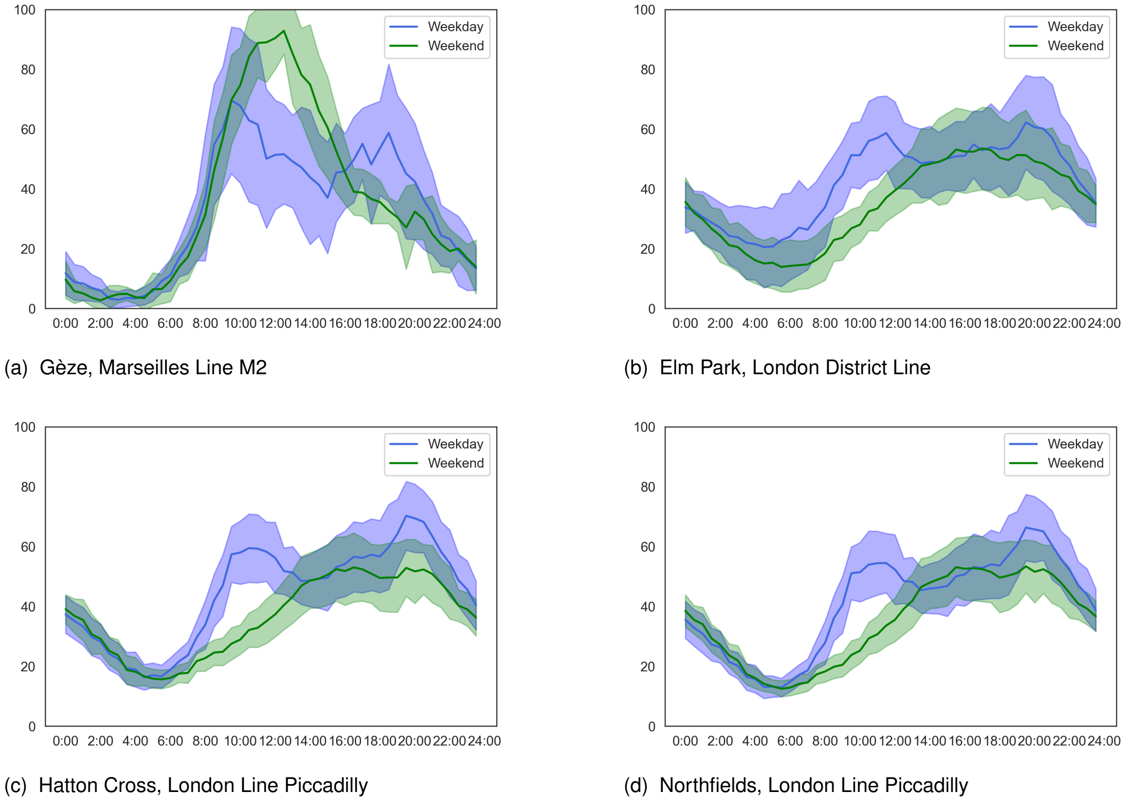
(c) El Carrascal, Madrid Line 12

(d) Parque, Lisbon Line Blue

(e) Tynemouth, Newcastle Line Yellow

(f) Partnachplatz, Munich Line U6

Figure 10 Crowdedness pattern 1: Similar waveforms with higher weekday popularity



(a) Gèze, Marseilles Line M2

(b) Elm Park, London District Line

(c) Hatton Cross, London Line Piccadilly

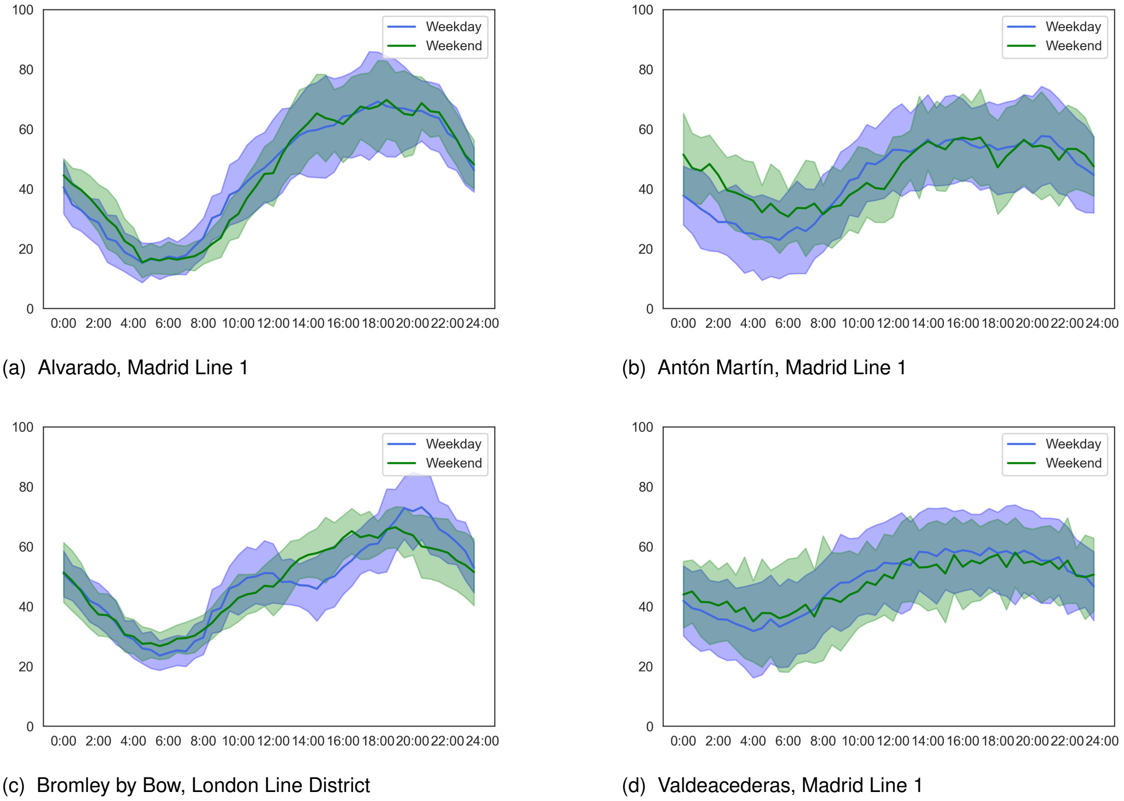
(d) Northfields, London Line Piccadilly

Figure 11 Crowdedness pattern 2: Diverging peaks between weekday and weekend

Pattern 3: Overlap of Weekday and Weekend Curves

Figure 12 illustrates this pattern. This pattern is primarily observed along Madrid Metro Line 1 and London Line Piccadilly, where most stations display similar crowded patterns on weekdays and weekends. Interestingly, the average weekend popularity is slightly higher than that on weekdays at some stations, a rare phenomenon among the stations in our research area. Noticeably, not all stations on line 1 appear to adhere to this pattern, and several stations fall into the first category above.

Based on the analysis, we classify station crowdedness patterns into three main types: similar waveforms with higher weekend crowdedness, weekday double-peak, and weekend single-peak, and nearly identical weekday and weekend curves. Additionally, we observed that stations on the same metro line tend to exhibit a dominant pattern. For instance, the majority of stations on the London Piccadilly Line follow the second pattern, while Madrid Line 10 predominantly features the first pattern, with most stations even sharing the same sub-type (the double-peak form), highlighting significant consistency along the line. In contrast, stations on Munich's U6 line mainly belong to the single-peak sub-type within pattern one. However, no specific pattern emerges for stations across different lines within the same city. For example, in our analysis of four Madrid metro lines (Lines 1, 7, 10, and 12), no commonality in crowdedness patterns was observed between the lines. Similarly, we found no distinctive pattern shared by cities within the same country.



(a) Alvarado, Madrid Line 1

(b) Antón Martín, Madrid Line 1

(c) Bromley by Bow, London Line District

(d) Valdeacederas, Madrid Line 1

Figure 12 Crowdedness pattern 3: Overlap of Weekday and Weekend Curves

These classifications enhance our understanding of public transport station crowdedness patterns and provide insights into transport demand studies, public transport network analysis, and comparative studies of station usage characteristics across cities and metro lines. This knowledge can also contribute to urban mobility planning, transit system optimization, and demand-responsive transit service design.

6.1.2. Crowdedness Pattern under Special Event Scenario

Figure 14, 16, 17 and 19 illustrates the spatio-temporal distribution of station popularity along selected metro lines during special event periods. The events, along with their details, are listed in Table 10. In the popularity heatmaps, stations are ordered according to their real-world placement on each line (Madrid Line 12 is a loop), with the stations corresponding to event venues displayed in blue text.

For Madrid Line 12, as seen in Figure 13, this line is relatively distant from the city's core, fully located in fare zones B1 and B2. In contrast, Line 1 traverses central Madrid, although the stadium on Line 1, Estadio de Vallecas, is not located in the city center, and its capacity is relatively small compared to larger venues like Santiago Bernabéu. The District and Piccadilly Lines are part of the London Tube system, characterized by their branch structures. For the District Line, we focus on the Ealing Broadway branch, and for Piccadilly, we analyze the Heathrow Terminal 4 branch. Both lines run through multiple key areas of London, serving as essential corridors for commuter and tourist traffic, reflecting their central role in the

City	Line	Event	Start Time	End Time
Madrid	Line 12	Getafe - Las Palmas	20240302-1830	20240302-2030
Madrid	Line 12	Getafe - Sevilla	20240330-1400	20240330-1600
London	District	Chelsea - Manchester Union	20240404-2115	20240404-2315
Madrid	Line 1	Rayo Vallecano - Getafe	20240413-1615	20240413-1815
Madrid	Line 1	Rayo Vallecano - Osasuna	20240420-1615	20240420-1815
London	Piccadilly	Arsenal - Chelsea	20240423-2100	20240423-2300
London	District	Chelsea - Bournemouth	20240519-1700	20240519-1900
London	Piccadilly	Arsenal - Everton	20240519-1700	20240519-1900

Table 10 Details of football matches near selected stations

transportation network of this bustling metropolis.

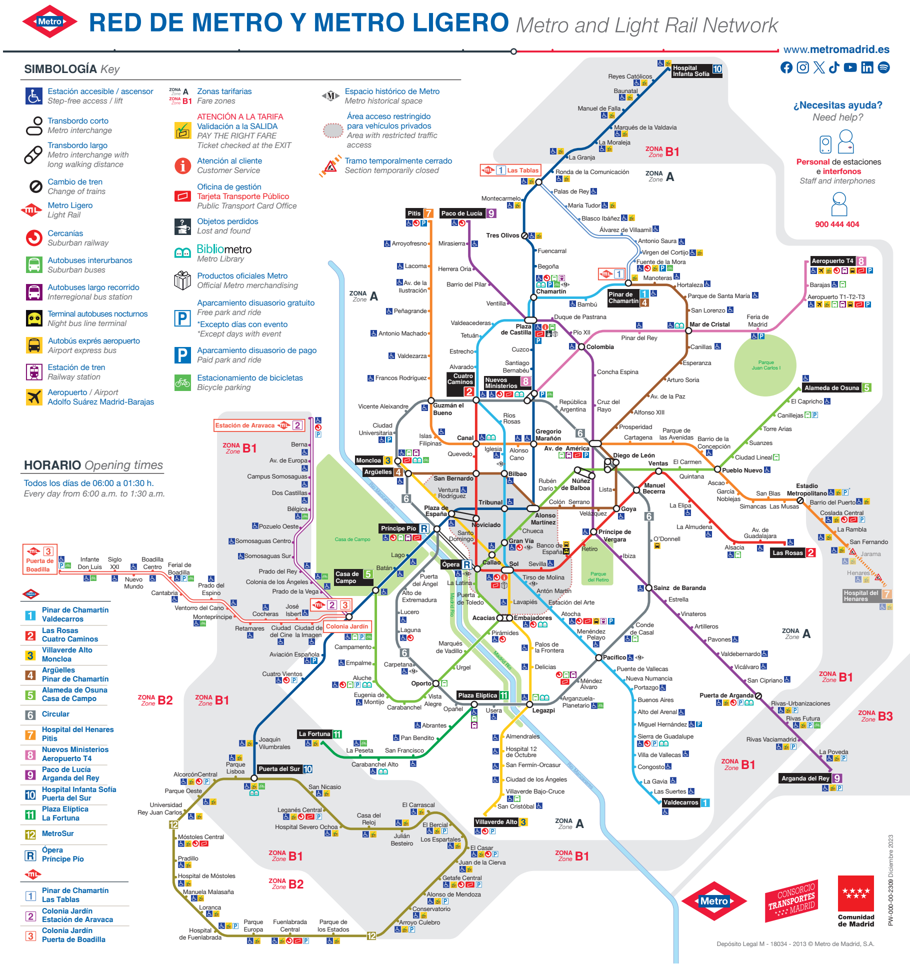
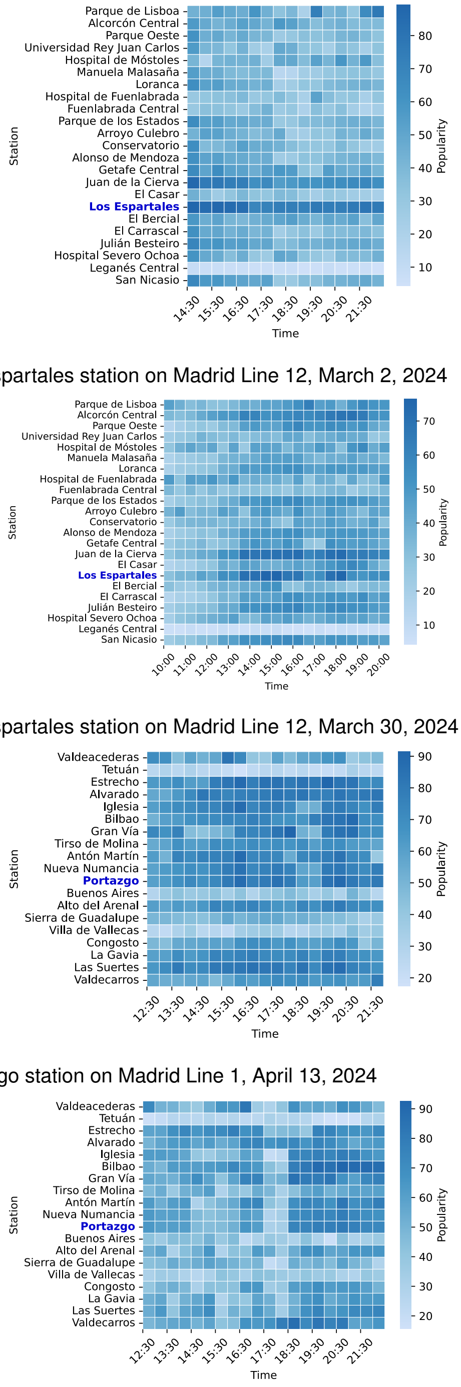

Metro
RED DE METRO Y METRO LIGERO Metro and Light Rail Network
www.metromadrid.es
SIMBOLOGÍA Key
Estación accesible / ascensor Step-free access / lift
Transbordo corto Metro interchange
Transbordo largo Metro interchange with long walking distance
Cambio de tren Change of trains
Metro Ligero Light Rail
Cercanías Suburban railway
Autobuses interurbanos Suburban buses
Autobuses largo recorrido Interregional bus station
Terminal autobuses nocturnos Night bus line terminal
Autobús exprés aeropuerto Airport express bus
Estación de tren Railway station
Aeropuerto / Airport Adolfo Suárez Madrid-Barajas
Zonas tarifarias Fare zones
ATENCIÓN A LA TARIFA Validación a la SALIDA PAY THE RIGHT FARE Ticket checked at the EXIT
Atención al cliente Customer Service
Oficina de gestión Tarjeta Transporte Público Public Transport Card Office
Objetos perdidos Lost and found
Bibliometro Metro Library
Productos oficiales Metro Official Metro merchandising
Aparcamiento disuasorio gratuito Free park and ride
*Excepto días con evento *Except days with event
Aparcamiento disuasorio de pago Paid park and ride
Estacionamiento de bicicletas Bicycle parking
Espacio histórico de Metro Metro historical space
Área acceso restringido para vehículos privados Area with restricted traffic access
Tramo temporalmente cerrado Section temporarily closed
¿Necesitas ayuda? Need help?
Personal de estaciones e interfonos Staff and interphones
900 444 404
HORARIO Opening times
Todos los días de 06:00 a 01:30 h.
Every day from 6:00 a.m. to 1:30 a.m.
1 Pinar de Chamartín Valdecarros
2 Las Rosas Cuatro Caminos
3 Villaverde Alto Moncloa
4 Argüelles Pinar de Chamartín
5 Alameda de Osuna Casa de Campo
6 Circular
7 Hospital del Henares Pitis
8 Nuevos Ministerios Aeropuerto T4
9 Paco de Lucía Arganda del Rey
10 Hospital Infanta Sofía Puerta del Sur
11 Plaza Elíptica La Fortuna
12 MetroSur
R Ópera Príncipe Pío
1 Pinar de Chamartín Las Tablas
2 Colonia Jardín Estación de Aravaca
3 Colonia Jardín Puerta de Boadilla
ZONA A
ZONA B1
ZONA B2
ZONA B3
Depósito Legal M - 18034 - 2013 © Metro de Madrid, S.A.
CONSORCIO TRANSPORTES MADRID
Comunidad de Madrid

Figure 13 Madrid metro map (Metro de Madrid, 2024)

As shown in Figures 14a and 14b, during special events, the stations near stadiums on Line 12 experience significantly higher popularity compared to surrounding stations. Figure 15 demonstrates that during game times, the popularity is markedly higher than on regular days, confirming the substantial impact of special events on crowdedness patterns, which can introduce challenges for prediction models that must effectively capture and adapt to these shifts. Moreover, by comparing the event start and end times, we can observe a certain delay in peak popularity in the heatmaps. Conversely, stations on Madrid Line 1, aside from those near the stadium, show patterns similar to regular weekday peaks, particularly during the evening rush hour, unaffected by event participants. At the Portazgo station, however, the crowdedness pattern during special events differs significantly from typical days, with elevated levels of crowding. Interestingly, at other high-traffic stations on Line 1, such as Gran Vía and Antón Martín (the former being an interchange station with Line 5 and located in Madrid's historic city center), the crowdedness remains unaffected by events, indicating that these stations do not pose additional challenges for the predictive model during special events.

The same phenomenon can be observed at Arsenal station on London's Piccadilly Line.



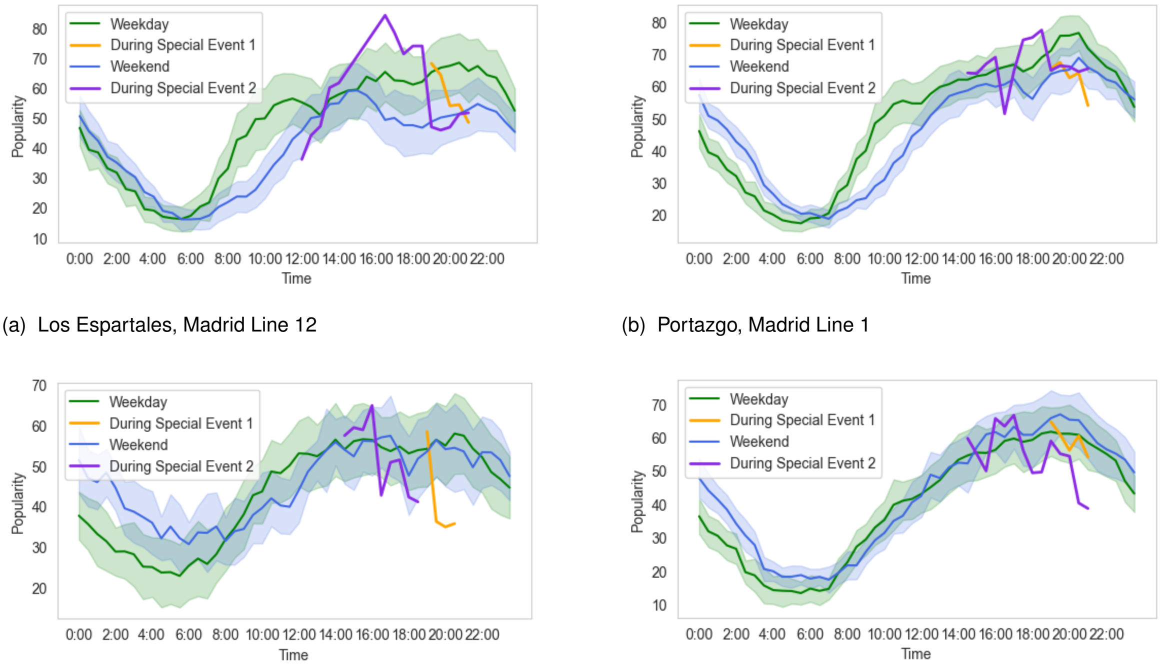
(a) Match at the stadium near Los Espartales station on Madrid Line 12, March 2, 2024

(b) Match at the stadium near Los Espartales station on Madrid Line 12, March 30, 2024

(c) Match at the stadium near Portazgo station on Madrid Line 1, April 13, 2024

(d) Match at the stadium near Portazgo station on Madrid Line 1, April 20, 2024

Figure 14 Spatio-temporal distribution of station popularity along urban transit lines during football matches in Madrid



(a) Los Espartales, Madrid Line 12

(b) Portazgo, Madrid Line 1

(c) Antón Martín, Madrid Line 1

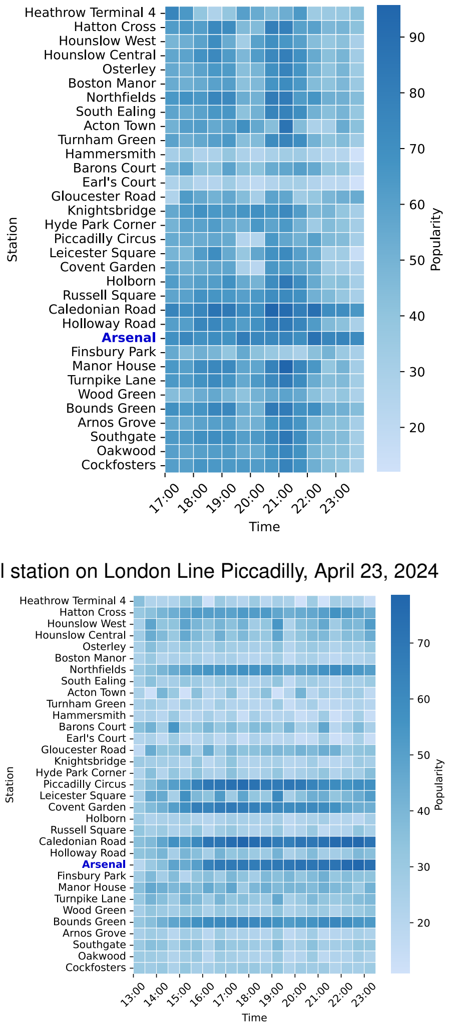
(d) Gran Vía, Madrid Line 1

Figure 15 Crowdedness pattern under special event scenario, Madrid

Located in the borough of Islington, East London, Arsenal station serves Emirates Stadium, home to Arsenal Football Club, one of the largest stadiums in the country with a substantial fanbase. As shown in Figures 16a and 16b, stations along this line, including Arsenal, exhibit significantly higher popularity during events compared to surrounding stations. Upon further examination of Figures 16a and 16b, a pattern similar to that of Madrid Line 1 emerges: while Arsenal station shows significantly increased popularity during events compared to regular days, other stations along the Piccadilly Line, such as Piccadilly Circus (a major interchange station with high footfall), do not display such increases in crowding.

An interesting observation is that on May 19, 2024, two simultaneous football matches took place in London: one at Arsenal's Emirates Stadium and another at Chelsea's Stamford Bridge. Figures 16b and 17b show the spatial and temporal distribution of popularity on the corresponding metro lines. While the Piccadilly Line's Arsenal station saw a significant crowd gathering, the District Line's Fulham Broadway station, serving Stamford Bridge, did not experience similar crowding despite hosting an event. Figure 17a further confirms this. This highlights that the impact of special events on station crowdedness patterns is not uniform and may be influenced by multiple factors, such as the event's scale, venue accessibility, and even cultural or geographical differences across cities.

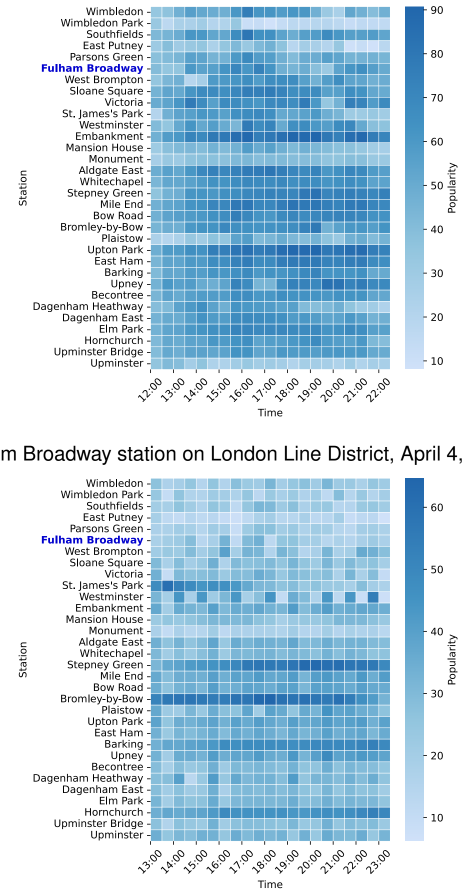
Lastly, let us consider an example from Munich's U6 line. The Fröttmaning station is situated on the city's outer periphery and primarily serves Allianz Arena, home to FC Bayern Munich, with few other facilities in its vicinity. Given this, daily footfall at this station is relatively low, but football matches at Allianz Arena have a significant impact. As seen in Figure 20a, the



(a) Match at the stadium near Arsenal station on London Line Piccadilly, April 23, 2024

(b) Match at the stadium near Arsenal station on London Line Piccadilly, May 19, 2024

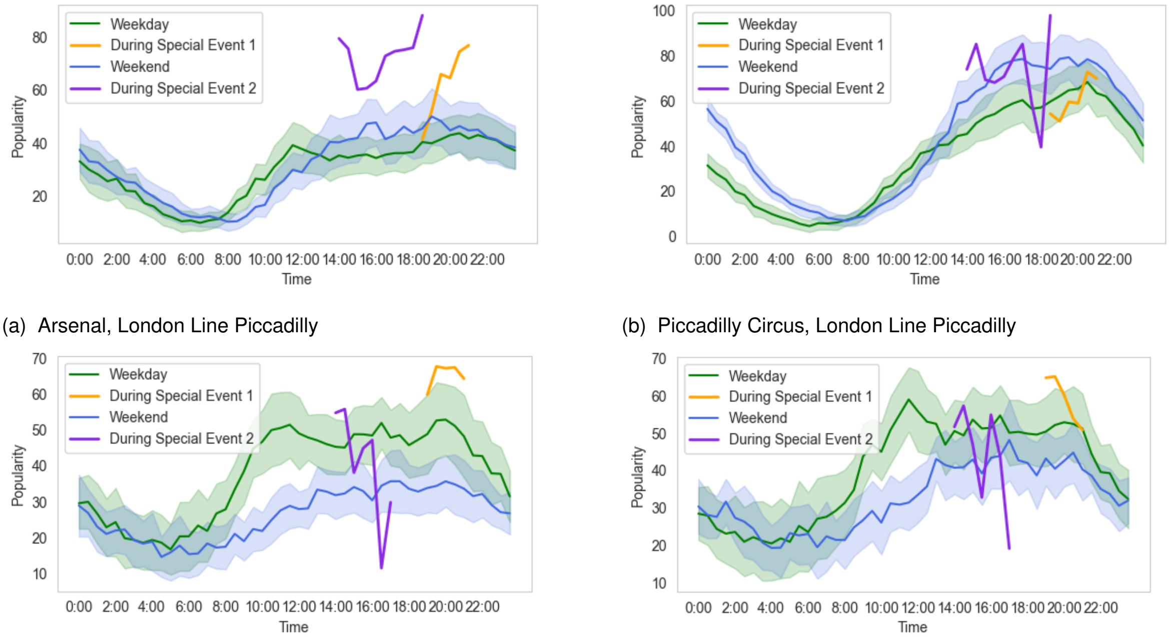
Figure 16 Spatio-temporal distribution of station popularity along urban transit lines during football matches at Line Piccadilly, London



(a) Match at the stadium near Fulham Broadway station on London Line District, April 4, 2024

(b) Match at the stadium near Fulham Broadway station on London Line District, May 19, 2024

Figure 17 Spatio-temporal distribution of station popularity along urban transit lines during football matches at Line District, London



(a) Arsenal, London Line Piccadilly

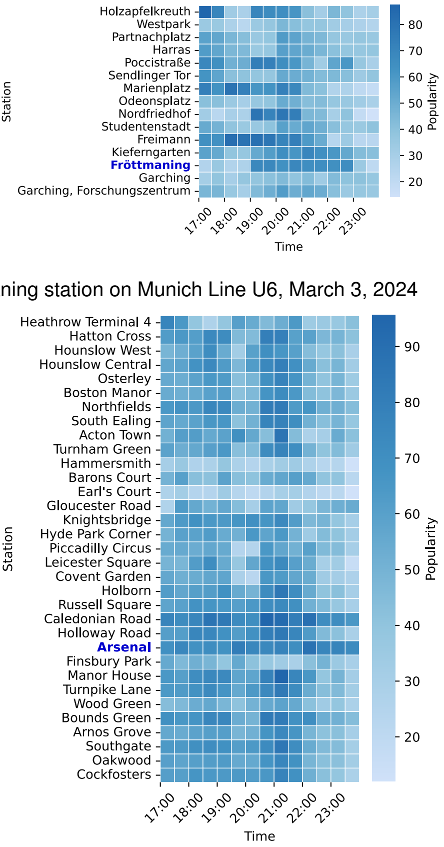
(b) Piccadilly Circus, London Line Piccadilly

(c) Fulham Broadway, London Line District

(d) Westminster, London Line District

Figure 18 Crowdedness pattern under special event scenario, London

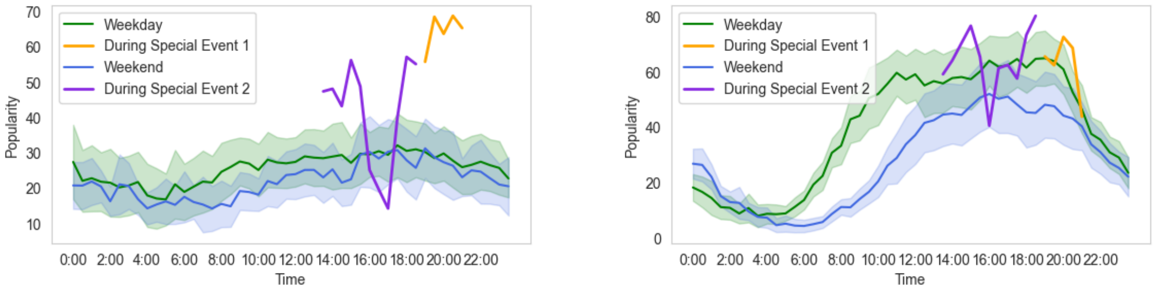
special event curves show a clear double-peak pattern, corresponding to the pre- and post-game crowdedness, perfectly reflecting the event's contribution to crowdedness at the station. Similar to previous analyses, we compared this station to another high-traffic station on the same line. Marienplatz, located in Munich's city center, typically experiences high passenger volumes. As shown in Figure 20b, Unlike other metro lines in different cities, the impact of special events on Marienplatz is evident, indicating that the influence of such events is not just dependent on the station's proximity to the venue, but also on the broader characteristics of the city's transportation network and the spatial distribution of its urban functions.



(a) Match at the stadium near Fröttmaning station on Munich Line U6, March 3, 2024

(b) Match at the stadium near Fröttmaning station on Munich Line U6, April 23, 2024

Figure 19 Spatio-temporal distribution of station popularity along urban transit lines during football matches in Munich



(a) Fröttmaning, Munich Line U6

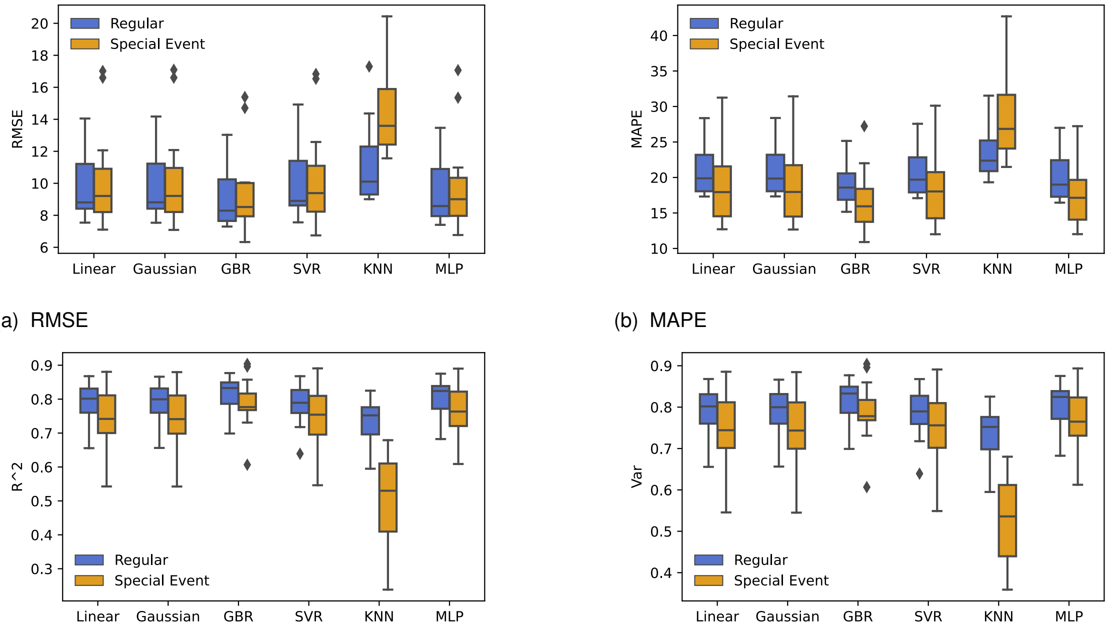
(b) Marienplatz, Munich Line U6

Figure 20 Crowdedness pattern under special event scenario, Munich

6.2. Crowdedness Forecasting Result

6.2.1. Results of Statistical Models

In this study, six distinct statistical models were evaluated for predicting station popularity under both regular and special event scenarios using data from various lines. The performance of these models is summarized in Figure 21.



(a) RMSE

(b) MAPE

(c) R^2

(d) Var

Figure 21 Performance evaluation of statistical models across different lines under regular and special event scenarios. The models evaluated are as follows: Linear = Linear Regression, Gaussian = Gaussian Process, GBR = Gradient Boosting Regression, SVR = Support Vector Regression, KNN = K-Nearest Neighbors, and MLP = Multi-Layer Perceptron

An examination of the RMSE metrics reveals that, in general, the performance metrics are slightly worse under special event conditions compared to regular scenarios. This indicates that the models tend to be less accurate in predicting popularity during special events. Specifically, RMSE values are higher in the special event scenarios, reflecting the increased difficulty in predicting station popularity when events cause atypical spikes in passenger flow.

Interestingly, the MAPE shows a different trend. Except for the KNN model, other models exhibit a decrease in MAPE during special events. This result is likely due to the higher baseline popularity values during special events, which make percentage errors appear smaller even if the absolute prediction errors are larger. Therefore, a lower MAPE does not necessarily imply better predictive accuracy in this context.

When comparing the performance of different models, it is observed that Linear Regression, GPR, GBR, SVR, and MLP models exhibit relatively similar average performance. However, GBR, SVR, and MLP models demonstrate greater stability across different lines compared to the other models. This stability can be attributed to their ability to handle complex, non-linear relationships and interactions between features, which are more prevalent across various lines.

On the other hand, the KNN model shows comparatively poorer performance, especially under special event conditions. This decline in performance may be due to the nature of the KNN algorithm, which relies on local patterns in the data. KNN's effectiveness is sensitive to the choice of the number of neighbors, and even though cross-validation was used to select number of neighbors, the model may still struggle with the high variability and non-linearity associated with special events. Moreover, KNN models can be adversely affected by the curse of dimensionality, where the performance deteriorates as the number of features increases, which might be exacerbated in scenarios with complex patterns and high-dimensional data.

Model	Scenario	RMSE	MAPE	R^2	Var
Linear	Regular	9.8144	20.6555	0.7905	0.7906
	Special Event	10.3990	18.9880	0.7491	0.7516
Gaussian	Regular	9.8312	20.6635	0.7899	0.7901
	Special Event	10.4135	19.0478	0.7485	0.7510
GBR	Regular	9.1819	18.9715	0.8161	0.8163
	Special Event	9.5513	16.7502	0.7878	0.7886
SVR	Regular	10.0132	20.3924	0.7827	0.7829
	Special Event	10.3861	18.5003	0.7484	0.7537
KNN	Regular	11.1350	23.6239	0.7331	0.7335
	Special Event	14.6723	28.9877	0.5011	0.5182
MLP	Regular	9.5041	19.8998	0.8030	0.8033
	Special Event	9.9848	17.8577	0.7696	0.7737

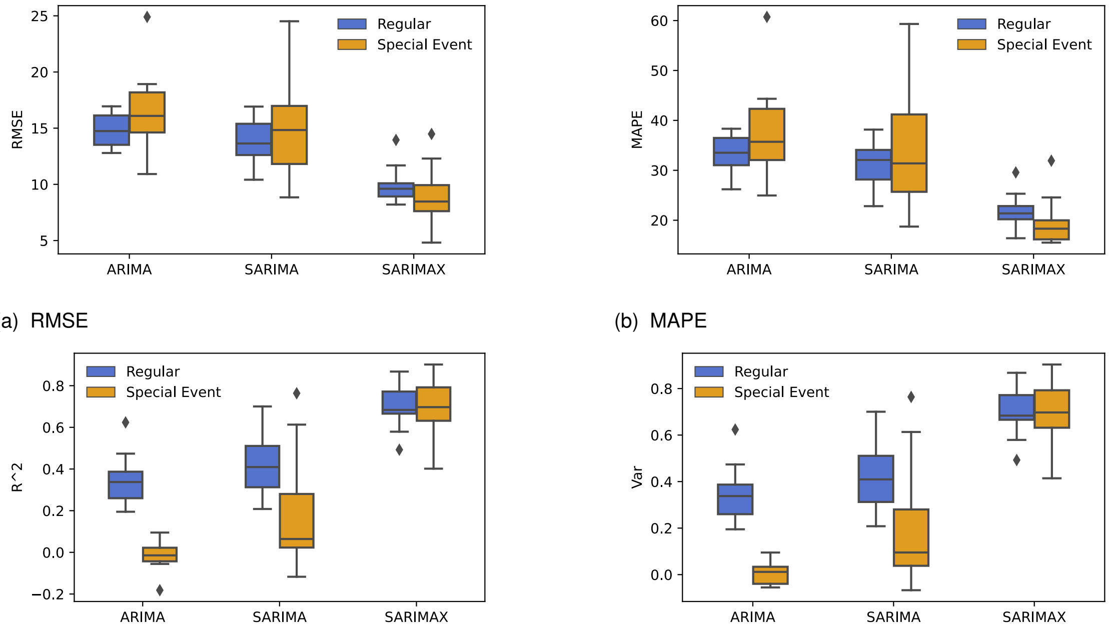
Table 11 The crowdedness forecasting performance of statistical models. 'Linear' = Linear Regression, 'Gaussian' = Gaussian Process

Table 11 summarizes the average performance metrics of all statistical models across the data from various lines. In summary, while most models show similar average performance, the stability of GBR, SVR, and MLP across different lines suggests a robust handling of diverse data characteristics. The KNN model, despite cross-validation optimization, performs less effectively, particularly in special event scenarios, highlighting potential limitations of KNN in capturing the complex dynamics of passenger popularity during such events.

6.2.2. Results of Time Series Models

Results of ARIMA models

Figure 22 provides a comprehensive overview of the performance of various ARIMA models across different scenarios. The most striking observation from the plots is the significant improvement in model performance when exogenous features are incorporated. Specifically, the ARIMA model, when applied without exogenous variables and ignoring the seasonal characteristics of the data (i.e., using only the popularity data from last 8 intervals for forecasting), shows poor performance, even in regular scenarios. This is evident from the R^2 score, where ARIMA without external data performs worse than some statistical models. For example, the linear regression model in regular scenarios achieves an R^2 of 0.7905, whereas ARIMA records a significantly lower value of 0.3487. This discrepancy is likely due to the fact that, although ARIMA models account for temporal dependencies by leveraging past data points, the statistical models benefit from a broader range of features, particularly the inclusion of event-related factors. In the case of statistical models, incorporating time-related features allows them to capture temporal patterns, mimicking the behavior of time series models in certain respects.



(a) RMSE

(b) MAPE

(c) R^2

(d) Var

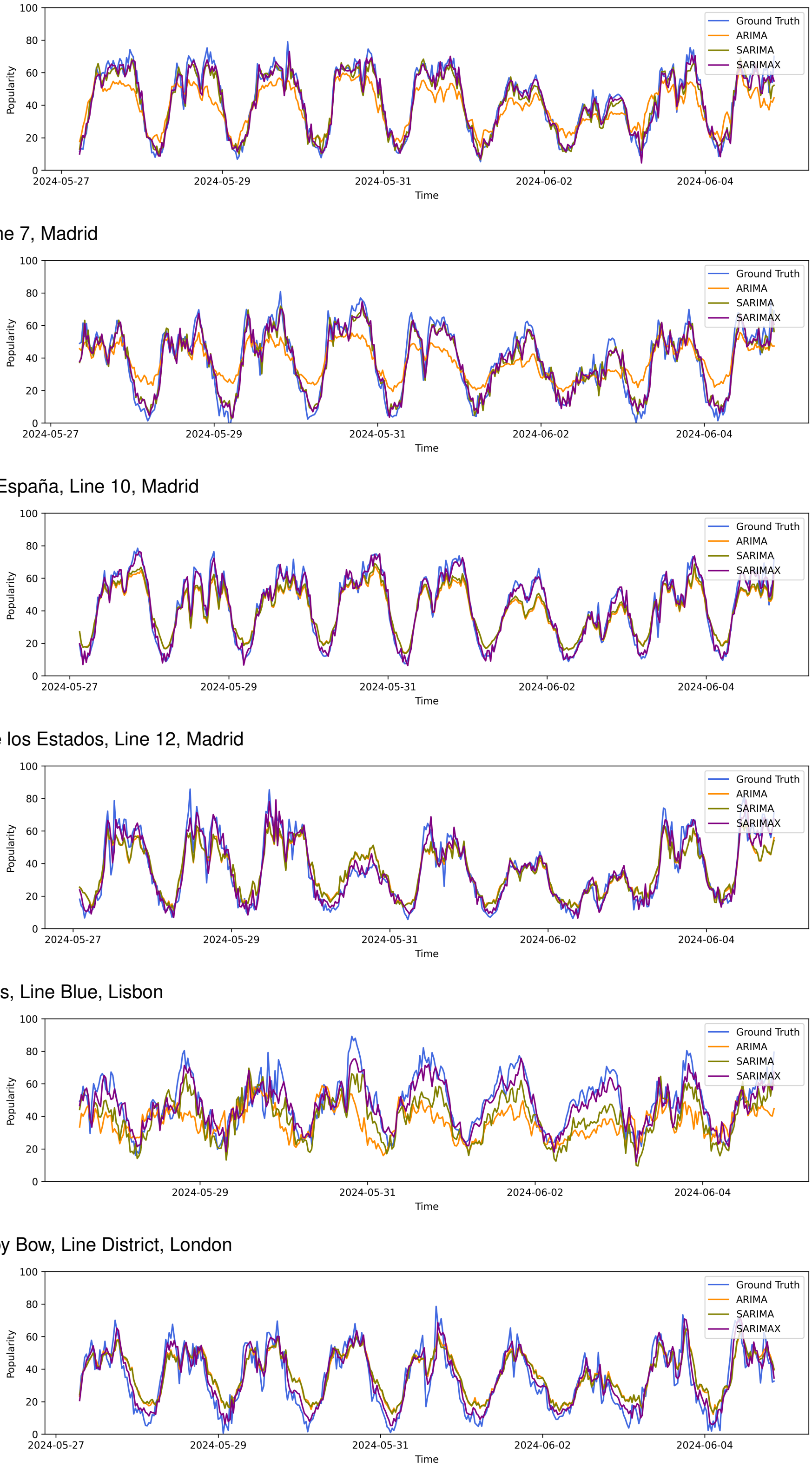
Figure 22 Performance evaluation of ARIMA models across different lines under regular and special event scenarios

When the seasonal component of the data is accounted for using the SARIMA model, the performance improves marginally. The R^2 value increases from 0.3487 to 0.4251, indicating that the model is better able to capture periodic fluctuations within the data. However, the inclusion of seasonal parameters also introduces greater instability in the predictions, particularly in special event scenarios. This suggests that while SARIMA models can capture seasonality, they may struggle to generalize under dynamic conditions such as special events.

The most notable improvement in model accuracy occurs when exogenous variables are incorporated. In regular scenarios, the R^2 jumps to 0.6999, and in special event scenarios, the model performs almost on par with regular conditions, achieving an R^2 of 0.6802. This represents a dramatic improvement compared to ARIMA without exogenous data, where the R^2 was -0.0124, and SARIMA, where the value was 0.1739. The integration of exogenous variables, such as special event data, allows the model to better capture external influences that affect station crowdedness, leading to more reliable predictions across different scenarios.

Figure 23 showcases the prediction results for selected stations across the three models. It is evident that, at some stations, the SARIMA model demonstrates a significant improvement in prediction accuracy compared to the ARIMA model. However, at other stations, the performance difference between SARIMA and ARIMA is minimal. This observation suggests that the seasonal periodicity, which SARIMA aims to capture, may not be effectively represented within the eight intervals used for prediction at these particular stations. Additionally, these patterns are not evenly distributed across different lines; within the same line, some stations show a significant improvement in prediction accuracy with the SARIMA model, while at others, there is little to no difference between the predictions of the SARIMA and ARIMA models.

Tables 12, 13, and 14 provide a detailed breakdown of the prediction performance for each model across various metro lines. Table 15 summerizes the performance of different ARIMA models.



(a) Ascao, Line 7, Madrid

(b) Plaza de España, Line 10, Madrid

(c) Parque de los Estados, Line 12, Madrid

(d) Laranjeiras, Line Blue, Lisbon

(e) Bromley by Bow, Line District, London

(f) Friedrichshagen, Line S3, Berlin

Figure 23 Crowdedness forecasting of ARIMA models at example stations

Line	Scenario	RMSE	MAPE	R^2	Var
1	Regular	13.9645	31.7307	0.3411	0.3411
	Special Event	17.9973	43.5220	0.0796	0.0796
7	Regular	15.5236	38.1982	0.3340	0.3340
	Special Event	13.9781	34.3623	0.0335	0.0335
10	Regular	16.5570	35.9993	0.2626	0.2626
	Special Event	18.9234	44.3286	-0.0310	-0.0304
12	Regular	13.7963	26.2028	0.4568	0.4569
	Special Event	12.8359	24.9617	0.0169	0.0265
blue	Regular	13.5605	33.8878	0.4738	0.4738
	Special Event	16.5007	39.9120	0.0949	0.0950
District	Regular	12.8474	31.1548	0.3438	0.3438
	Special Event	10.9183	32.6192	-0.0396	-0.0390
M2	Regular	16.1226	35.3037	0.6238	0.6238
	Special Event	17.2816	27.1737	-0.1817	0.0345
M3	Regular	16.1647	37.9375	0.1949	0.1949
	Special Event	15.3632	30.3852	-0.0427	-0.0411
Piccadilly	Regular	12.7913	30.6525	0.3252	0.3252
	Special Event	14.8333	35.5269	-0.0556	-0.0550
s3	Regular	16.9438	38.3299	0.2517	0.2517
	Special Event	18.7512	41.9150	0.0184	0.0215
u6	Regular	15.7979	29.7825	0.2132	0.2132
	Special Event	15.6858	35.9035	-0.0439	-0.0438
yellow	Regular	13.3935	33.1561	0.3639	0.3639
	Special Event	24.9022	60.7288	0.0021	0.0022

Table 12 The crowdedness forecasting performance of the ARIMA model

Line	Scenario	RMSE	MAPE	R^2	Var
1	Regular	13.8146	31.2406	0.3552	0.3552
	Special Event	17.5083	42.2013	0.1289	0.1290
7	Regular	10.4135	22.8223	0.7003	0.7003
	Special Event	8.8445	18.7300	0.6130	0.6133
10	Regular	12.0716	24.3111	0.6080	0.6080
	Special Event	9.0617	21.5190	0.7636	0.7639
12	Regular	13.7753	26.1393	0.4585	0.4585
	Special Event	12.7322	24.3392	0.0327	0.0426
blue	Regular	13.5016	33.7355	0.4783	0.4783
	Special Event	16.5479	40.8607	0.0898	0.0898
District	Regular	11.7902	28.8318	0.4474	0.4474
	Special Event	8.9124	26.5962	0.3073	0.3074
M2	Regular	16.1853	35.1385	0.6209	0.6209
	Special Event	16.8033	26.1475	-0.1172	0.1007
M3	Regular	16.0316	37.9433	0.2081	0.2081
	Special Event	14.7532	29.1532	0.0384	0.0397
Piccadilly	Regular	12.7888	30.6644	0.3254	0.3254
	Special Event	14.9170	36.0882	-0.0675	-0.0669
s3	Regular	16.9210	38.1570	0.2537	0.2537
	Special Event	18.9848	42.7159	-0.0062	-0.0029
u6	Regular	15.1781	33.0321	0.2737	0.2737
	Special Event	13.1071	33.6286	0.2712	0.2712
yellow	Regular	13.3121	32.9028	0.3716	0.3716
	Special Event	24.5154	59.3170	0.0329	0.0329

Table 13 The crowdedness forecasting performance of the SARIMA model

Line	Scenario	RMSE	MAPE	R^2	Var
1	Regular	9.6885	20.1698	0.6828	0.6828
	Special Event	10.5587	24.5702	0.6832	0.6836
7	Regular	9.6799	21.3021	0.7411	0.7411
	Special Event	7.6465	15.5209	0.7108	0.7108
10	Regular	11.3180	22.7586	0.6554	0.6554
	Special Event	8.6350	18.3654	0.7853	0.7854
12	Regular	8.2013	16.3972	0.8081	0.8081
	Special Event	8.3104	16.2276	0.5879	0.5879
blue	Regular	8.9602	21.4465	0.7702	0.7702
	Special Event	7.5169	15.9851	0.8122	0.8141
District	Regular	8.9093	20.7091	0.6844	0.6844
	Special Event	6.3709	18.4719	0.6460	0.6461
M2	Regular	9.5616	19.9540	0.8677	0.8677
	Special Event	12.2971	16.6361	0.4017	0.5237
M3	Regular	11.6871	25.3112	0.5791	0.5791
	Special Event	8.8514	15.9976	0.6539	0.6703
Piccadilly	Regular	8.9271	20.2021	0.6713	0.6713
	Special Event	7.6652	19.3084	0.7181	0.7181
s3	Regular	13.9522	29.5825	0.4926	0.4926
	Special Event	14.4860	31.9298	0.4142	0.4142
u6	Regular	8.4160	23.0897	0.7767	0.7767
	Special Event	4.8156	18.2554	0.9016	0.9033
yellow	Regular	9.6583	22.6931	0.6692	0.6692
	Special Event	9.7244	22.0117	0.8478	0.8479

Table 14 The crowdedness forecasting performance of the SARIMAX model

Model	Scenario	RMSE	MAPE	R^2	Var
ARIMA	Regular	14.7886	33.5280	0.3487	0.3487
ARIMA	Special Event	16.4976	37.6116	-0.0124	0.0070
SARIMA	Regular	13.8153	31.2432	0.4251	0.4251
SARIMA	Special Event	14.7240	33.4414	0.1739	0.1934
SARIMAX	Regular	9.9133	21.9680	0.6999	0.6999
SARIMAX	Special Event	8.9065	19.4400	0.6802	0.6921

Table 15 The summary of crowdedness forecasting performance of ARIMA models

Results of LSTM

Table 16 presents the evaluation of the LSTM model's prediction performance across the datasets used in this study. A comparative analysis reveals that when juxtaposed with the current leading time series model, SARIMAX, the LSTM model demonstrates slightly superior performance in regular scenarios. This suggests that, under typical conditions, deep learning models, such as LSTM, which leverage their capacity to uncover latent features within the data, can outperform models that rely solely on explicitly provided features.

However, this improved performance is not uniformly observed across all datasets. The LSTM model's predictive accuracy varies, with some lines exhibiting results that do not consistently surpass those of the SARIMAX model. This variability indicates that while LSTM models can capture complex patterns and relationships in data, their effectiveness may be contingent on the specific characteristics of the dataset and the presence of underlying temporal dependencies.

In special event scenarios, the performance of the LSTM model appears to be less favorable compared to the SARIMAX model, although it still outperforms both ARIMA and SARIMA models. This reduced efficacy in special event contexts may stem from several factors. Firstly, LSTM models, despite their deep learning architecture, may struggle to adequately capture and adapt to the abrupt and unique variations introduced by special events. Unlike SARIMAX, which integrates seasonal and event-specific parameters, LSTM models might not fully leverage the temporal context and event-driven anomalies. Additionally, the inherent design of LSTM networks, which focuses on learning patterns over long sequences, may not be as adept at adjusting to sudden shifts or atypical patterns that are characteristic of special events.

Overall, while the LSTM model showcases an improved performance in regular scenarios and offers improvements over traditional time series models, its performance in special event scenarios highlights the need for further refinement. This could involve incorporating additional features or hybridizing models to better capture and respond to unique and dynamic

Line	Scenario	RMSE	MAPE	R^2	Var
1	Regular	4.9046	26.2883	0.7077	0.7079
	Special Event	6.8650	57.4117	0.5269	0.5396
7	Regular	4.3651	22.2876	0.7479	0.7486
	Special Event	5.0218	33.1555	0.7503	0.7513
10	Regular	7.8632	45.4060	0.2363	0.2715
	Special Event	9.1237	165.7673	0.1730	0.1869
12	Regular	3.6801	19.0664	0.8219	0.8234
	Special Event	6.1478	15.6613	0.6262	0.6262
blue	Regular	4.8986	39.9652	0.6644	0.6917
	Special Event	6.5062	34.1016	0.5780	0.5835
District	Regular	5.2673	26.0498	0.6052	0.6054
	Special Event	5.0267	18.5231	0.7475	0.7476
M2	Regular	5.1538	14.6668	0.6068	0.6094
	Special Event	9.1255	13.9239	0.1526	0.1527
M3	Regular	5.4525	37.8743	0.6098	0.6132
	Special Event	7.7330	15.9188	0.4080	0.4276
Piccadilly	Regular	4.3985	22.4546	0.7302	0.7314
	Special Event	5.8391	17.2203	0.6596	0.6606
s3	Regular	5.9636	47.1293	0.4952	0.5055
	Special Event	6.9838	37.8180	0.5197	0.5249
u6	Regular	3.9298	34.9994	0.7685	0.7719
	Special Event	5.7855	21.6637	0.6644	0.6645
yellow	Regular	5.5479	38.8799	0.6469	0.6469
	Special Event	7.5860	28.1077	0.4213	0.4340

Table 16 The crowdedness forecasting performance of the LSTM model

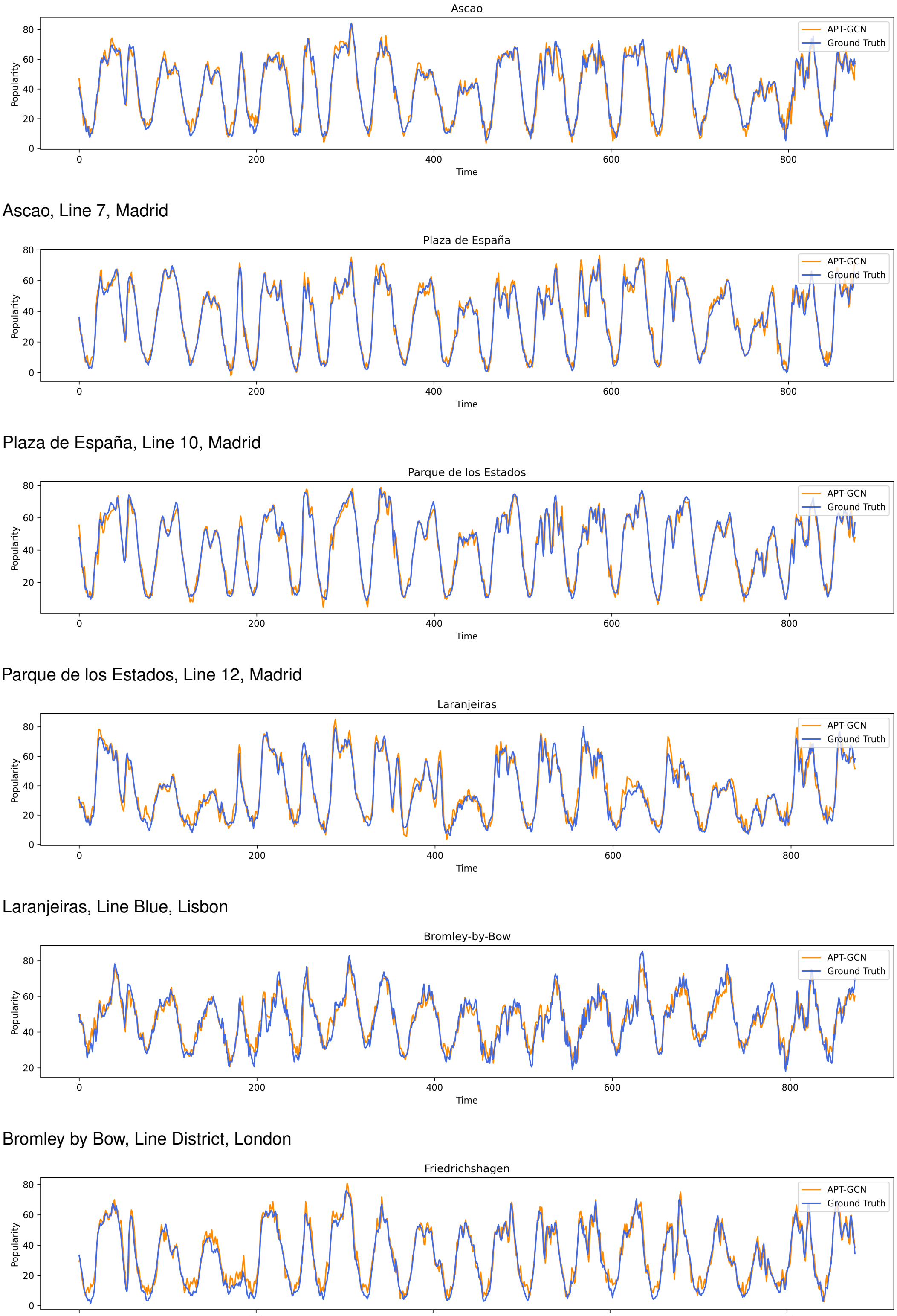
aspects of special events.

6.2.3. Results of APT-GCN

Fig. 24 illustrates the prediction results of our proposed APT-GCN model on the same stations previously shown in Fig. 23. It is evident that, with the structure specifically designed for urban rail transport passenger flow and the incorporation of tailored exogenous features, our GNN model not only leverages its capacity to uncover latent patterns within the data but also effectively integrates the provided feature set. This is reflected in the model's superior prediction accuracy and robustness when compared to alternative models.

The evaluation metrics, as presented in Table 17, clearly demonstrate the enhanced performance of APT-GCN.

Additionally, we conducted experiments on various exogenous features to assess their impact on the model's prediction capacity. The results suggest that the APT-GCN model is already well-equipped to capture the dynamics of special events without needing additional exogenous inputs.Fig. 25 compares the prediction results of the APTGCN model with and without the use of exogenous features (denoted as APTGCN-X for the former and APTGCN for the latter, where only popularity data was used for training and prediction). As seen, the difference in prediction performance between the two models is minimal, further showcasing the robustness of the APT-GCN architecture in capturing key dynamics without significant reliance on external features.



(a) Ascao, Line 7, Madrid

(b) Plaza de España, Line 10, Madrid

(c) Parque de los Estados, Line 12, Madrid

(d) Laranjeiras, Line Blue, Lisbon

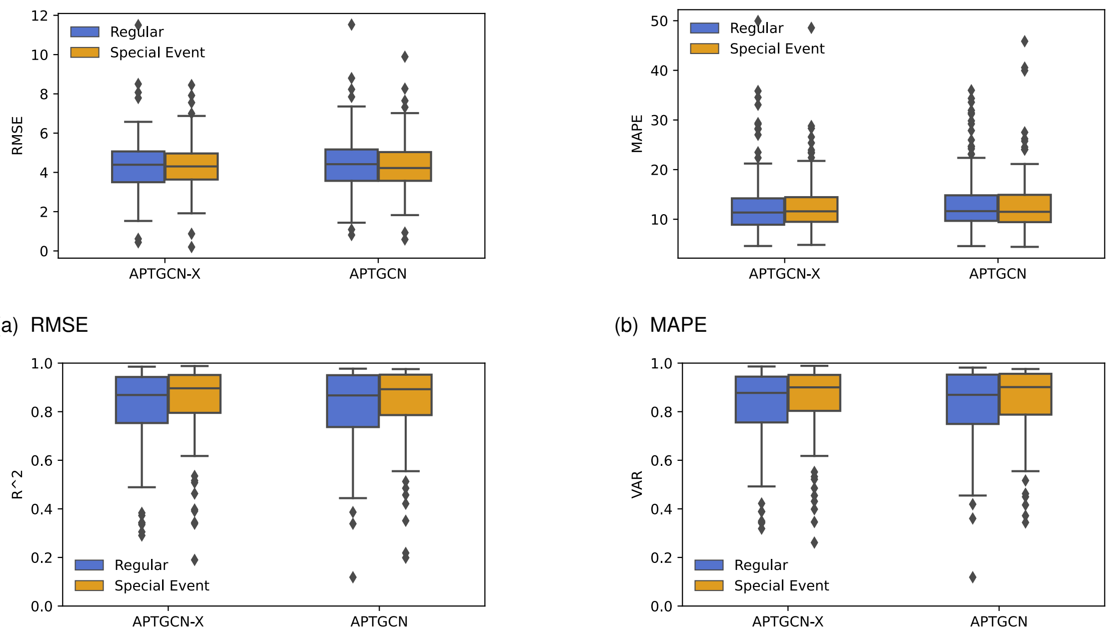
(e) Bromley by Bow, Line District, London

(f) Friedrichshagen, Line S3, Berlin

Figure 24 Crowdedness forecasting of APT-GCN model at example stations

Line	Scenario	RMSE	MAPE	R^2	Var
u6	Regular	3.8583	10.8266	0.9014	0.9015
	Special Event	4.4148	13.5528	0.8946	0.8950
s3	Regular	5.0895	11.1217	0.5707	0.5708
	Special Event	5.3321	18.3871	0.8964	0.8977
Piccadilly	Regular	6.0523	14.8218	0.7089	0.7152
	Special Event	5.2549	14.3049	0.8509	0.8512
blue	Regular	6.4483	16.5839	0.8359	0.8379
	Special Event	5.2389	18.2890	0.9079	0.9147
7	Regular	4.4556	13.0931	0.8097	0.8165
	Special Event	4.9065	11.7224	0.9186	0.9191
1	Regular	4.7332	8.3860	0.7867	0.7895
	Special Event	3.9400	8.5664	0.9421	0.9432
10	Regular	2.4463	5.0122	0.8420	0.8421
	Special Event	4.2919	18.4991	0.9348	0.9358
M2	Regular	2.6022	9.0455	0.9880	0.9888
	Special Event	2.7093	33.2585	0.9844	0.9845
M3	Regular	3.4771	9.9868	0.9677	0.9680
	Special Event	4.3448	9.9769	0.9271	0.9280
yellow	Regular	4.0616	48.5713	0.9661	0.9662
	Special Event	4.4461	10.7655	0.9266	0.9273
12	Regular	3.2794	7.4966	0.9706	0.9709
	Special Event	3.8675	9.4827	0.9516	0.9516
District	Regular	2.8969	7.5806	0.4630	0.4851
	Special Event	4.3524	12.2107	0.9018	0.9022

Table 17 The crowdedness forecasting performance of the APT-GCN model



(a) RMSE

(b) MAPE

(c) R^2

(d) Var

Figure 25 Performance evaluation of APT-GCN models under regular and special event scenarios. APT-GCN-X: APT-GCN model with special event feature as exogenous features, APT-GCN: APT-GCN model without exogenous features

6.2.4. Model Comparison and Conclusion

Finally, we summarize the performance of all models in forecasting crowdedness patterns. Our research evaluated a total of 11 models across three major categories: statistical models, time series models, and the APT-GCN model. These models were applied to datasets from 11 urban transit lines across eight European cities. The results show that traditional regression models, as well as the MLP model from machine learning, performed reasonably well. The prediction accuracy (R^2) for both regular and special event scenarios exceeded 0.75, with only a slight decrease in accuracy during special events. This may be due to the effectiveness of the event index feature 4, which successfully captured the fluctuation in crowdedness during these scenarios.

However, the performance of the KNN model was less satisfactory, particularly in the special event scenario, where the R^2 value dropped to 0.5011. We speculate that this is likely due to KNN's inherent limitations in capturing the complex, non-linear relationships present in special event data, which often involve abrupt changes in crowdedness patterns that KNN may struggle to model effectively.

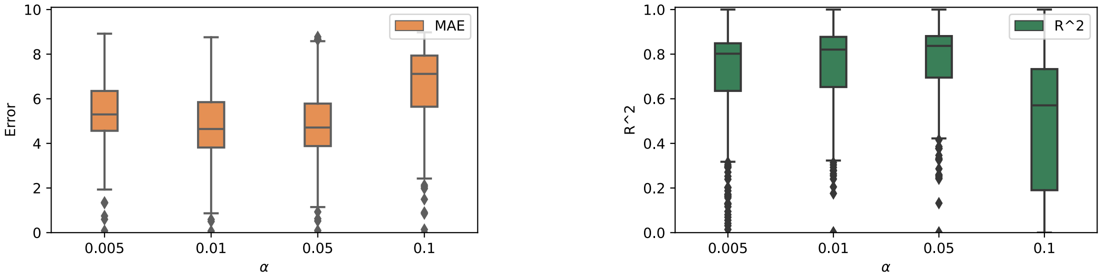
For the time series models, ARIMA and LSTM, we found that whether leveraging our provided feature set or relying on the deep learning model's ability to extract latent features from the data, the ability to uncover these underlying features proved more critical for prediction performance than solely relying on historical seasonal patterns. Specifically, the LSTM model's capacity to automatically learn complex temporal dependencies contributed significantly to its prediction accuracy, surpassing traditional time series approaches like ARIMA and SARIMA.

Finally, our proposed APT-GCN model demonstrated the best performance from all perspectives, exhibiting strong accuracy and robustness across both regular and special event scenarios. This robustness stems from the model's inherent design, which effectively captures spatio-temporal features, and does not rely heavily on additional exogenous data. The model's ability to incorporate both structural information from the transport network and dynamic temporal data enables it to outperform other models, making it a highly effective tool for forecasting crowdedness patterns in urban transit systems.

6.3. Results of Crowdedness Shift Modeling

6.3.1. Crowdedness Shift Matrix Estimation by Regression

As described in Section 4.6.1, the Lasso regression was applied to the weight data obtained from the attention layer of our APT-GCN model. This was used to estimate the crowdedness shift across the transportation network. The Lasso regression parameter α plays a key role in controlling the regularization strength, where higher values of α impose stronger regularization, potentially reducing model overfitting but at the cost of higher bias. Conversely, smaller α values allow the model to capture more complex relationships but may lead to overfitting. In this analysis, we experimented with four different values of α: 0.005, 0.01, 0.05, and 0.1. The performance evaluation of these models is presented in Figure 26, where the MAE (Mean Absolute Error) and R^2 scores are used to compare model effectiveness. From these results, it is clear that the model with $\alpha = 0.05$ offers the best performance, striking a balance between bias and variance. Therefore, the crowdedness shift matrix computed using this optimal model is utilized for further analysis.

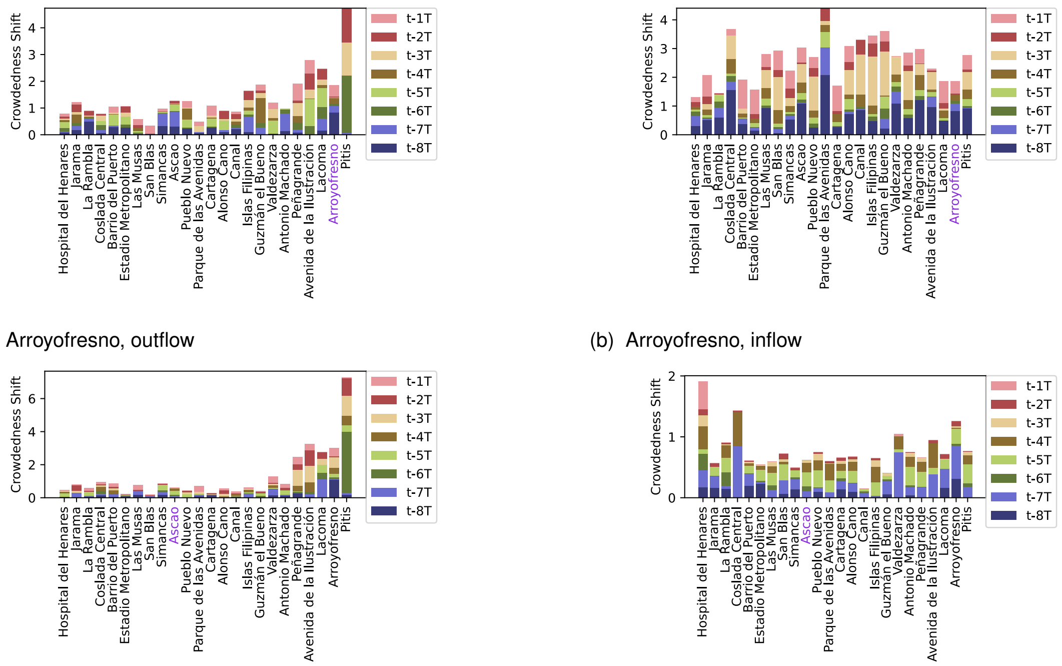


(a) MAE

(b) R^2

Figure 26 Performance evaluation of Lasso regression model for crowdedness shift modeling with varying regularization parameters (α)

Figure 27 provides an example of the crowdedness shift calculation for several stations along Madrid Metro Line 7. The shifts are categorized into two types: one representing the outflow of passenger volume from the station to others and the other showing the inflow of passengers from other stations to the given station. Each stacked bar in the figure represents the accumulated passenger flow during different intervals, with $t - nT$ indicating the amount of



(a) Arroyofresno, outflow

(b) Arroyofresno, inflow

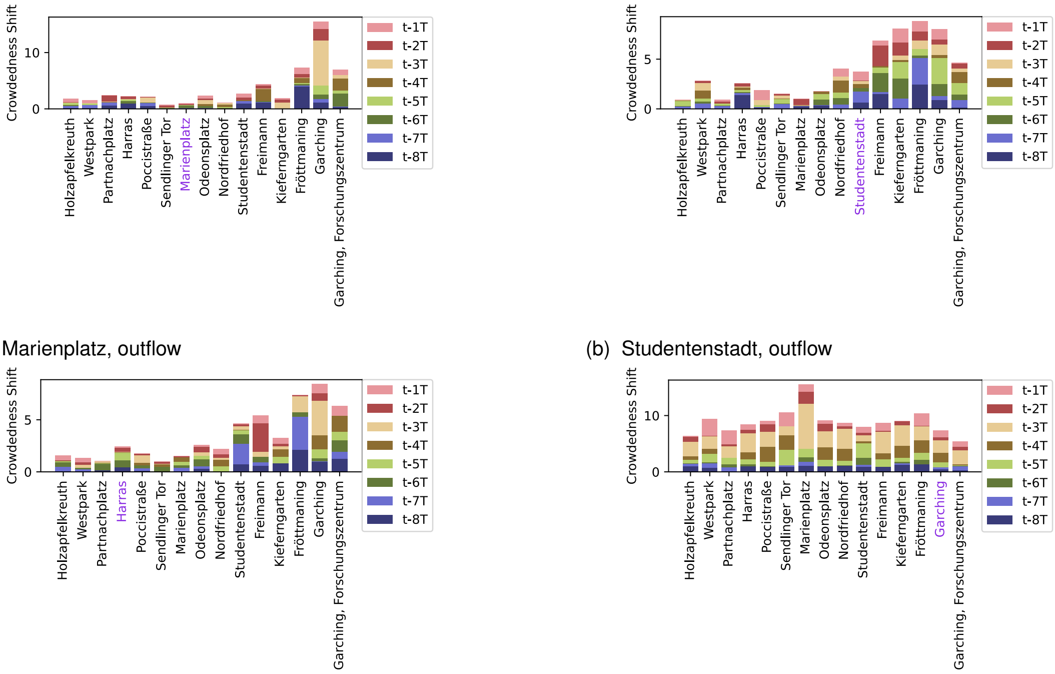
(c) Ascao, outflow

(d) Ascao, inflow

Figure 27 Crowdedness shift patterns for Arroyofresno and Ascao stations on Madrid Metro Line 7, illustrating the inflow and outflow distributions across different intervals and stations

passenger movement in the n^{th} past interval. The stations on the x-axis are ordered sequentially according to their actual layout on Line 7, with the current station highlighted in purple. It is worth noting that inflow and outflow patterns reveal a certain degree of self-flow, meaning the station itself retains some passengers. This is due to the inclusion of self-attention weights, which reflect either natural passenger growth or delayed passengers from previous intervals. Observing the outflow patterns of these two stations, it becomes evident that the distribution of passenger flow is not strictly correlated with geographical distance. Some farther stations attract more flow, indicating that attraction factors are not merely distance-dependent. For instance, the Cartagena station on Line 7, located between Parque de las Avenidas and Alonso Cano, attracts less inflow compared to its adjacent stations (Figure 27a, Figure 27c).

The crowdedness shift results could aid in identifying key stations that attract significant passenger volumes, which indicates that the location of that station is possibly a center area. Figure 28 illustrates the attractiveness of Garching station on Munich Metro Line U6. Garching, located in the northern part of Munich, serves as a commercial center for the surrounding suburban area. Comparing the passenger flow from other stations on the line, as shown in Figures 28a, 28b, and 28c, it is evident that Garching attracts a considerable volume of passengers. The inflow to Garching from other stations, as depicted in Figure 28d, is also substantial. From an urban rail network topology perspective, Garching is not a transfer station, as no other subway or urban rail lines intersect with it. This reinforces the idea that its crowdedness results primarily from its role as a local center of activity rather than from its



(a) Marienplatz, outflow

(b) Studentenstadt, outflow

(c) Harras, outflow

(d) Garching, inflow

Figure 28 Crowdedness shift modeling for Munich Metro Line U6

structural position in the transportation network.

6.3.2. Network Centrality Evaluation Based on Crowdedness Shift

Based on the crowdedness shift matrix, we further conducted a network centrality analysis, calculating four types of centralities: weighted out-degree centrality, weighted in-degree centrality, weighted eigenvector centrality, and PageRank centrality. The weight matrix used is the crowdedness shift matrix. Fig. 31 displays results for several lines. From Fig. 31b, we can further explore the hypothesis mentioned in the previous section—i.e., that the calculated passenger volume shifts can identify the stations and their surrounding areas with the most attraction potential for passengers, possibly indicating their centrality in the urban context. For example, Garching Munich shows a high weighted in-degree centrality. Interestingly, Freimann station even exceeds Garching in terms of weighted in-degree centrality. Meanwhile, both Studentenstadt and Fröttmaning exhibit significantly high weighted out-degree centrality, not only when compared to their other centrality metrics but also relative to other stations on the line.

When we turn to the Yellow Line in Newcastle (Fig. 31c), South Shields station stands out. Fig. 29 shows Newcastle's metro map, highlighting that this station serves as both a ferry and a main bus interchange. However, despite North Shields also connecting to the ferry and bus services, it does not display a notable increase in centrality. For Marseilles' Line M2 (Fig. 31a), one of the stadiums hosting special events in our study area lies between Rond-Point du Prado and Sainte-Marguerite Dromel stations. Both stations exhibit a high weighted degree centrality, with Sainte-Marguerite Dromel, the terminal station, showing a particularly

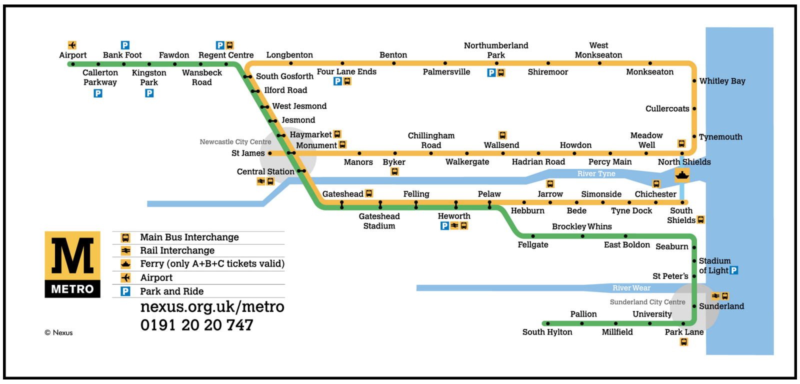


Figure 29 Newcastle metro map (Newcastle Metro, 2024)

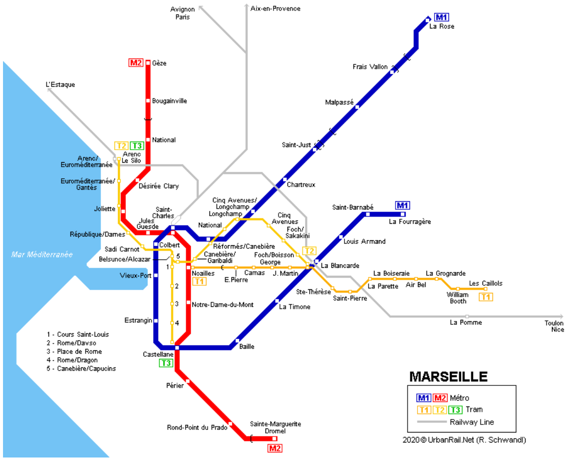
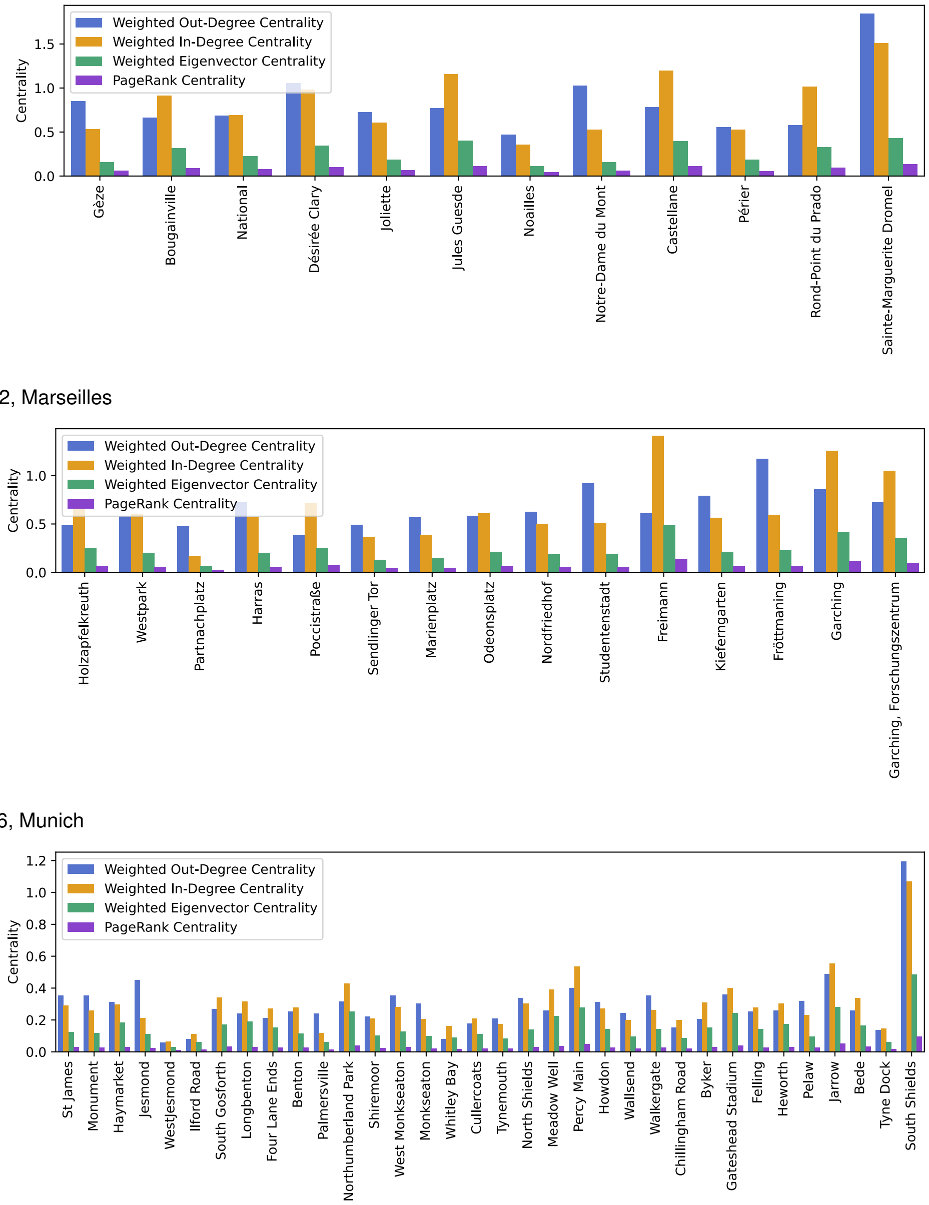


Figure 30 Marseilles metro map (UrbanRail, 2024)

high out-degree centrality. This could be explained by the fact that it integrates a major bus terminal, suggesting the station gathers demand from surrounding areas via bus services and transports it towards the city center along Line M2. Also noteworthy are the adjacent stations Notre-Dame du Mont and Castellane, both in the busy downtown commercial area. However, they display vastly different inflow and outflow characteristics: Castellane's in-degree is much higher than its out-degree centrality, while Notre-Dame du Mont exhibits the opposite. Fig. 30 shows that Castellane is a transfer station connected to Line M1, whereas Notre-Dame du Mont is not. Additionally, Castellane's eigenvector centrality is significantly higher.



(a) Line M2, Marseilles

(b) Line U6, Munich

(c) Line Yellow, Newcastle

Figure 31 Comparison of network centrality metrics across different stations on Line M2 in Marseilles, Line U6 in Munich, and Line Yellow in Newcastle

7. Conclusion and Outlook

7.1. Summary and Contribution

This thesis provides a comprehensive approach to analyzing crowdedness patterns in urban public transportation systems, particularly under regular conditions and special event scenarios. The core contributions of this work can be summarized as follows:

First, this study developed a comprehensive pipeline for constructing a dataset for crowdedness analysis, leveraging open-source data such as GPT. This pipeline addresses the challenge of data accessibility, especially in regions where public transport systems do not utilize AFC systems or do not make the data available. The use of GPT data provides an effective alternative for analyzing passenger flow in cities with limited or unavailable AFC systems (Chen et al., 2020; Carvajal and Garcia-Colon, 2003).

Second, a novel GNN-based model, the Attention-based Passenger Transport iGraph Convolutional Network (APT-GCN), was proposed. This model captures both the spatial and temporal dimensions of passenger flow along transit lines, offering improved accuracy in forecasting crowdedness patterns. By incorporating attention mechanisms into the GNN architecture, the APT-GCN can dynamically adjust the importance of different nodes (stations) based on their relationships and passenger flow interactions, which is especially crucial in urban transit networks where station interdependencies are significant.

Third, this research collected and analyzed data from multiple cities and special events, encompassing eight cities, twelve transit lines, and over 100 special events. This comprehensive dataset enabled a rigorous evaluation of the APT-GCN model across various real-world scenarios. The diversity of the dataset ensures that the model is robust and generalizable to different urban settings and event types.

Lastly, a novel method for analyzing crowdedness shifts was introduced, leveraging the attention weights from the APT-GCN model. This approach provides a potential framework for analyzing passenger flow patterns within urban transit networks, revealing how attention mechanisms can be used to detect shifts in crowdedness over time and space.

7.2. Limitations

Despite its contributions, this thesis has several limitations that should be addressed in future work:

- Limited Temporal Scope: The data collection period spans from March to June, which restricts the analysis of seasonal variations in passenger flow. A more extended period of data collection would enable a deeper understanding of how crowdedness patterns evolve over different seasons or weather conditions.

- Hyperparameter Tuning in APT-GCN: The hyperparameters of the APT-GCN model were not extensively tuned. While the model shows promising results, future work should involve more detailed experiments with different hyperparameter configurations to optimize performance.
- Crowdedness Shift Analysis: The analysis of crowdedness shifts did not separately model regular and special event scenarios. Future work could build distinct models for these scenarios to better understand how special events specifically impact crowdedness patterns.
- Simplified Weighting in Crowdedness Shift Matrix: In calculating the crowdedness shift matrix, equal weights were assigned to all previous intervals, simplifying the analysis. Future research should explore weighting schemes that account for the significance of different intervals, potentially providing a more nuanced analysis of shift patterns.

7.3. Future Research

Several areas of future research have emerged from this thesis:

- Refinement of Limitations: Future work should address the limitations identified in this thesis, particularly by collecting data over a more extended period to capture seasonal variations. Additionally, more detailed hyperparameter tuning of the APT-GCN model could yield improved performance.
- Exploring the Relationship Between Network Structure and Crowdedness Patterns: Further research could investigate whether the structure of a public transport network, station locations, and the surrounding urban environment influence crowdedness patterns. This would contribute to understanding whether certain urban configurations produce unique passenger flow characteristics (Villiers et al., 2019).
- Link Between Crowdedness Patterns and Forecasting Performance: Future studies could explore whether certain crowdedness patterns are more difficult to forecast than others, potentially identifying characteristics that challenge forecasting models. Such research would be crucial in improving model robustness.
- Transfer Learning Across Networks: The potential for applying transfer learning between different transit networks should be explored. This would allow models trained on one network to be adapted to another, leveraging shared characteristics across different urban transit systems.
- Analysis of Euro 2024 Dataset: We plan to analyze the dataset collected during the 2024 UEFA European Football Championship (Euro 2024) in Germany, focusing on special

events in host cities. The Euro 2024 dataset provides a rich source of information on special event scenarios, and analyzing it would offer new insights into how major events influence crowdedness patterns in public transportation systems.

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous systems. `https://www.tensorflow.org/`, 2015.

M. Bagchi and P.R. White. The potential of public transport smart card data. *Transport Policy*, 12(5): 464–474, 2005. ISSN 0967-070X.

R. Balcombe, R. Mackett, N. Paulley, J. Preston, J. Shires, H. Titheridge, M. Wardman, and P. White. The demand for public transport: a practical guide. Report, Transportation Research Laboratory, 2004.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 5th edition, 2015.

Brunella Caroleo, Silvia Chiusano, Elena Daraio, Andrea Avignone, Eleonora Gastaldi, Mauro Paoletti, and Maurizio Arnone. Machine learning methods to forecast public transport demand based on smart card validations. In Ana Lucia Martins, Joao C. Ferreira, Alexander Kocian, Ulpan Tokkozhina, Berit Irene Helgheim, and Svein Bråthen, editors, *Intelligent Transport Systems*, pages 194–209. Springer Nature Switzerland, 2024. ISBN 978-3-031-49379-9.

A. Carvajal and V.R. Garcia-Colon. High capacity motors on-line diagnosis based on ultra wide band partial discharge detection. In *4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003.*, pages 168–170, 2003. doi: 10.1109/DEMPED.2003.1234567.

Enhui Chen, Zhirui Ye, Chao Wang, and Mingtao Xu. Subway Passenger Flow Prediction for Special Events Using Smart Card Data. *IEEE Transactions on Intelligent Transportation Systems*, 21(3): 1109–1120, March 2020. ISSN 1558-0016.

Xinyu Chen, Zhaocheng He, Yixian Chen, Yuhuan Lu, and Jiawei Wang. Missing traffic data imputation and pattern discovery with a bayesian augmented tensor factorization model. *Transportation Research Part C: Emerging Technologies*, 104:66–77, 2019. ISSN 0968-090X.

Zhao Chen and Jia Liu. A review on graph neural networks in intelligent transportation systems. *ITS Review*, 14:235–249, 2022.

Zhanhong Cheng, Martin Trépanier, and Lijun Sun. Incorporating travel behavior regularity into passenger flow forecasting. *Transportation Research Part C: Emerging Technologies*, 128:103200, 2021a. ISSN 0968-090X.

Zhanhong Cheng, Martin Trépanier, and Lijun Sun. Incorporating travel behavior regularity into passenger flow forecasting. *Transportation Research Part C: Emerging Technologies*, 128:103200, 2021b. ISSN 0968-090X.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014a.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014b.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

Zhengyan Cui, Junjun Zhang, Giseop Noh, and Hyun Jun Park. Adstgcn: A dynamic adaptive deeper spatio-temporal graph convolutional network for multi-step traffic forecasting. *Sensors*, 23(6950): 1–18, 2023.

Luigi dell'Olio, Angel Ibeas, and Patricia Cecin. The quality of service desired by public transport users. *Transport Policy*, 18:217–227, January 2011. ISSN 0967-070X.

Alireza Ermagun and David Levinson. Spatiotemporal traffic forecasting: review and proposed directions. *Transport Reviews*, 38(6):786–814, 2018.

Ruwangi Fernando. The impact of planned special events (pses) on urban traffic congestion. *EAI Endorsed Transactions on Scalable Information Systems*, 6, jul 2019.

Scheidt & Bachmann GmbH. Open payment in bonn, germany: Bonnsmart, 2020. URL `https://www.scheidt-bachmann.de/en/fare-collection-systems/references`. Accessed on 2024-09-21.

Jay A. Goodwill and Ann Joslin. Special event transportation service planning and operations strategies for transit. Technical Report BD 459-09; NCTR 576-09, National Center for Transit Research, University of South Florida. Center for Urban Transportation Research, March 2006. URL `https://rosap.ntl.bts.gov/view/dot/38501`. Corporate Contributors: United States Federal Transit Administration, United States Department of Transportation.

Google. About popular times, wait times & visit duration data - google business profile help, n.d. URL `https://support.google.com/business/answer/6263531`. Accessed on 2024-09-21.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, November 1997. ISSN 0899-7667.

Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 207:117921, 2022.

Jian Gang Jin, Kwong Meng Teo, and Amedeo R. Odoni. Optimizing Bus Bridging Services in Response to Disruptions of Urban Transit Rail Networks. *Transportation Science*, 50:790–804, 2016. ISSN 0041-1655.

Pramesh Kumar and Alireza Khani. Evaluating Special Event Transit Demand: A Robust Principal Component Analysis Approach. *IEEE Transactions on Intelligent Transportation Systems*, 22: 7370–7382, December 2021. ISSN 1558-0016.

Sian Lun Lau and S. M. Sabri Ismail. Towards a real-time public transport data framework using crowd-sourced passenger contributed data. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, 2015.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2018.

Yang Li, Xudong Wang, Shuo Sun, Xiaolei Ma, and Guangquan Lu. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transportation Research Part C: Emerging Technologies*, 77:306–328, April 2017. ISSN 0968-090X.

Liheng Ma, Reihaneh Rabbany, and Adriana Romero-Soriano. Graph Attention Networks with Positional Embeddings. In Kamal Karlapalem, Hong Cheng, Naren Ramakrishnan, R. K. Agrawal, P. Krishna Reddy, Jaideep Srivastava, and Tanmoy Chakraborty, editors, *Advances in Knowledge Discovery and Data Mining*, volume 12712, pages 514–527. Springer International Publishing, 2021. ISBN 978-3-030-75761-8 978-3-030-75762-5.

Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.

Metro de Madrid. Metro de madrid maps, 2024. URL `https://www.metromadrid.es/en/travel-in-the-metro/metro-de-madrid-maps`. Accessed on 2024-09-21.

Alexandra Millonig, Marek Sleszynski, and Michael Ulm. Sitting, waiting, wishing: Waiting time perception in public transport. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1852–1857, 2012. doi: 10.1109/ITSC.2012.6338777.

Roberta Guglielmetti Mugion, Martina Toni, Hendry Raharjo, Laura Di Pietro, and Samuel Petros Sebathu. Does the service quality of urban public transport enhance sustainable mobility? *Journal of Cleaner Production*, 174:1566–1587, February 2018. ISSN 0959-6526.

MVV open data. OpenData, 2024.

Newcastle Metro. Metro map - newcastle, 2024. URL `https://www.nexus.org.uk/metro/maps`. Accessed on 2024-09-21.

Xiaoguang Niu, Zhen Wang, Qiongzan Ye, Yihao Zhang, and Jiawei Wang. A hierarchical-learning-based crowdedness estimation mechanism for crowdsensing buses. In *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2017.

Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

Peyman Noursalehi, Haris N. Koutsopoulos, and Jinhua Zhao. Real time transit demand prediction capturing station interactions and impact of special events. *Transportation Research Part C: Emerging Technologies*, 97, dec 2018. ISSN 0968-090X.

Agostino Nuzzolo and Antonio Comi. Advanced public transport and intelligent transport systems: new modelling challenges. *Transportmetrica A: Transport Science*, 12:674–699, September 2016. ISSN 2324-9935.

Open Street Map. Geocoding - openstreetmap wiki, 2024. URL `https://wiki.openstreetmap.org/wiki/Nominatim`. Accessed on 2024-09-21.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Marie-Pier Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568, 2011a. ISSN 0968-090X.

Marie-Pierre Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568, 2011b.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 880–887. Association for Computing Machinery, 2008. ISBN 9781605582054.

F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Trans. Neural Netw.*, 20(1):61–80, January 2009a. ISSN 1045-9227, 1941-0093.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009b.

Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. `https://www.statsmodels.org/`, 2010.

Nicolas Tempelmeier, Stefan Dietze, and Elena Demidova. Crosstown traffic - supervised prediction of impact of planned special events on urban traffic. *Geoinformatica*, 24:339–370, apr 2020. ISSN 1573-7624.

LBIP Thilakasiri, DMPM Alwis, R. T. Nanayakkara, GMRI Godaliyadda, M. P. B. Ekanayake, HMVR Herath, and Janaka B. Ekanayake. Integrated Video Based Crowdedness Forecasting Framework with a Review of Crowd Counting Models. In *2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS)*, pages 29–34. IEEE, 2021.

Transperth. Smart ticketing in public transport: The case of transperth. *Public Transport Authority of Western Australia*, 2018.

UrbanRail. UrbanRail.Net > Europe > France > Métro de MARSEILLE, 2024. URL `https://www.urbanrail.net/eu/fr/marseille/marseille.htm`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

Claude Villiers, Long D. Nguyen, and Janusz Zalewski. Evaluation of traffic management strategies for special events using probe data. *Transportation Research Interdisciplinary Perspectives*, 2: 100052, sep 2019. ISSN 2590-1982.

Eleni I. Vlahogianni, John C. Golias, and Matthew G. Karlaftis. Short-term traffic forecasting: Overview of objectives and methods. *Transport Reviews*, 24(5):533–557, September 2004. ISSN 0144-1647, 1464-5327.

Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

Gang Xue, Shifeng Liu, Long Ren, Yicao Ma, and Daqing Gong. Forecasting the subway passenger flow under event occurrences with multivariate disturbances. *Expert Systems with Applications*, 188:116057, February 2022. ISSN 0957-4174.

Jiexia Ye, Juanjuan Zhao, Kejiang Ye, and Chengzhong Xu. How to build a graph-based deep learning architecture in traffic domain: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):3904–3924, May 2022. ISSN 1558-0016. doi: 10.1109/tits.2020.3043250. URL `http://dx.doi.org/10.1109/TITS.2020.3043250`.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3634–3640. IJCAI, 2018.

Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Xucai Zhang, Yeran Sun, Fangli Guan, Kai Chen, Frank Witlox, and Haosheng Huang. Forecasting the crowd: An effective and efficient neural network for citywide crowd information prediction at a fine spatio-temporal scale. *Transportation Research Part C: Emerging Technologies*, 143: 103854, October 2022. ISSN 0968-090X.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020.

Renata Żochowska and Teresa Pamuła. Impact of traffic flow rate on the accuracy of short-term prediction of origin-destination matrix in urban transportation networks. *Remote Sensing*, 16(7), 2024. ISSN 2072-4292.