A

Thesis Report

On

Bus demand prediction using historical ticketing data Delhi

By

Jatin Bhandari

21566007

Under the Guidance of

Prof. Amit Agarwal

Mehta Family School of Data Science and Artificial Intelligence

IIT Roorkee

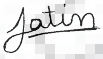
Mehta Family School of Data Science and Artificial Intelligence

Indian Institute of Technology Roorkee

May 2023

CANDIDATE'S DECLARATION

I hereby declare that the work carried out in this thesis entitled **"Bus demand prediction using historical ticketing data Delhi"** is presented in the partial fulfilment of the requirements for the award of degree of "Master of Technology" with specialization in Data Science, submitted to the Mehta family school of data science and artificial intelligence, Indian Institute of Technology Roorkee, under the guidance of **Prof. Amit Agarwal** Joint Faculty Mehta Family School of Data Science and Artificial Intelligence, IIT Roorkee. The matter embodied in this dissertation has not been submitted for the award of any other degree.



Prof. Amit Agarwal
Assistant Professor
Joint Faculty
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology, Roorkee

Jatin Bhandari
21566007

Date: May 15, 2023

Acknowledgement

I would like to express my sincere gratitude and appreciation to the following individuals who have been instrumental in the successful completion of my dissertation.

First and foremost, I would like to extend my heartfelt thanks to my respected supervisor **Prof. Amit Agarwal**, Assistant Professor, Joint Faculty, Mehta Family School of Data Science and Artificial Intelligence, Indian Institute of Technology Roorkee. For his invaluable guidance, support, and encouragement throughout my research journey. His expertise, insights, and constructive feedback have been invaluable in shaping my ideas and refining my work.

I would also like to acknowledge the contributions of **Mr. Rupam Fedujwar** and **Ms. Nishtha Rawat**, who provided me with their valuable insights, feedback, and assistance during the various stages of my research. Their input and suggestions have been extremely valuable in helping me refine my work and develop a more comprehensive understanding of my subject.

Lastly, I would like to express my deep appreciation to my family and friends for their unwavering support and encouragement throughout this challenging yet fulfilling journey.

Jatin Bhandari
(**21566007**)

Abstract

When discussing "smart cities", public transportation often comes to mind. Numerous technologies and applications have been implemented to improve the quality of public transportation in smart cities, with a particular focus on determining when buses will arrive. However, research on the prediction of crowding and occupancy on the stop level or inside the bus is limited. Accurately predicting crowding can enhance urban bus planning, and service quality and reduce operating costs.

Utilizing Electronic Ticketing Machine (ETM) data to predict crowding is more precise than using manual surveys, particularly for numerous bus routes, where balancing the accuracy and efficiency of passenger flow predictions is crucial. Merging ETM and General Transit Feed Specification (GTFS) data is necessary to gain a deeper understanding of bus scheduling and stop sequence along each route.

The study examined six months of ETM data for buses in Delhi. At first, boarding stops were inferred using the distance between consecutive stops and the time between two stops, which were calculated by the ticket's issuing time. Furthermore, alighting stops were determined using a combination of explanatory variables, such as POI data, population density data, residential areas, green areas and industrial areas.

Machine learning algorithms, including Random Forest, Extreme Gradient Boosting (XG Boost), and Artificial Neural Networks (ANN), as well as time series models like Auto Regression (AR), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA), have been used to forecast passenger flow. In this study, ANN was used to predict Stop crowding and Bus occupancy in a 15-minute time bin.

Keywords: ETM data, Boarding and Alighting inference, Public transport, ANN

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General

Considering the global trend toward prioritising public transportation. Every resident of a city wishes for good public transportation. Commuters are unsatisfied and disgruntled with the city's public services due to issues like bus delays, overcrowded buses, crowded stops and lack of public transit options, particularly during rush hours.

There is a way to make public transportation reliable and hassle-free by increasing investment in transportation infrastructure, Another way of making public transportation efficient is to do short-term bus passenger demand forecasting (Xue et al., 2015). By doing it, the bus network can be planned, and how often the buses run. It was possible to establish dynamic bus scheduling by using real-time passenger demand estimates using machine learning algorithms (Lv et al., 2022), time series (Xue et al., 2015) and neural networks(Vasconcelos et al., 2021).

To apply the above-written algorithms, the data which is needed is known as Electronic Ticketing Machine (ETM) data. ETM data provides more detailed features in the time and space dimensions than other types of data. In addition, in order to conduct an analysis of the passenger flow on a trip-by-trip basis, it is necessary to combine the data from this ETM with the data from the General Transit Feed Specification (GTFS). The following provides a short explanation of these.

1.1.1 Overview of ETM data

ETM refers to Electronic Ticketing Machines, which are utilised to give tickets to passengers in a variety of modes of transportation. When a passenger boards a bus, the conductor asks for his alighting stop and issues him a ticket using this device. The cumulative passenger data from all buses are used in this study. ETM data contains the

following attributes: The unique ticket issue number, the name of the boarding station, the name of the destination station, the date and time of the ticket's issuance, Route Number, Bus Number, Trip Number, Amount, User Count per ticket etc.

1.1.2 Overview of GTFS data

GTFS is a standard data format for storing routes, stops, fares, and schedules for public transportation. It can be used to visualize the public transportation system and identify trends in trip frequency. A GTFS feed consists of at least six or more comma-separated values (CSV) files, including Routes table, Trips table, Schedule table, Stops table, Stop time tables, agency table, etc., in which all the tables are interconnected using at least one foreign key between two or more tables using which the above visualizations can be done. The complete Database schema which is used to merge ETM and GTFS is shown in the given Fig. 1.1.

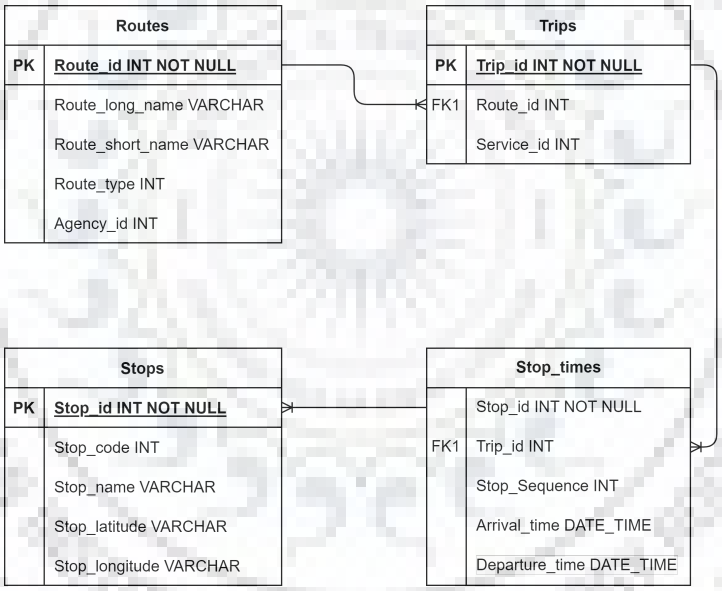


Figure 1.1: General transit feed specification (GTFS) entity relation model

1.2 Need for Study

The prediction of crowding at bus stops and inside buses is becoming increasingly important due to the growing demand for public transportation. Understanding when and where crowding is likely to occur can help transit agencies make informed decisions about scheduling and deployment of resources, ultimately leading to a better experience for riders. Additionally, predicting crowding can help improve safety by reducing the risk of overcrowding, which can lead to accidents or other safety hazards. Finally, being able to

accurately predict crowding can help transit agencies plan for future capacity needs and potentially justify additional investment in the transit system.

1.3 Objective

The main objective of predicting crowding at bus stops and inside buses is to improve the overall experience and safety for transit riders. By predicting when and where crowding is likely to occur, transit agencies can take proactive measures to mitigate the negative impacts of crowding. For example, they can adjust bus schedules, deploy additional buses or drivers, or communicate with riders in real time to inform them of potential crowding and suggest alternative travel options. Additionally, predicting crowding can help transit agencies optimize their resources and plan for future capacity needs, ultimately leading to a more efficient and effective transit system. Overall, the objective of predicting crowding at bus stops and inside buses is to enhance the quality and reliability of public transportation for all riders.

1.4 Organization of report

This seminar report includes four chapters and is organized as follows:

1. Chapter 1 introduces the topic, the need for the study, and objectives to identify crowding and the proposed methodology.
2. Chapter 2 includes the brief literature related to valuation study and the factors involved in crowding valuation, i.e., crowding representation, ways of measuring the value of crowding, modelling framework, the models used to estimate forecast, and their results.
3. Chapter 3 includes the proposed methodology for the study, the Data cleaning, wrangling, Boarding and alighting inferences and model development.
4. Chapter 4 includes the results obtained after doing the required spatio-temporal analysis and predictions of the model.
5. Chapter 5 concludes the thesis report and provides recommendations for future work.

Chapter 2

Literature Review

2.1 General

Public bus transport is an affordable and accessible mode of transportation used within a city or region. It's sustainable but can face challenges like delays and overcrowding. Nonetheless, it remains an important transportation option for many people. In Kota Bharu, Malaysia, Surveys were conducted, and results showed that passengers are dissatisfied with the service due to lack of punctuality and low frequency (Yaakub and Napiah, 2011).

Nwachukwu (2014) surveyed 300 public bus transport users in Abuja, Nigeria and found that they were also not satisfied with the service. Four factors were identified that influence passenger satisfaction: comfort, accessibility, adequacy, and bus stop facilities. Shen et al. (2016) looks at how passengers perceive bus comfort, which is affected by both the number of passengers on the bus (passenger load) and the amount of time spent on the bus (in-vehicle time). The study uses survey data to analyze the relationship between these factors and then proposes a model for evaluating comfort levels on buses.

To address the problems like crowding inside buses and crowding at a bus stop, many researchers have pointed out that forecasting in public transportation is an important thing that not only helps agencies increase their profit but also helps passengers who use it because they feel safer and more comfortable and can arrive at their destination on time. Crowding predictions can only be possible with the help of data.

2.2 Datasets used in previous studies

Researchers used various types of data to forecast passenger demand both temporally and spatially. This includes the passenger origin-destination survey, POI, Socioeconomic,

School enrolments and Data of road systems (Vasconcelos et al., 2021), Passengers ticketing data, weather data, temperature, rainfall data (Arabghalizi and Labrinidis, 2019). Examining the spatial–temporal dynamics of bus passenger travel behaviour using smart card data (Tao et al., 2014). Data sets used in the past and their merge with other data is shown in Fig. 2.1.

Passenger ticketing data is the most important one by which one can get a better understanding of passenger flow, but in most cases, this data is incomplete, i.e. the origin and destination of the tickets are not exact, so one has to correctly infer boarding and alighting stops of the passenger from the given data to get a better understanding of the passenger flow pattern. Some research is being done which finds that passenger travel is highly correlated with the POI locations, i.e. people tend to travel from these locations and their homes frequently (Lv et al., 2022). Several studies have performed temporal and spatial variability analysis on this data, which gives a brief insight into the distribution of it and how to proceed further.

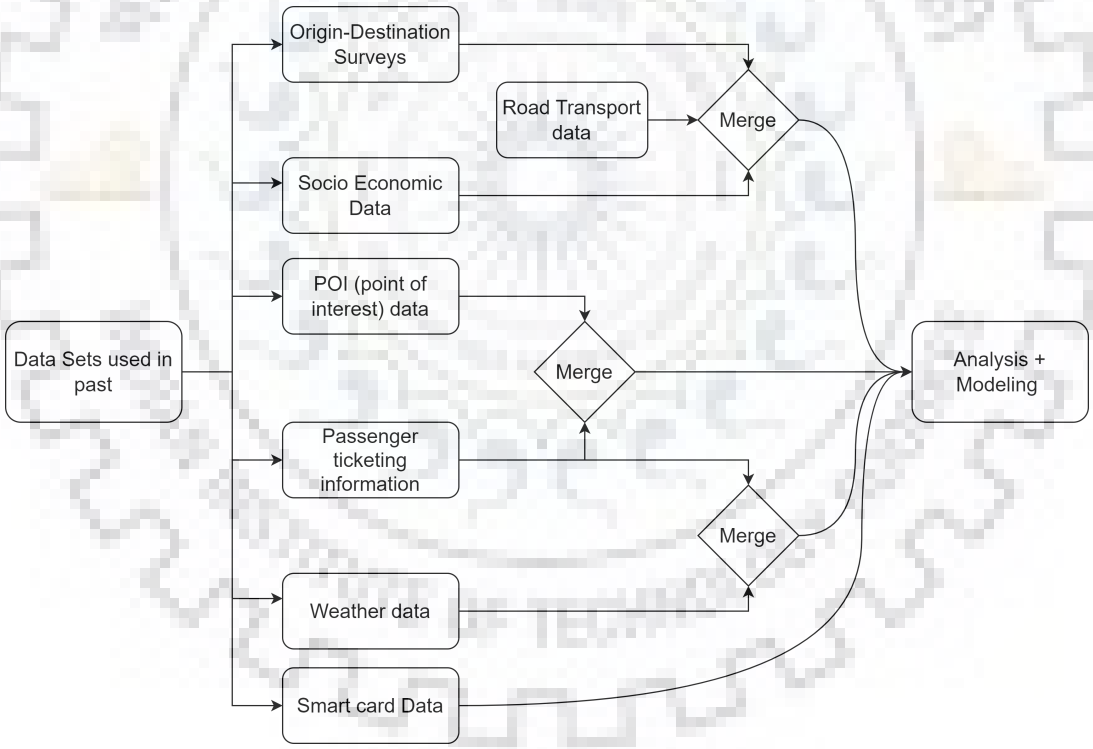


Figure 2.1: collected data and its usage

2.3 Data Analysis

The number of bus passengers during peak hours tends to be much higher than during non-peak hours. Peak hours are typically the morning and evening rush hours when

people are commuting to and from work or school. During these times, buses are often crowded, and passengers may have to stand due to the high demand for seats. In contrast, non-peak hours generally see a lower number of passengers as many people are at work or school, and there is less demand for public transportation. Overall, bus passenger ridership can vary significantly depending on the time of day and the purpose of the trip. Along with the temporal variability, the Spatial variability of bus ridership is an important factor in understanding the effectiveness of public transportation systems. This literature review examines the research that has been conducted on this topic, focusing on the factors that influence spatial variability and how it can be addressed. Land use is a major factor in determining the spatial variability of bus ridership. Areas with higher levels of residential density, commercial activity, and employment opportunities tend to have higher levels of bus ridership, as do areas with more public transit infrastructure such as bus stops and shelters (Cervero, 1997; Huang and Herman, 1996; Giuliano, 2004).

Studies have found that socio-demographic characteristics such as poverty, education, and minority populations can influence spatial variability of bus ridership, with areas having higher levels of these factors tending to have lower levels of bus ridership (Cervero, 1997; Huang and Herman, 1996; Giuliano, 2004) along with this service quality and amenities can influence bus ridership. It is also found that areas with shorter wait times, more frequent service, and comfortable seating tend to have higher levels of bus ridership (Talbott, 2011). Studies have found that pricing is also a major factor influencing spatial variability of bus ridership, with areas with lower fares and discounted fares for certain groups having higher levels of bus ridership (Talbott, 2011).

Spatial and temporal analysis on the data is performed to get insights from the data using the time stamp of the issuance of the ticket and the latitude and longitude of the boarding and alighting stop of the passenger (Tao et al., 2014), this type of analysis helps in visualizing the flow pattern on the passengers in the different time frame as well as on different locations.

Arabghalizi and Labrinidis (2019) have used the passengers ticketing information to create the extra features which would be helpful to predict the passenger flow like Bus type, Temperature, Rainfall, Snowfall etc. because these are the factors that affect passenger flow.

The spatio-temporal analysis is used to explore passenger flow patterns in public transportation systems. Several studies predicted this passenger demand using several models, enabling informed decisions to be made based on future demand. By analyzing how passenger flows change over time and space, transit agencies can optimize their

services to meet the needs of their riders better.

2.4 Models used

Data transformation plays a crucial role in building accurate predictive models, and it depends on the type of model being used. For instance, when using time series models, the data must vary only with respect to time as an independent variable (Xue et al., 2015). On the other hand, machine learning models like Random Forest and XG Boost require the creation of features that aid in prediction (Lv et al., 2022; Arabghalizi and Labrinidis, 2019). For deep learning models like Artificial Neural Networks, data with different types of features are required to build accurate models (Vasconcelos et al., 2021). Therefore, choosing the appropriate data transformation method is essential for accurate and effective predictions in different models.

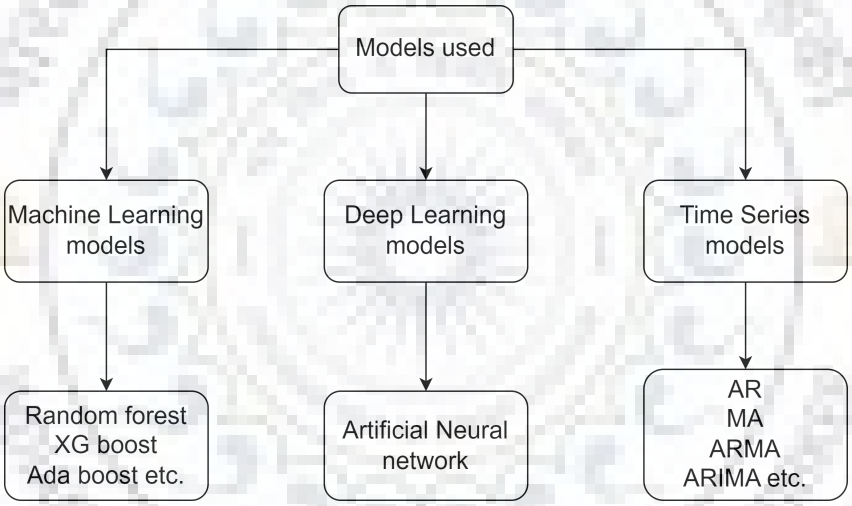


Figure 2.2: Models used

Studies have been conducted to forecast passenger demand and congestion on buses using different techniques. One approach involves transforming passenger ticketing data into time series data and applying time series models with careful consideration of the data scale and its characteristics (Xue et al., 2015). However, in situations where ticketing data is not available, surveys can be used in conjunction with socio-economic data to analyze passenger travel patterns (Vasconcelos et al., 2021). Another study proposed a win-win situation where passengers can exchange a few minutes of their time for incentives from nearby businesses in exchange for data on their arrival times at bus stops. This approach utilized a machine learning algorithm, specifically a random forest classifier, to predict the level of congestion on buses travelling along a specific route based on the number of passengers arriving at a bus stop in 15-minute time bins (Arabghalizi and Labrinidis,

2019). As the quantity of data increases, some researchers have used neural networks for predictions (Vasconcelos et al., 2021), while others have merged passenger ticketing data with point of interest (POI) data to gain better insights from the combined data (Lv et al., 2022).

2.5 Results of models from literature review

Table 2.1 presents a summary sorted in ascending order by MSE percentage, with the best results obtained for each of the architectures of the settings made. It is observed that the best architecture, among the proposals in this paper, was architecture 12 (Arq-12) indicating that the complete model with four independent variables, population, School admissions, employment, and per-capita income, is the best among them (Vasconcelos et al., 2021).

Table 2.1: Summary of the best results ordered by MSE% (Vasconcelos et al., 2021)

Architecture no.	Topology	Training rate/ Momentum	MES%	Coefficient r	R2 Score
Arq-12	[4,5,10,1]	0.9/0.9	0.045	0.487	0.228
Arq-06	[4,9,5,1]	0.5/0.5	0.05	0.288	0.083
Arq-05	[4,3,2,1]	0.9/0.5	0.053	0.209	0.043
Arq-09	[4,3,6,1]	0.5/0.9	0.053	0.408	0.230
Arq-07	[4,2,1,1]	0.9/0.5	0.057	0.145	0.021
Arq-11	[4,2,4,1]	0.9/0.5	0.061	0.031	0.001
Arq-08	[4,5,3,1]	0.5/0.5	0.064	0.419	0.175
Arq-03	[4,2,1]	0.9/0.9	0.078	0.209	0.084
Arq-10	[4,9,18,1]	0.5/0.5	0.109	0.491	0.241

For Time series models used in (Xue et al., 2015) there are a total of three models which they worked on namely 15-min, weekly and daily time difference models.

1. Weekly model in which Both ACF and PACF go to zero after a certain number of lags over the course of a day. Autoregressive moving average and moving average models are both good ways to model weekly time series because it is a stationary time series. Both are good for modelling weekly time series. It was found that the best way to choose the right order of AR(p) models was to look at different orders p [1, 6]. The adjusted R2 and the Akaike information criterion (AIC) were used to find the best order. Among all AR models, AR (3) is the best. The same test is performed on the ARMA (p, q) model (p, q) ϵ [1, 6]), the various combinations of p, q are checked. The results reveal that ARMA (2, 2) is the best of the ARMA

models. The ARMA (2, 2) model was chosen after a comparison of the two models. As shown in Fig. 2.3

ARMA (2,2) model:

$$y_w(t) = \emptyset_{w1} y_w(t-1) + \emptyset_w y_w(t-2) + \theta_w \varepsilon_w(t-1) + \theta_w \varepsilon_w(t-2) + \varepsilon_w(t)$$

Where:

y_w = prediction result of ARMA model.

ϵ_w = White noise of time series.

ϕ_w,θ_w = Parameters of weekly model

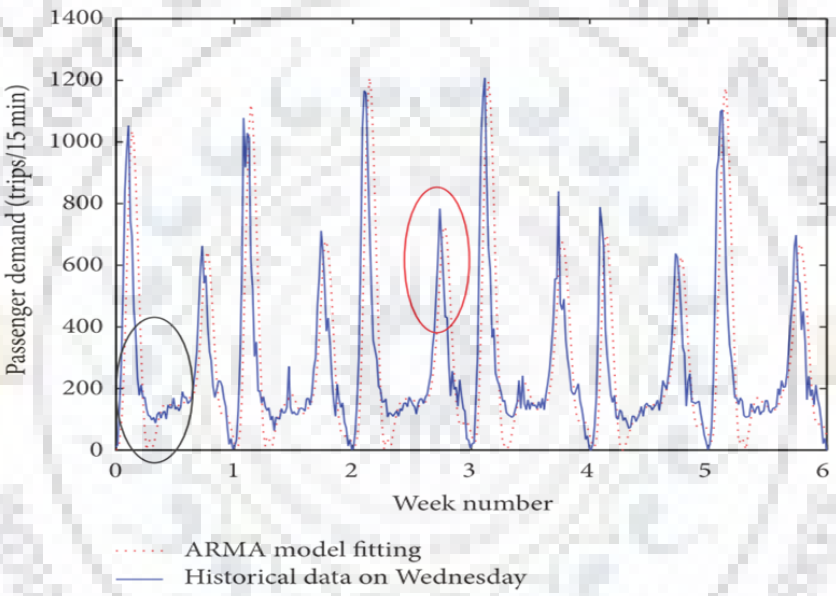


Figure 2.3: Weekly passenger demand time series (Xue et al., 2015)

2. Daily model - By comparing ACF and PACF, ACF exhibits noticeable seasonal oscillations within a range, implying that doing differencing in daily time series won't make it stationary. In Figure 6, SARIMA (p, d, q) (P, D, Q)s is used to forecast seasonal time series. The parameters p and q represent the orders of non - seasonal ARMA processes, whilst P and Q represent seasonal ones; d represents the nonseasonal differencing order and D represents the seasonal one. SARIMA(2,0,3)(1,0,0) is giving best results (Xue et al., 2015).

   SARIMA(2,0,3)(1,0,0) equation is written below:

$$\begin{aligned} y_d(i) =& \emptyset_{d1} y_d(i-1) + \emptyset_{d2} y_d(i-2) + \emptyset_{d3} y_d(i-24) + \emptyset_{d4} y_d(i-25) + \\ & \emptyset_{d5} y_d(i-26) + \theta_{d1} \varepsilon_d(i-1) + \theta_{d2} \varepsilon_d(i-2) + \theta_{d3} \varepsilon_d(i-3) + \varepsilon_d(i) \end{aligned}$$

Where "d" is written for day time series. In comparison to the ARMA model, the SARIMA model performs better, particularly during nonpeak periods, and the forecast is also improved.

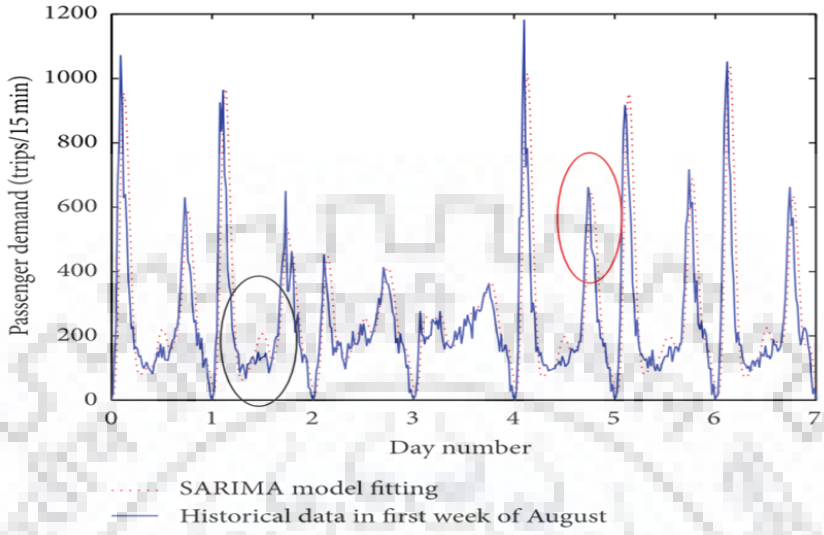


Figure 2.4: Daily passenger demand time series (Xue et al., 2015)

3. 15-min model - In a day, bus passenger demand changes a lot in 15-minute time series. There are two peaks in the morning 7:00 AM to 9:00 AM and the afternoon 4:00 PM to 5:00 PM. Based on previous research, the nonpeak passenger arrival rate is close to the Poisson distribution, which makes passenger demand more unpredictable. The ACF, PACF, and ADF tests show that the 15-minute time series is not stationary, or we can say it is not predictable by AR, MA, ARIMA models. Therefore, some changes are required to remove the volatility in the 15 min time series (Xue et al., 2015).

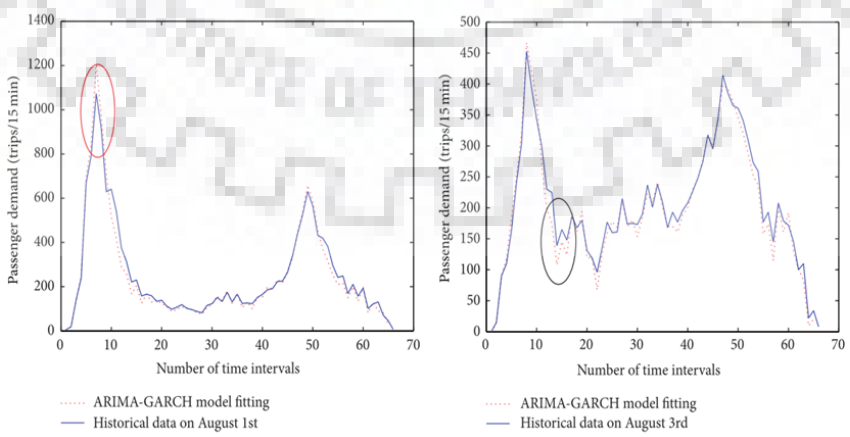


Figure 2.5: Passenger demand time series per day (15min per step) (Xue et al., 2015)

In the ARIMA model, there is a lot of heteroscedasticity and autocorrelation in the

residuals. Based on ARIMA(2, 1, 0), they try to build a more accurate 15-minute model. The ARCH effect is tested again to see if it can be used to build a more accurate model. Most people think that the GARCH(1, 1) model is most simple and robust model currently and also plotted in Fig. 2.5.

Researcher (Arabghalizi and Labrinidis, 2019) have used the machine learning model of random forest. They train it with different sets of data as shown in Table 2.2 and Table 2.3. In Table 2.2 where yes means that they are considering those data values for model building.

Table 2.2: Features to be used in classification model (Arabghalizi and Labrinidis, 2019)

Features	FS1	FS2	FS3	FS4	FS5	FS6	FS7	FS8	FS9
TOD2-TOD96	Yes	Yes	Yes	Yes	Yes	Yes	Yes		
DOW2-DOW5		Yes	Yes	Yes	Yes	Yes	Yes		
MOY2-MOY12		Yes	Yes	Yes	Yes	Yes	Yes		
Bus Type		Yes	Yes	Yes	Yes	Yes	Yes		
Rainfall, Snowfall, Temp.		Yes	Yes	Yes	Yes	Yes	Yes		
PLoad1-PLoad5			Yes					Yes	
PLoad1-PLoad10				Yes					Yes
PLoad5					Yes				
PLoad10						Yes			
PLoad5-PLoad10							Yes		

Table 2.3: Independent variables description (Arabghalizi and Labrinidis, 2019)

Variable	Description
TOD	96 variables of time of day (15min duration stamp)
DOW	Day of week has 5 variables (only weekdays)
MOY	Month of year has 12 Variables
Bus type	On bus type we have one variable (Single or Double)
Temperature	Average temperature every hour
Rainfall	Average rainfall every hour
Snowfall	Average Snowfall every hour
PLoad1-10	Bus load in the 10 previous stops (10 variables) PLoad1 is the bus stop load immediately before the current one.

The performance of these models stated in Table 2.2 is compared with the base line model Fig. 2.6 see which is a model with average loads. In this baseline, they compute the average load for each route-stop, for every 15-minute interval in a day, over the one year-long data.

They have chosen two performance metrics namely Log Loss and F1 score to evaluate the predictions coming from the baseline and the Random Forest models. Log Loss is

Figure 2.6: The average occupancy in the bus w.r.t given station name and time interval (Arabghalizi and Labrinidis, 2019)

a measure of how good probability estimates are (also known as cross entropy) as seen in Fig. 2.8. The F1 score is defined as the harmonic mean of precision and recall and is known to be more useful than accuracy if there is class imbalance in classification as seen in Fig. 2.7.

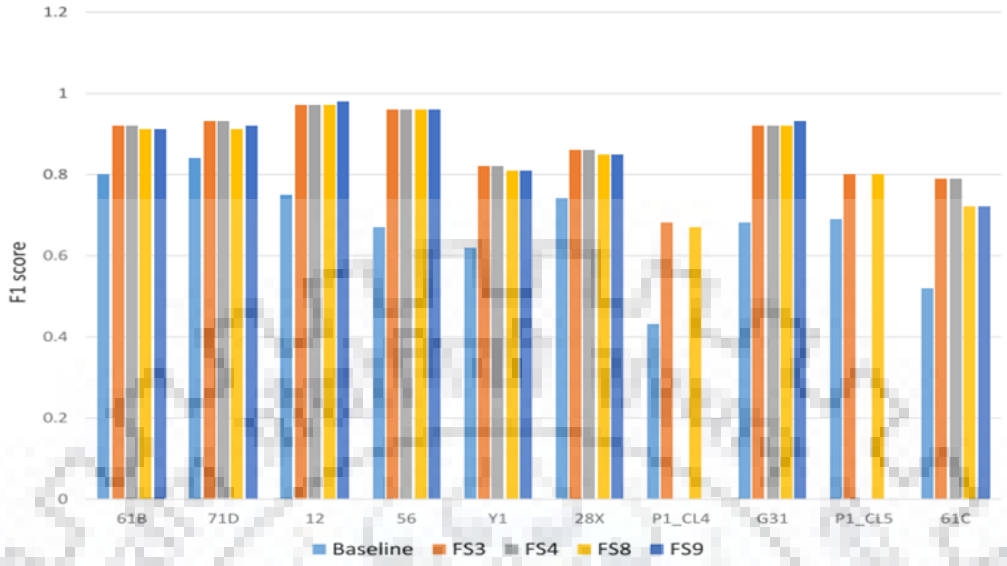


Figure 2.7: F1 Score histogram for all selected best routes (Arabghalizi and Labrinidis, 2019)

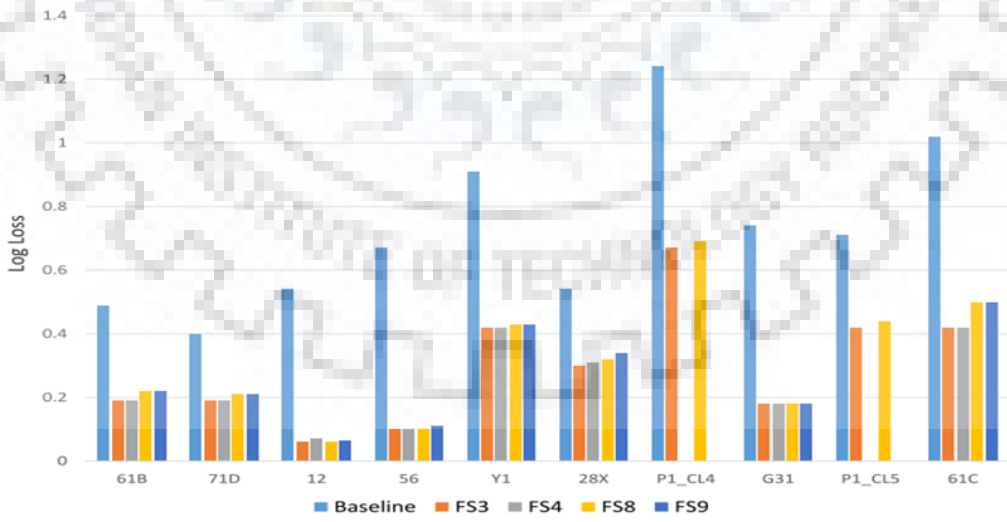


Figure 2.8: Log Loss histogram for all selected best routes (Arabghalizi and Labrinidis, 2019)

Chapter 3

Proposed methodology

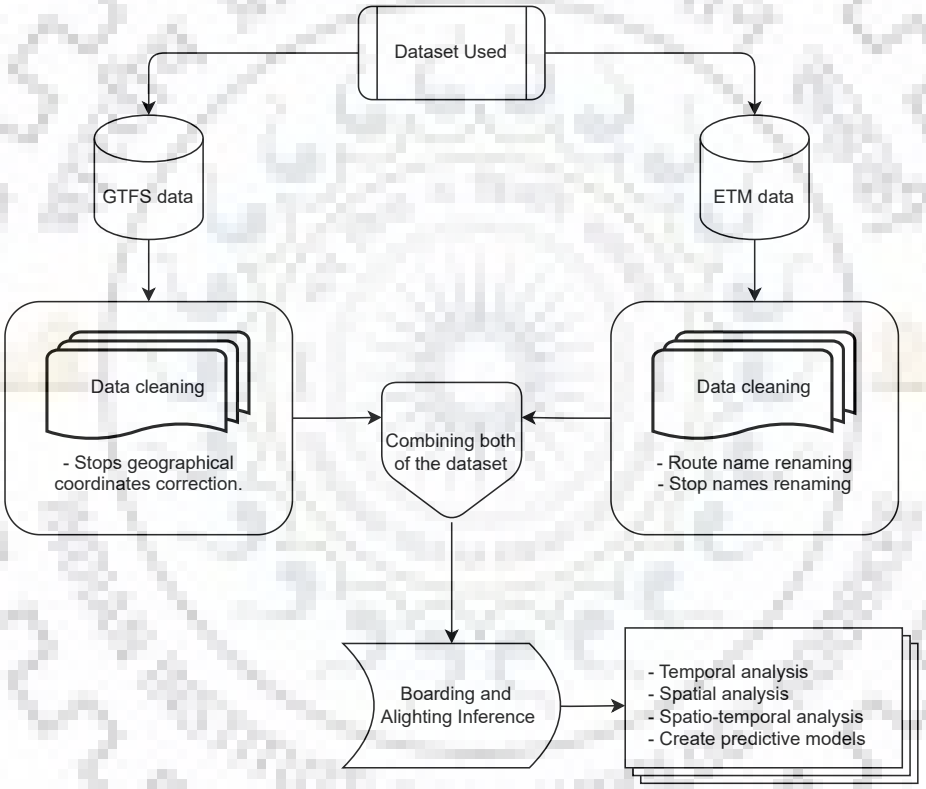


Figure 3.1: Methodology

In this study, ETM data is used to analyse and forecast bus occupancy and crowding at stop level in order to improve scheduling. To do so, ETM data is merged with GTFS data in order to sequence the route stops. It is observed that the stop names which are mentioned in the passenger tickets are not accurate (i.e. the origin name which is written on the ticket is not the one from which the passenger boarded). Only a few boarding and destination stops are selected, and tickets are issued to the passengers from those stops only, and this has to be fixed for a better understanding of passenger travel. Boarding inference was done using the arrival time of the bus at each stop, and the

alighting inference was done based on the POI, residential area, green area, industrial area and population density data. The above-mentioned data is gathered for a one-kilometer radius buffer around each stop. The whole data is then reduced to one dimension using the t-SNE (t-Distributed Stochastic Neighbor Embedding) algorithm, which includes the information of all columns.

After completing the inference stage, several spatio-temporal analyses were performed to visualize the data. Additionally, two artificial neural networks (ANNs) were trained to make predictions related to public transportation. The first ANN is capable of predicting the level of crowding at a particular bus stop within a 15-minute time interval. The second ANN predicts the occupancy levels inside a bus that is expected to arrive at a particular stop within a 15-minute time bin. This information can be incredibly useful for commuters who want to avoid overly crowded stops and buses and choose a more comfortable option and also useful for bus operators who want to optimize their routes and ensure that buses are not running at less than capacity.

3.1 Data Description

3.1.1 ETM Data

DIMTS (Delhi integrated multimodal transit system) has provided the ETM data. The data is for six months, beginning on 1 July 2019 and ending on 31 December 2019. The entire quantity of the data is approximately 24 GB, and there are 13,66,12,576 entries. In addition, the Government of Delhi has announced that women will no longer be required to pay a fare to ride any government-operated transportation after October 29, 2019, so the data on women passengers beyond that date is not available. Below are the fields present in ETM data:

1. Message Id – This is the unique id that is issued to all the tickets (all entries).
2. Route name – This includes the name of the route on which that ticket is issued.
3. Trip id – The trip id is the number that represents the different number of journeys that a bus makes along a single route. This field has a unique value for each journey across all routes.
4. Bus id – Bus number is stored in this field in which that ticket is issued.
5. Route origin stop – The stop name from which a passenger is boarding is stored in this field.

6. Route destination stop – The stop name where the passenger is going to alight.
7. Amount – This field contains the amount being paid by the passenger for that ticket.
8. User count – It includes the total number of passengers travelling on that one ticket, maximum of 5 passengers are travelling per ticket in this case and a minimum of 1.
9. Ticket Date – The date of issuance of the ticket is stored here.
10. Ticket time – The time of issuance of the ticket is stored here.

3.1.2 GTFS Data

GTFS is a data specification that allows public transit agencies to publish their transit data in a format that a wide variety of software applications can consume. The GTFS data has been briefly explained (see Fig. 1.1). Below is a description of how this data will be used in conjunction with the ETM data:

1. In this ETM data, the Route name column will serve as the foreign key and correspond to the Route long name Primary Key of the Routes table of the GTFS data.
2. After combining the Route name from ETM and Route long name from GTFS, the Route id for each route name is obtained from the Routes table itself.
3. This Route_id works as the foreign key here and corresponds to the Route_id in the Trips table. By using this relation, all the trips of every single route can be obtained.
4. Now that this trip id acts as a foreign key that corresponds to the trip id column in the Stop times table, the Stop id, and the order of stops that the bus makes during a trip can be extracted.
5. At the end for knowing the names of those stops, a Stops table is available which helps in getting the stop name for every stop_id which was obtained from the previous step.

3.2 Data Cleaning

The ETM and GTFS data need to be merged, and the first step in doing so is matching the Route name of the ETM data with the route_long_name from the Routes table in the GTFS data. During this process, it was observed that 360 Routes out of 791 from

ETM do not match the GTFS. Manual matching is done to match these route names with GTFS data. Hence, a process for matching these route names is created which is as follows:

1. Pick a route name from ETM data and go through all the stop names on which the ticket had been issued on this route.
2. In the next step, pick the route name with the same Id (there could be more than one route with the same Id but a different path, for example, 717AUP and 717BUP will have different path) one by one from GTFS and goes through all the stop names present in that route.
3. If all the stops from ETM route data are present in the stops of GTFS route data, then this route name is renamed with the picked route name from GTFS.
4. Else next route will be picked with the same number until the correct one is found. Repeat the same procedure till all the route names are matched.

The above process for one route is explained in the Fig. 3.2

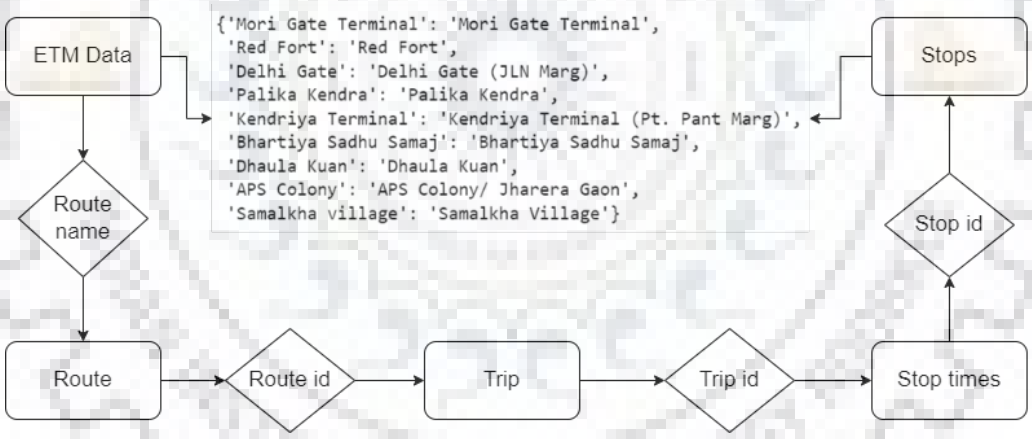


Figure 3.2: Flow for cleaning the Route and station names.

The analysis shows that some of the route names present in ETM does not match the route name presented in the GTFS, and hence those records have been dropped. After deletion, 98.3% of the records are retained for analysis

3.3 Boarding stops inference

It has been observed that the conductor issues the tickets only at certain stops along the route, not at every stop where passengers board the bus. Because of this, only 30 to 40% of the actual stops present in GTFS are listed in the ETM dataset along a route. Hence,

the following steps are proposed to infer the actual stops along a route where passengers board the bus.

1. Using the time of issuance of the first ticket issued at each stop on every journey would give the time at which the bus arrives at that stop. And the difference between the times of any two stops tells how long it took the bus to reach that stop ($T = t_2 - t_1$).
2. Further, the distance between these two stops are calculated by their latitude and longitude (latitude and longitude will be accessible using combined ETM and GTFS data) using the haversine distance formula; let's call this distance D. (it is the distance which is calculated from stop 1 to all the intermediate stops and then to stop 2, in simple words, it is the distance travel by the bus on that trip from stop 1 to stop 2.

$$D = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\emptyset_2 - \emptyset_1}{2}\right) + \cos\emptyset_1 * \cos\emptyset_2 * \sin^2\left(\frac{\gamma_2 - \gamma_1}{2}\right)}\right)$$

   Where :

   ϕ_1, ϕ_2 are the latitude of point 1 and latitude of point 2.

   λ_1, λ_2 are the longitude of point 1 and longitude of point 2.

   t_1, t_2 are time of issuance of the first ticket at stop 1 and stop 2 in a trip.
3. After determining the time and distance between two stops, the average speed is computed from stop 1 to stop 2. This is the approximate average speed at which the bus travels between those two stops.

$$Speed = D/T$$

4. After finding the average speed between those two stops, the time at which the bus would have reached the intermediate stops can be inferred, i.e. let's say there are three stops between stop 1 and stop 2, and the distance between stop 1 and all the intermediate stops is already computed. Hence the time at which the bus would have reached that intermediate stop is now known by dividing the distance by the calculated average speed. Let t_3 be the time to reach the first intermediate stop, t_4 for the second and t_5 for the third. Now the tickets can be redistributed based on t_1, t_3, t_4, t_5, t_2, i.e. tickets which are issued from t_1 to t_3 will belong to stop 1, tickets from t_3 to t_4 belong to the first intermediate stop (i.e. t_3) and so on.

5. After applying this method, about 80–90 percent of the boarding stops were inferred. This seems logical given that passengers can enter the vehicle at any of the designated stops at any point throughout the journey; nonetheless, the total number of passengers may differ depending on the type of stop and its location.

6. Validation of inferred boarding location can be done using the Real-time GTFS data.

   In Real-time GTFS data, the locations of all buses are recorded every 30 to 45 seconds. These locations are captured on all routes throughout Delhi. The time interval can be set for which the location of the buses is required. It can be used to validate the results by charting the actual journey distance and time of the bus and the time inferred; this graph must be as near to the real-time graph as possible.

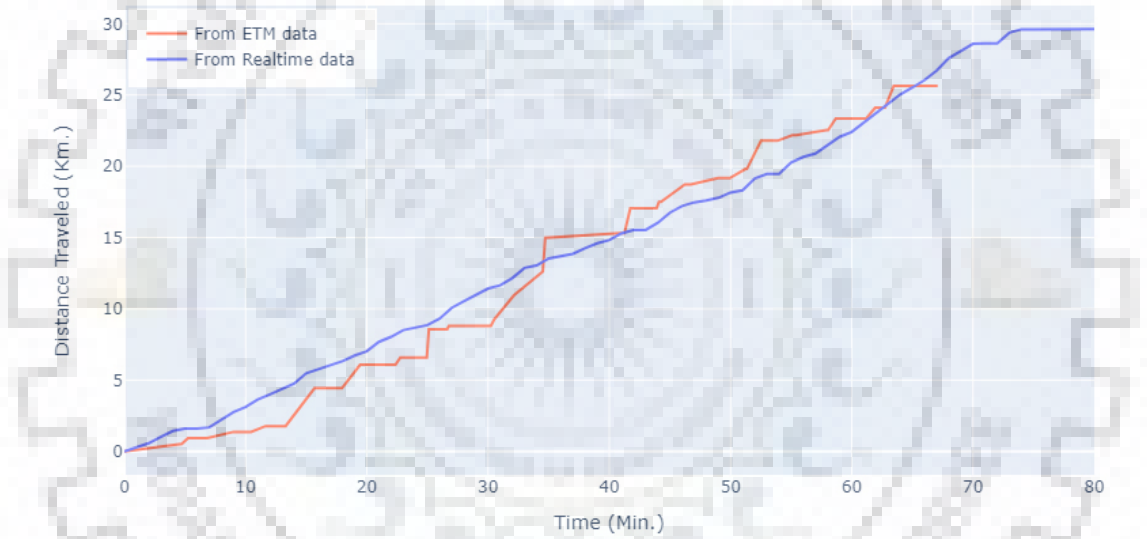


Figure 3.3: comparison of inferred bus scheduel with the real time data.

Fig. 3.3 shows the comparison between the inferred timing with the real-time data. It can be seen that the inferred timing and the actual timing of the bus are very similar and follows the same pattern.

3.4 Alighting stops inference

The alighting information of the bus passengers is present on a staged basis, i.e. just like the case seen in boarding, and hence to be inferred before further analysis. To infer alighting stops POI, residential area, green area, industrial area and population density data is used in the current study. The above-mentioned data is gathered for a one-kilometer radius buffer around each stop, which consists of the following columns:

number of POI, area of POI data, green area, residential area, industrial area, and population density data. The whole data is then reduced to one dimension using t-SNE (t Distributed Stochastic Neighbor Embedding) algorithm, which includes the information of all columns.

3.4.1 t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a machine learning algorithm used for dimensionality reduction and data visualization. It was introduced by (Maaten and Hinton, 2008) and has since become a widely used tool in various fields, including computer vision, natural language processing, and bioinformatics. Unlike traditional dimensionality reduction techniques like PCA (Principal Component Analysis) or LDA (Linear Discriminant Analysis), t-SNE is based on probability distributions rather than linear projections. This allows t-SNE to capture the nonlinear relationships between data points that may be missed by linear methods (Anowar et al., 2021).

One of the major advantages of t-SNE over other dimensionality reduction techniques is its ability to preserve local clustering structures in reduced space. This means that points that are close together in the original high-dimensional space will also be close together in the reduced space. This makes t-SNE particularly useful for tasks like cluster analysis and anomaly detection. Another advantage of t-SNE is its ability to handle complex, high-dimensional data sets that may be difficult to visualize with traditional techniques. Because t-SNE is based on probability distributions, it can effectively capture the complex relationships between data points, even in high-dimensional spaces. The t-SNE cost function can be written as follows:

$$C = \sum_{i=1}^{n} \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3.1}$$

where n is the number of data points. p_{ij} is the similarity between data points i and j in the high-dimensional space. q_{ij} is the similarity between data points i and j in the low-dimensional space, and the summation is over all pairs of data points. The similarity between two data points i and j in the high-dimensional space is defined as:

$$p_{ij} = \frac{\exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_{k \neq l} \exp(-||x_k - x_l||^2 / 2\sigma_i^2)} \tag{3.2}$$

where x_i and x_j are the high-dimensional representations of data points i and j. σ_i is a bandwidth parameter that is determined based on the local density of the data, and the denominator is a normalization term that ensures that $\sum_{j \neq i} p_{ij} = 1$. The similarity

between two data points i and j in the low-dimensional space is defined as:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{3.3}$$

where y_i and y_j are the low-dimensional representations of data points i and j, and the denominator is a normalization term that ensures that $\sum_{j \neq i} q_{ij} = 1$. The t-SNE algorithm minimizes the cost function C with respect to the low-dimensional representations y_i using gradient descent. The gradient of C with respect to y_i is given by:

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1} \tag{3.4}$$

The above equation is used to update the low-dimensional representations y_i iteratively until convergence. One of the key advantages of t-SNE is its ability to reveal structure and patterns in the data that might be hard to see using other methods. By representing the data in a lower-dimensional space, t-SNE can uncover clusters and groups of similar data points that might be hidden in the original high-dimensional space. Additionally, t-SNE is robust to outliers and can handle non-linear relationships between the data points. So the above-mentioned data is reduced to a single dimension using t-SNE algorithm and on top of it, normalization is done on stage basis (on which the tickets are issued). This gives us the overall probability ratio of the passengers for alighting at that particular stop.

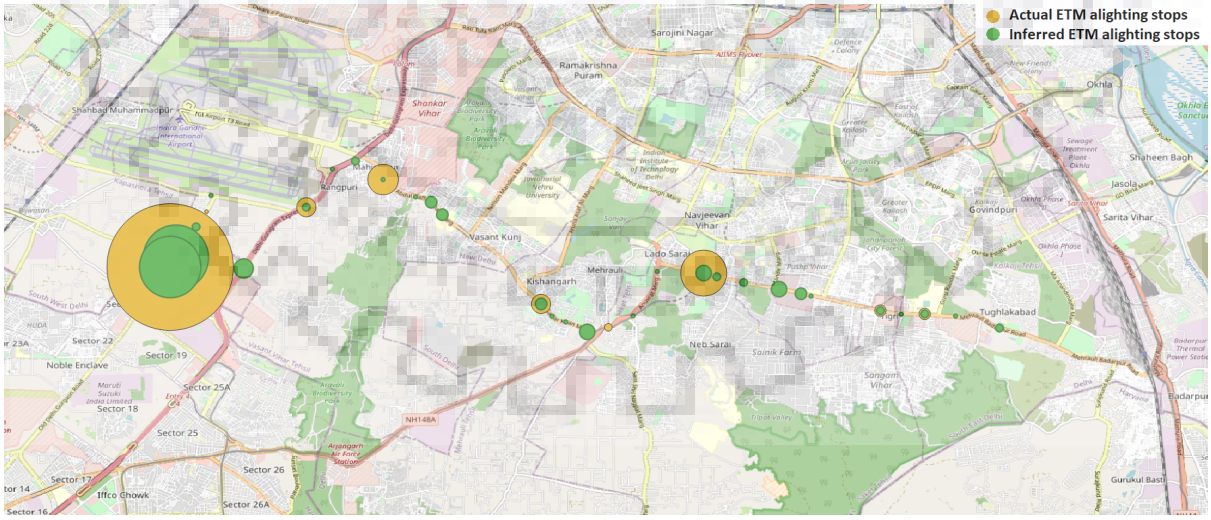


Figure 3.4: Inferred alighting stops of ETM data.

As seen in Fig. 3.4 initially in actual ETM data, the number of alighting stops on which the passengers are alighting is less, and after inference, there is a distribution of passengers in the nearby stops which are lying in their respective stages and The

size of the circle on each stop express the respective alighting count of the passengers. Inferring alighting stops can help bus agencies to optimize their services more efficiently, plan for future improvements, and make better decisions about where to invest in new infrastructure.

3.5 Model development

Predicting the level of crowding at a bus stop and the crowding on the next arriving bus is a challenging task. The level of crowding can also vary depending on the route and the frequency of service, with more frequent and popular routes experiencing higher levels of crowding. Additionally, the demographics of the surrounding area can influence ridership patterns. Factors such as time of day and day of the week can all influence the level of crowding. Historical data on ridership patterns are used to analyse and identify trends and predict future demand.

Six months of ETM data is used for the purpose of this study. Using this data set, two artificial neural networks (ANN) are designed: one for predicting crowding at each bus stop within a 15-minute time window and another to predict how full the next bus would be when it arrives at any given stop within the same window. Therefore, using these two models, it is simple to understand the crowding at bus stops and inside buses at any time of day.

3.5.1 Data preparation

To begin developing a prediction model, first, the data is transformed into the proper form so that the model can understand it and make predictions based on that. Various features have been developed that will assist in making predictions. These characteristics are as follows: Time of day, day of the week, week of the year, weekend vs weekday, Month of Year.

1. Time of Day - a 15-minute time bin for each day is created to figure out how much crowding will be at that time.
2. Day of Week - This feature allows the model to look for the relationship of crowding with any day of the week.
3. Week of Year - this feature allows the model to look at the seasonality on a monthly and yearly basis.

4. Weekend vs weekday - This allows the model to train better on weekdays and weekend data, as it has a non-linearity.
5. Month of year - This is added to deal with the shock caused by the fact that women can travel for free.

Now that the model cannot understand the text data, this data needs to be converted into numeric form so that it can be fed into the model.

One Hot Encoding

One hot encoding is a common technique used in machine learning to convert categorical data into numerical data that can be used in various models. This technique is widely used in various applications, such as natural language processing, image recognition, and recommendation systems. In this section, a discussion on what one hot encoding is, how it works, and how it can be implemented in Python is done.

What is One Hot Encoding?

One hot encoding is a process of converting categorical variables into a binary vector representation that can be used as input for machine learning models. In other words, one hot encoding is a way of representing categorical data in a form that can be easily understood by machine learning algorithms.

In one hot encoding, each category is represented by a binary vector of 0s and 1s. The length of the vector is equal to the number of categories in the variable. Each position in the vector corresponds to a category, and if an observation belongs to that category, the corresponding position in the vector is set to 1; otherwise, it is set to 0. For example, suppose a categorical variable "fruit" is present with three categories: apple, banana, and orange. This variable can be presented using one-hot encoding as follows:

Table 3.1: One hot encoded data

Fruit	Apple	Banana	Orange
Apple	1	0	0
Banana	0	1	0
Orange	0	0	1

In this representation, each observation is represented as a vector with a length equal to the number of categories. If an observation is an apple, the vector is (1,0,0). If it is a banana, the vector is (0,1,0), and if it is an orange, the vector is (0,0,1).

Using Scikit-Learn

Scikit-Learn provides a built-in class "OneHotEncoder" that can be used to implement one hot encoding. The "OneHotEncoder" class takes a categorical variable and returns a sparse matrix with a binary variable for each category. For example, suppose a numpy array with a categorical variable "fruit" is available:

```
import numpy as np
from sklearn.preprocessing import OneHotEncoder

arr = np.array(['apple', 'banana', 'orange', 'banana', 'apple'])
encoder = OneHotEncoder()
one_hot = encoder.fit_transform(arr.reshape(-1, 1))
print(one_hot.toarray())
```

In this code, first, a numpy array is created with a categorical variable "fruit". Then an instance of the "OneHotEncoder" class is created and applied to the "fruit" array using the "fit_transform" method. The resulting sparse matrix is converted to a numpy array using the "toarray" method and printed to the console.

This output Table 3.1 shows that the "fruit" variable has been one hot encoded with three binary variables "apple", "banana", and "orange". Hence, using the above method, the categorical text data is converted into numerical data and below are the number of columns (features) obtained:

1. Time of Day - 96 columns.
2. Day of Week - 7 columns.
3. Week of Year - 52 columns.
4. Weekend vs weekday - 2 columns.
5. Month of year - 12 columns.

In total, 169 features are obtained, which are used in predicting crowding using an artificial neural network.

3.5.2 ANN and its Architecture

Artificial Neural Networks (ANNs) are a type of machine learning algorithm that is inspired by the structure and function of the human brain. They are used to solve a wide

range of problems, such as image and speech recognition, natural language processing, and predictive analytics. ANNs are particularly effective in applications where traditional machine learning algorithms may struggle, such as in situations where the data is highly non-linear or where there are many variables that interact with each other.

Structure of Artificial Neural Networks

The basic structure of an ANN consists of an input layer, one or more hidden layers, and an output layer. The input layer receives data from the outside world, and the output layer produces a prediction or output based on the input. The hidden layers are where the real computation happens. Each node in a layer is connected to every node in the next layer, and each connection has a weight associated with it. These weights determine the strength of the connections between the nodes, and they are adjusted during the training process in order to minimize the error between the predicted output and the actual output.

Neurons

The nodes in an ANN are commonly referred to as neurons. Each neuron receives input from other neurons or from the outside world, and it produces an output based on a non-linear function of the inputs. The most commonly used non-linear function is the sigmoid function, which maps any input to a value between 0 and 1. The output of a neuron is then fed to the neurons in the next layer.

Activation Functions

The activation function is the non-linear function that is applied to the output of a neuron in order to produce the final output of the ANN. The most commonly used activation functions are the sigmoid function, the hyperbolic tangent function, and the rectified linear unit (ReLU) function. The choice of activation function depends on the problem being solved and the characteristics of the data.

Training an Artificial Neural Network

The weights of the connections between the neurons in an ANN are adjusted during the training process in order to minimize the error between the predicted output and the actual output. This is typically done using a method called backpropagation, which involves computing the gradient of the error with respect to each weight in the network and using this gradient to update the weight. The training process is typically repeated

many times, with each repetition called an epoch, until the network produces predictions that are sufficiently accurate.

Applications of Artificial Neural Networks

Artificial Neural Networks have found applications in a wide range of fields, including:

- Image recognition: ANNs have been used to recognize objects in images, and they have been used in facial recognition software.
- Speech recognition: ANNs have been used to recognize speech and to translate spoken language.
- Natural language processing: ANNs have been used to understand and generate natural language.
- Predictive analytics: ANNs have been used to predict the likelihood of a particular event or outcome based on past data.
- Robotics: ANNs have been used to control robots and to enable them to learn from their environment.

Batch Normalization

Batch normalization is a technique used in neural networks to improve the stability and speed of training. It involves normalizing the activations of each layer by subtracting the batch mean and dividing by the batch standard deviation.

The batch normalization formula is as follows:

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3.5}$$

where $\hat{x}$ is the normalized value of x, μ_B is the batch mean, σ_B^2 is the batch variance, and ϵ is a small constant added for numerical stability. The normalized value $\hat{x}$ is then transformed using the following parameters:

$$y = \gamma\hat{x} + \beta \tag{3.6}$$

where y is the transformed value, γ is the scaling parameter, and β is the shift parameter. These parameters are learned during training.

During training, batch normalization is applied to the output of each layer. The batch statistics (μ_B and σ_B^2) are calculated using the current batch's mean and variance

of the activations. The normalized values $\hat{x}$ are then input to the next layer. The scaling parameter γ and shift parameter β are learned using backpropagation, just like any other trainable parameter in a neural network.

During inference, the batch statistics are no longer calculated using the activations in each batch. Instead, the population statistics (i.e., the mean and variance over the entire dataset) are used. This ensures that the model's behaviour during inference is consistent with its behaviour during training.

It has several benefits:

- It reduces the effect of vanishing and exploding gradients, making it easier to train deep neural networks.
- It reduces overfitting by adding noise to the input of each layer.
- It reduces the model's sensitivity to the choice of initialization and hyperparameters.

ANN architecture for Bus occupancy prediction

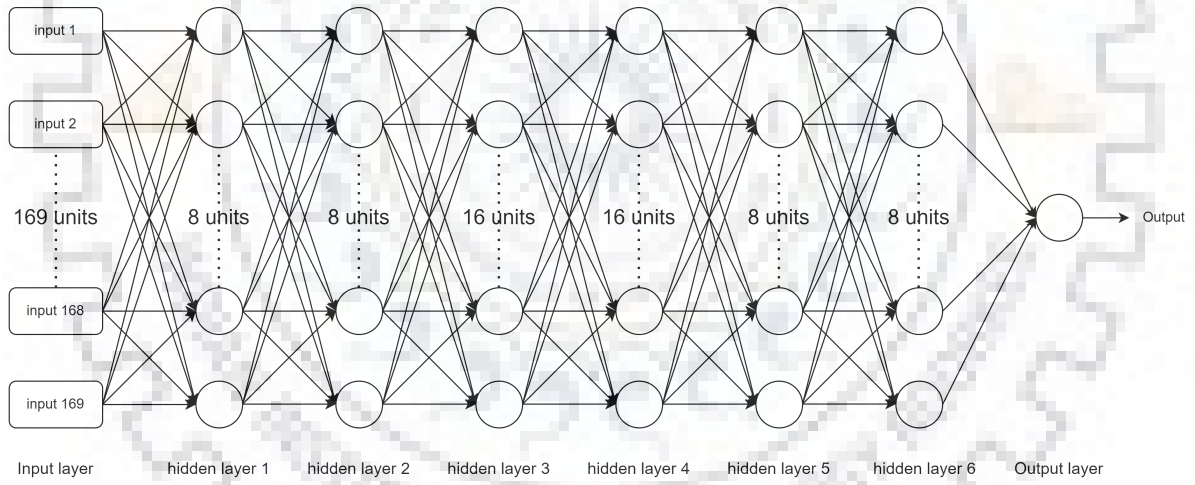


Figure 3.5: ANN architecture for bus occupancy prediction

1. Input layer: It contains 169 neurons which are nothing but the input features.
2. 1st Hidden layer: It contains eight neurons with Relu as an activation function followed by a batch normalization layer.
3. 2nd Hidden layer: It contains eight neurons with Relu as an activation function followed by a batch normalization layer.
4. 3rd Hidden layer: It contains 16 neurons with Relu as an activation function followed by a batch normalization layer.

5. 4th Hidden layer: It contains 16 neurons with Relu as an activation function followed by a batch normalization layer.
6. 5th Hidden layer: It contains eight neurons with Relu as an activation function followed by a batch normalization layer.
7. 6th Hidden layer: It contains eight neurons with Relu as an activation function followed by a batch normalization layer.
8. Output layer: It contains one neuron with Relu as an activation function.

So using the above architecture, the following is obtained:

- Total parameters: 80,969
- Trainable parameters: 80,841
- Non-trainable parameters: 128

This architecture is built to predict the Occupancy of the bus, which is going to arrive at a bus stop in 15-min time bin throughout the day in one single route, which means that a separate neural network has to build for every route. In this study, 551 models have been built (one for each route) having the same architecture Fig. 3.5 after training, validation loss and the respective predictions for some of the routes from 551 as shown in Fig. 3.6,Fig. 3.7,Fig. 3.8,Fig. 3.9.

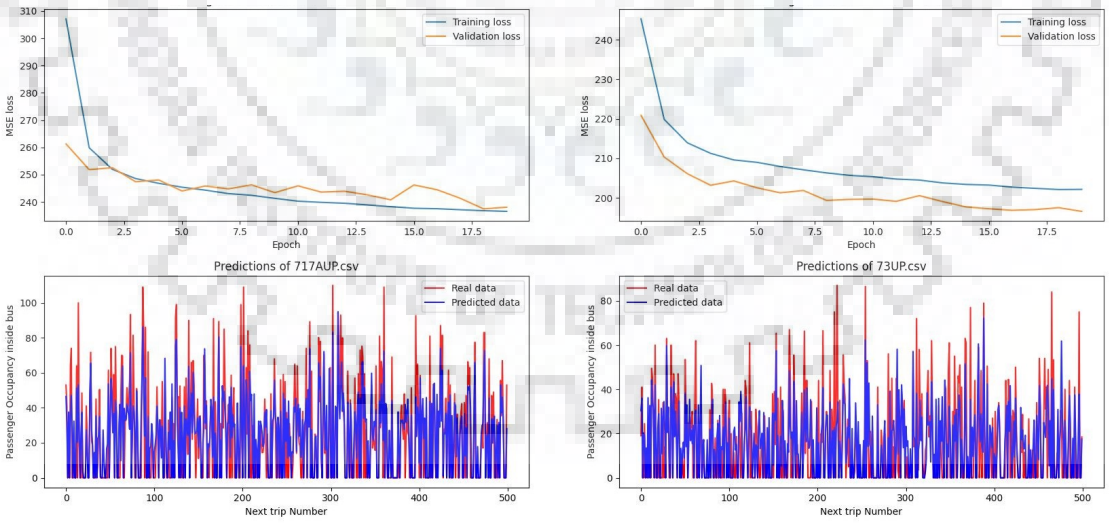


Figure 3.6: Training and validation loss with their predictions of route 717AUP and 73UP

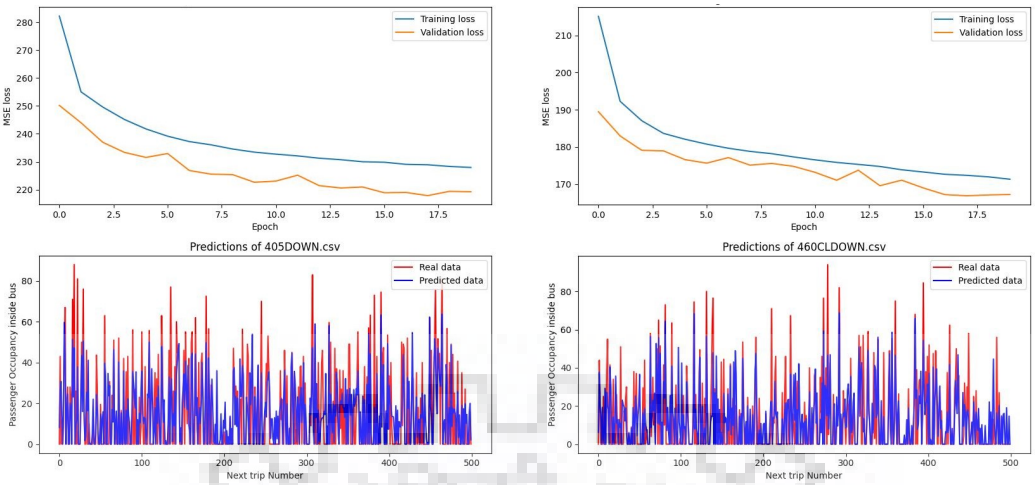


Figure 3.7: Training and validation loss with their predictions of route 405DOWN and 460CLDOWN

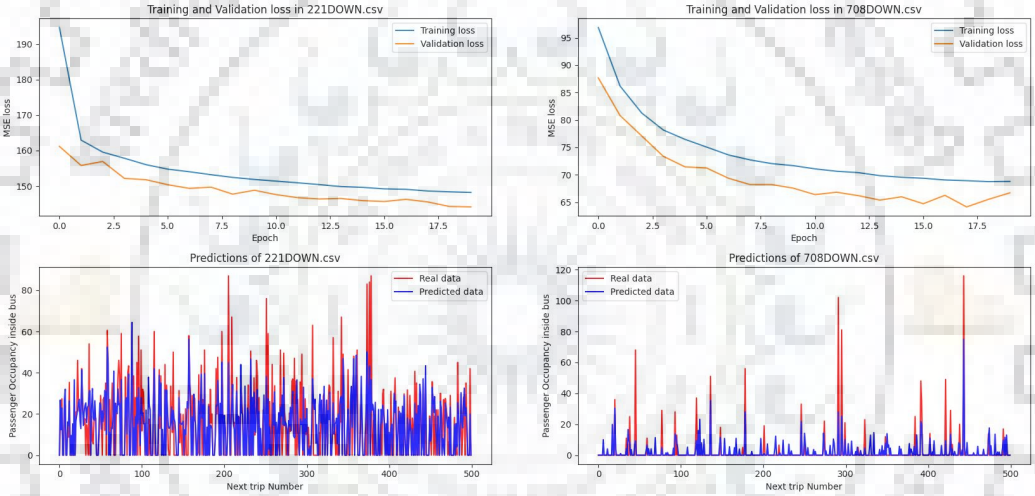


Figure 3.8: Training and validation loss with their predictions of route 221DOWN and 708DOWN

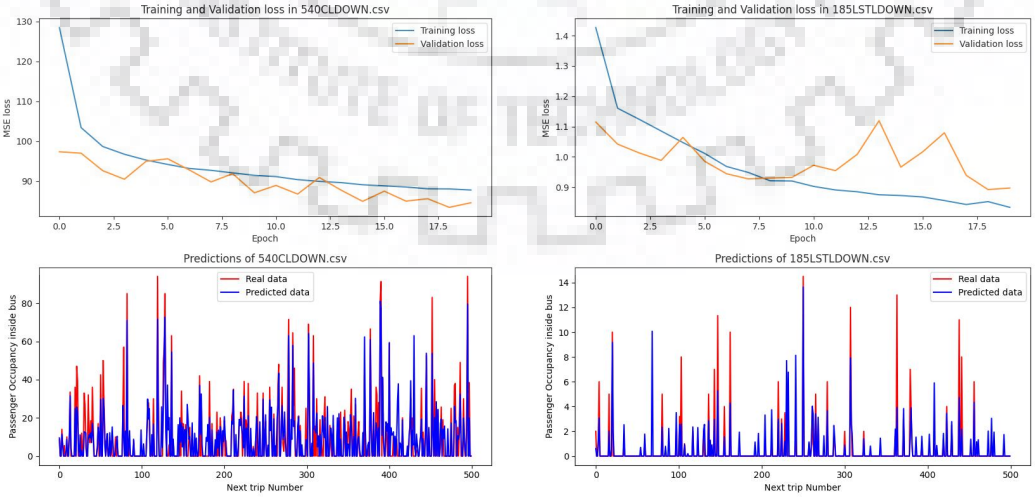


Figure 3.9: Training and validation loss with their predictions of route 540CLDOWN and 185LSTLDOWN

Network architecture for Stop crowding prediction

1. Input layer: It contains 1237 neurons which are nothing but the input features.
2. 1st Hidden layer: It contains 128 neurons with Relu as an activation function.
3. 2nd Hidden layer: It contains 256 neurons with Relu as an activation function.
4. 3rd Hidden layer: It contains 128 neurons with Relu as an activation function.
5. 4th Hidden layer: It contains 64 neurons with Relu as an activation function.
6. 5th Hidden layer: It contains 16 neurons with Relu as an activation function.
7. Output layer: It contains one neuron with Relu as an activation function.

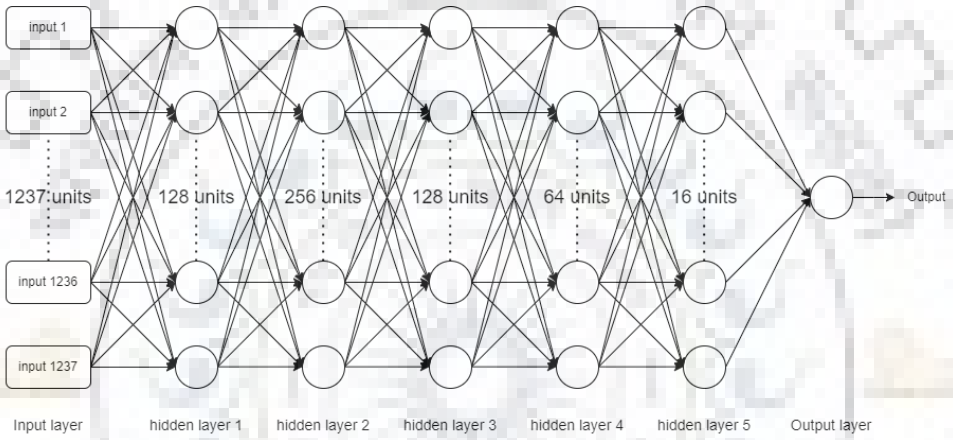


Figure 3.10: ANN architecture for stop level crowding prediction

This architecture is built to predict the Crowding at every stop throughout Delhi in 15-min time bin throughout the day, i.e. here, there is no need to build a separate model for each route. The crowding on individual stops is predicted with an accuracy of 86%. The training, validation loss and predictions are shown in Fig. 3.11

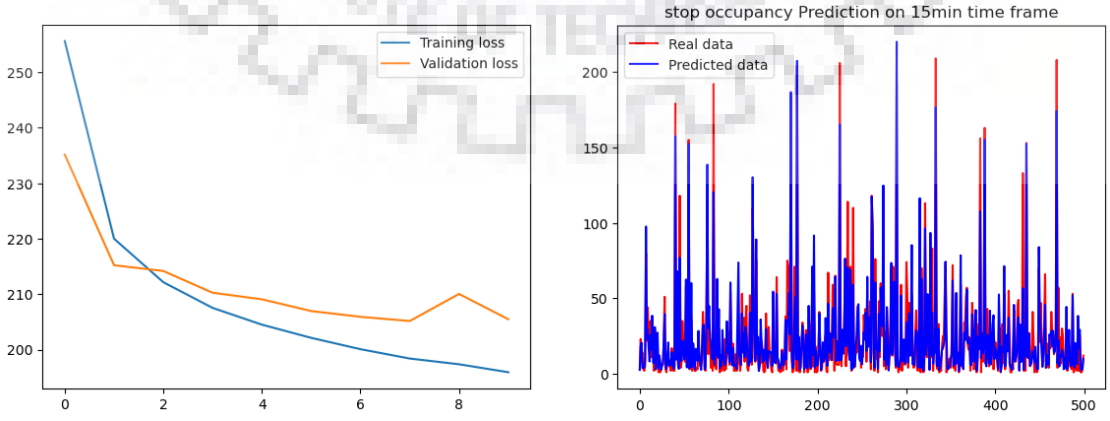


Figure 3.11: Training and validation loss with predictions

Chapter 4

Results and Discussion

Starting from November, the transportation fees for female passengers are waived off on government vehicles resulting in a monthly revenue decrease of approximately 61 million. It is observed that the largest revenue loss occurs on weekdays, which clearly states that women tend to travel less during weekends, accounting for roughly 30% of weekday revenue from female passengers see Fig. 4.1

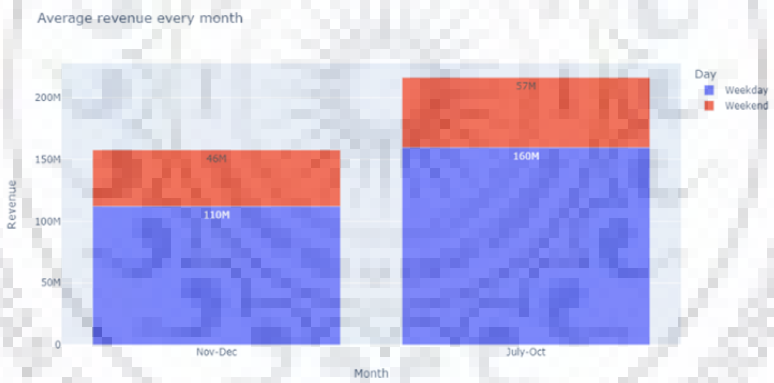


Figure 4.1: Average revenue before and after government act

It can be inferred that the elimination of transportation fees for female passengers has had a significant impact on the government's revenue. The largest revenue loss occurs during the weekdays, which may indicate that women are more likely to use government vehicles for work-related travel. The fact that there is a lower rate of female passenger revenue on weekends could mean that women are less likely to use government vehicles for leisure travel.

Analysis on passenger count on routes is also done as shown in Fig. 4.2. This phenomenon is known as the 80/20 rule or Pareto Principle, which states that 80% of the effects come from 20% of the causes. In this case, the top 20-30% of routes serve the

majority of the passenger traffic, making them the primary focus of transportation operations. It is important for transport operators to carefully manage these popular routes to ensure a high level of service, safety and comfort to their passengers. This can be achieved through regular maintenance and upgrades, implementation of new technologies and increasing frequency of services.

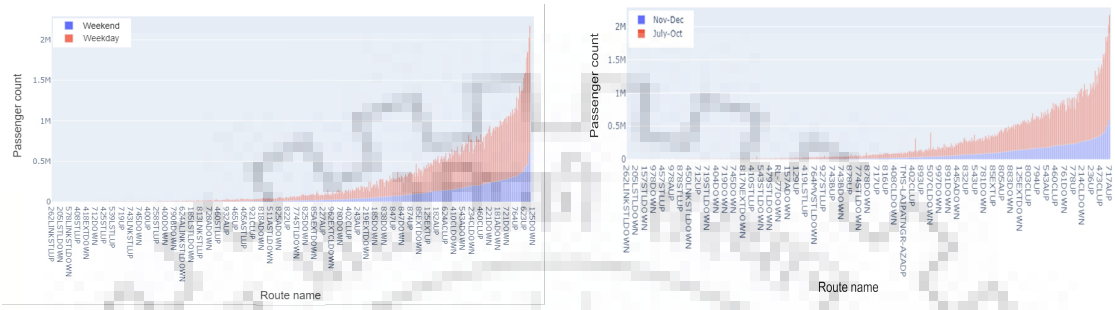


Figure 4.2: Passengers served by each route based on weekday vs weekend on the left side and before vs after government act on right side

On the other hand, the remaining routes serve only a small percentage of passengers, which raises the question of their sustainability. These routes could be consolidated or even discontinued if they consistently run at low capacity and generate low revenue. This can help transportation operators allocate their resources more effectively and improve their overall efficiency.

4.1 Data analysis

4.1.1 Temporal variability analysis

1. Fig. 4.3 displays the passengers' temporal fluctuation from July 2019 to December 2019. This variance is based on aggregating the passenger numbers hourly. After 29th October 2019, there is a noticeable decline in both the peaks i.e. the morning as well as the evening, which can be attributed to the Delhi government's announcement that women can use all public transportation for free. Consequently, no women passengers are given tickets, causing a noticeable decline.

2. Fig. 4.4 displays the passenger count was aggregated daily to determine this fluctuation. There are several modest, low peaks seen throughout the range. they are caused by Sunday, when fewer people travel. And Due to the national holiday on August 15th, there is also a lowest peak. There's one more low peak on October 27. This is because of the holiday of Diwali, which most people celebrate at home.

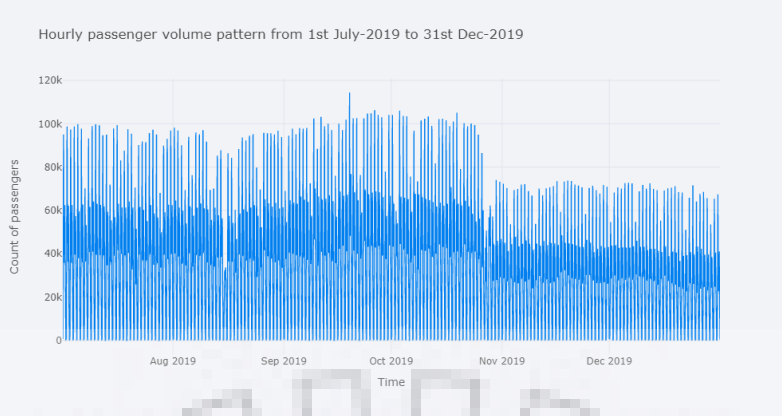


Figure 4.3: Hourly passenger volume flow from July to Dec. 2019

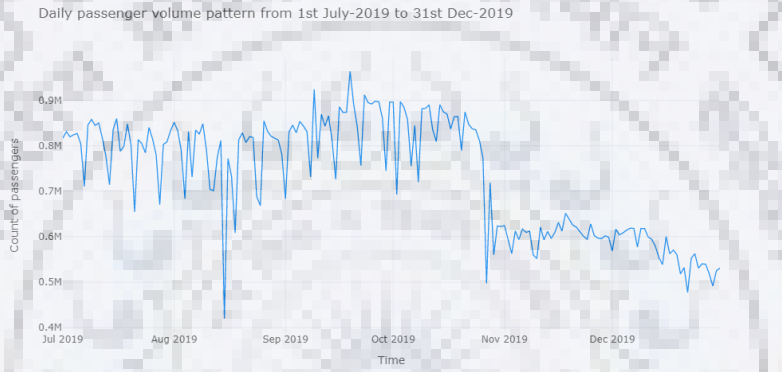


Figure 4.4: Daily passenger flow pattern from july to dec. 2019

After this holiday, on October 29, which is right after Bhai Dooj, the government makes it such that women can ride any public transportation for free.

3. Fig. 4.5 displays the passenger distribution over the hourly average for each month. The pattern of these passenger temporal variations is the same, but it can be observed that this trend shifted downward in November and December. This is because tickets aren't given to women passengers, so the total number of passengers is lower in the data, even though the number could be the same or even higher in reality. One more thing that can be figured out from the difference between the plots of November and December and the rest of the months is that the time variation of female passengers follows the same distribution as that of male passengers. Because the difference between these plots is larger during the peak hours and lower during the non-peaks hours.

4. Fig. 4.6 shows the hourly average passenger temporal fluctuation across each day of the week. It can be observed that the pattern is similar from Monday to Saturday, but different on Sunday. The fact that there are no offices, schools, or coaching

facilities open on Sunday may be one reason why the morning peak is lower on Sundays. The evening peak is following the same pattern, People used to move out for enjoying the weekend could be one of the reason, People also came back to their homes from other cities after going out in weekend to start work the next day.

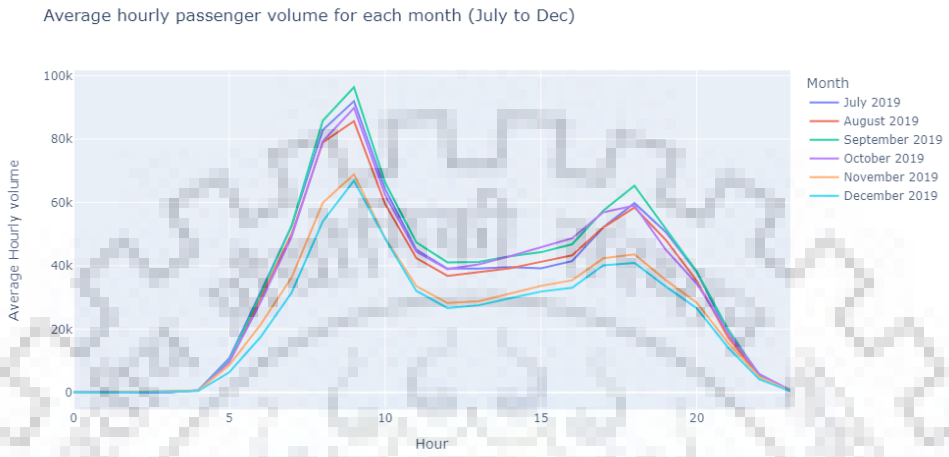


Figure 4.5: Average hourly passenger flow pattern for each month.

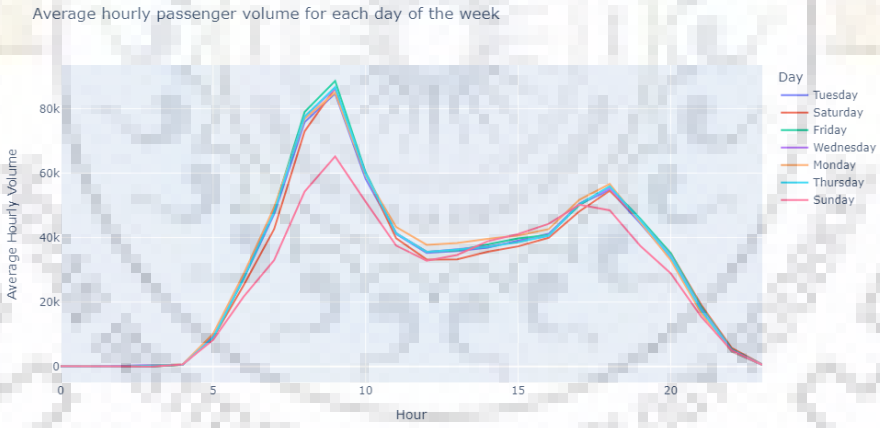


Figure 4.6: Average hourly passenger flow pattern for each Day of week.

4.1.2 Spatial variability analysis

Spatial variability analysis is an essential tool for understanding the flow of bus passengers. This type of analysis involves examining how passenger traffic varies across different parts of a bus network or route and identifying the key drivers of these variations. Factors that may contribute to spatial variability in bus passenger flow include demographics, land use patterns, POI etc.

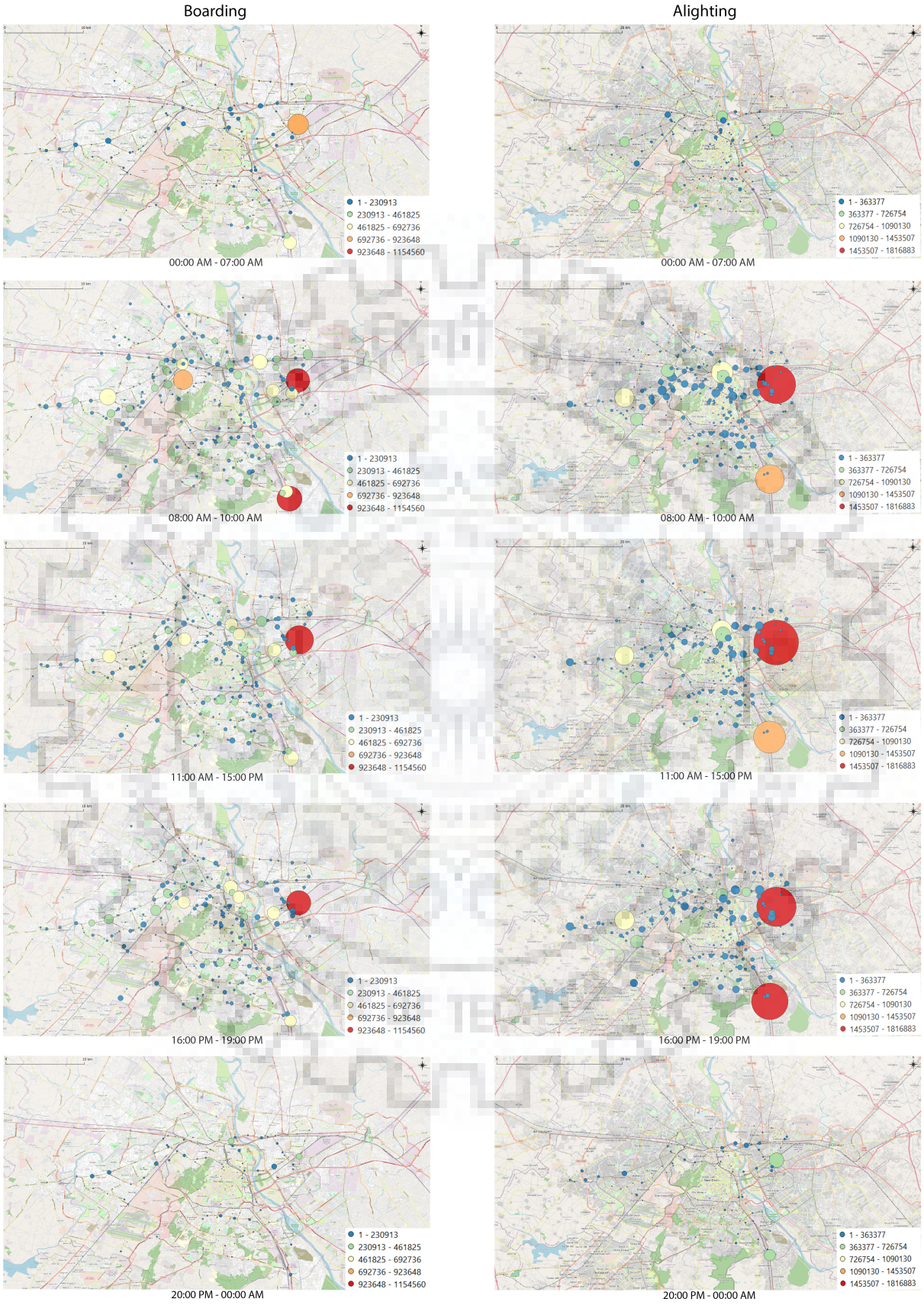


Figure 4.7: Passenger boarding and alighting based on time and region

The Boarding and alighting of passengers at different locations in different time frames are seen in Fig. 4.7. During peak hours in the morning (i.e. 8-11 AM) it is seen that people used to board from many locations spread over the whole Delhi and move towards the inner circle of the city and two specific locations (Red circles), which are the Railway station and Interstate Bus terminal of the city. It is observed that the passenger will travel back to its original location at the end of the day Tao et al., 2014. A similar situation is being observed in Delhi as people are alighting towards the outer city during the evening. One more fact that can be seen here is that people are not boarding from the Badarpur border (the bottom right Red circle in Alighting 16:00-19:00); only alighting is seen, which means that the people alight at this location and travel via train or other modes of transportation from this stop.

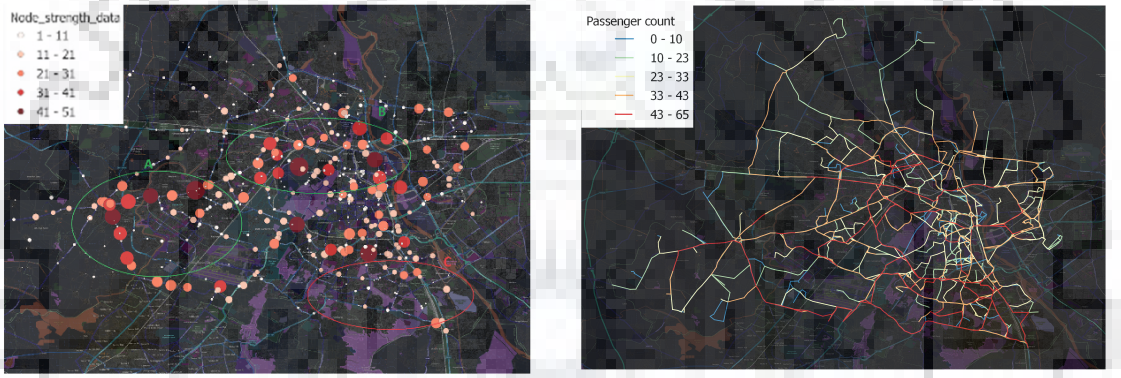


Figure 4.8: Number of routes passing through a stop on left side and the occupancy level of bus in between stops on right.

The flow map diagram, see Fig. 4.9 serves as a visual representation of passenger flow from source to destination. By using different shades and widths of the arrows, the diagram emphasizes the volume of passengers travelling between each pair of locations. The observation is that people prefer bus services for shorter distances. It also indicates the importance of adequate transportation options for longer distances, as passengers may choose alternative modes of transportation if bus services are not available or convenient.

The differences in the number of routes passing through stops in different regions can also impact the wait time and crowding at stops. As seen in Fig. 4.8 In Regions A and B, with a higher number of routes, passengers are likely to experience shorter wait times because the number of routes passing through each stop is high and which may result in less crowding on stops, which can contribute to a more efficient public transportation system. On the other hand, in Region C, where there is a lower count of routes, passengers may have to wait longer for a bus or train to arrive, causing crowding at stops and leading to potential frustration and dissatisfaction. To address this, transportation authorities

may consider adding more routes or increasing the frequency of existing routes in Region C to help alleviate the crowding and improve the overall passenger experience.

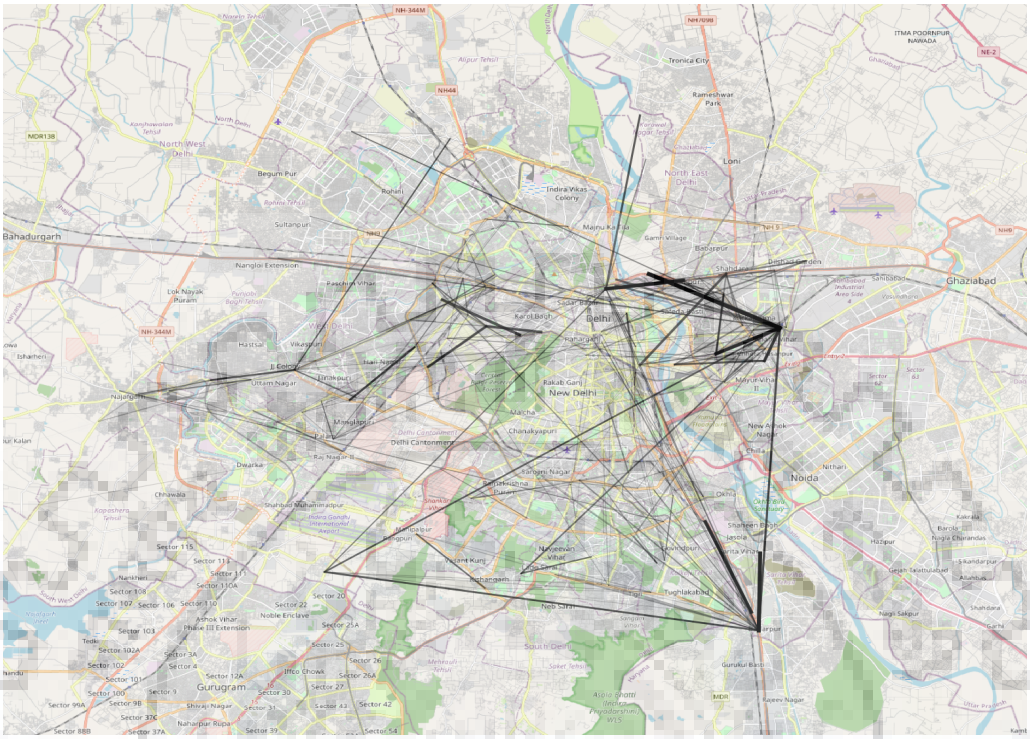


Figure 4.9: Flowmap for passenger travel

Crowding at bus stops is a common issue in urban areas where public transportation is widely used. It refers to a situation where the number of people waiting for a bus exceeds the available space on the bus stop. This can lead to longer wait times, frustration among commuters, increased likelihood of accidents and injuries, and even missed or delayed buses. In addition, crowding at bus stops may increase the risk of transmission of infectious diseases, particularly in light of the COVID-19 pandemic.

However, the prediction of crowding at bus stops can help mitigate some of these issues. For example, real-time data on the number of people waiting at a given bus stop, combined with information on the frequency of buses and their capacity, can be used to estimate the likelihood of crowding. This can allow transportation authorities to take proactive measures, such as increasing the frequency of buses or providing additional temporary stops, to prevent or alleviate crowding.

4.1.3 Model results

route level results

Route 717AUP is selected to apply the proposed methodology due to its higher ridership throughout the day. The bus stops shelters on route 717AUP are being supervised using

instant street view[1] to measure the sitting capacity at the bus stop (Fig 4.10). It is observed that the sitting capacity on this route stop varies from 8 to 16.

Figure 4.10: Ahinsa sthal bus stop shelter

The ratio of the total number of passengers waiting to board a bus to the total sitting capacity available at that stop shelter is considered to estimate the crowding at the stop and is named the crowding index. If the value of the crowding index is greater than 1, it means that there are some passengers standing at that stop and waiting for the bus.

Further analysis is done to know the number of passengers waiting for a bus at that stops. To do the same, the number of passengers boarding the bus at that stop for a time is predicted and divided by the total number of trips passing through those stops at the same time bin.

The result for route 717AUP shows that most of the stop suffers higher ridership and lower trip rate (Fig 4.11), which can be a cause of crowding as shown in Fig 4.12.

network analysis

Further, the network level analysis is done to find the critical stops where there is a higher boarding rate and served with a lower number of trips. In Fig 4.13, the size of the bubble denote the number of trips passing through that stops, and the colour denotes the total boarding in the same time bin. In Fig 4.13, it is clearly visible that some of the stops have a higher rate of passenger boarding, but less number of trips occurred during that time. Further, Fig 4.14 is drawn to know the critical stops/ locations where there is a possibility of crowding at stops by estimating the number of passengers going

[1]https://www.instantstreetview.com/

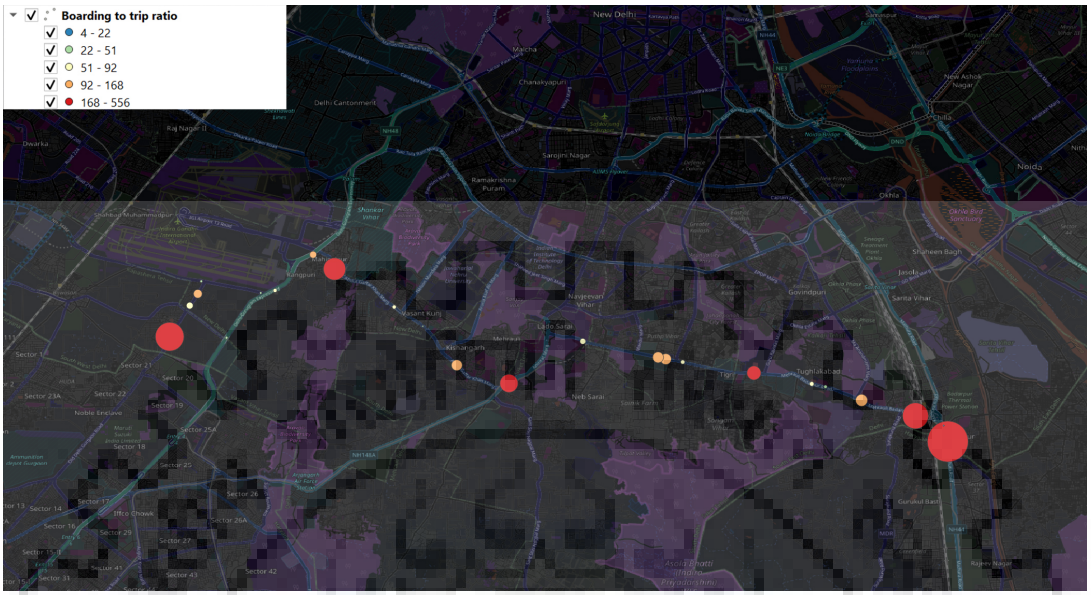


Figure 4.11: Boarding to trips ratio on route 717AUP

Figure 4.12: Crowding index on route 717AUP

to board per trip at that stop in the network. At some of the stops (represented in red), the passenger trip ratio is quite high. Meaning there are many passengers waiting at the stop for a bus to board. In addition to that, using Instant street view, sitting capacity at those locations is checked to estimate the crowding at those stops. It is found that a minimum of six times more passengers are standing at those locations for that particular time compared to seating capacity at those stops Fig 4.15). At some of the stops, a large number of passengers stood due to insufficient space on the transit stops (red dots). Hence, there is a need to provide a sufficient seating capacity to passengers at those locations to improve their comfort while waiting for the bus. As waiting is considered to be a factor which passengers dislike most, hence providing such a facility at a stop may increase their comfort during their trips.

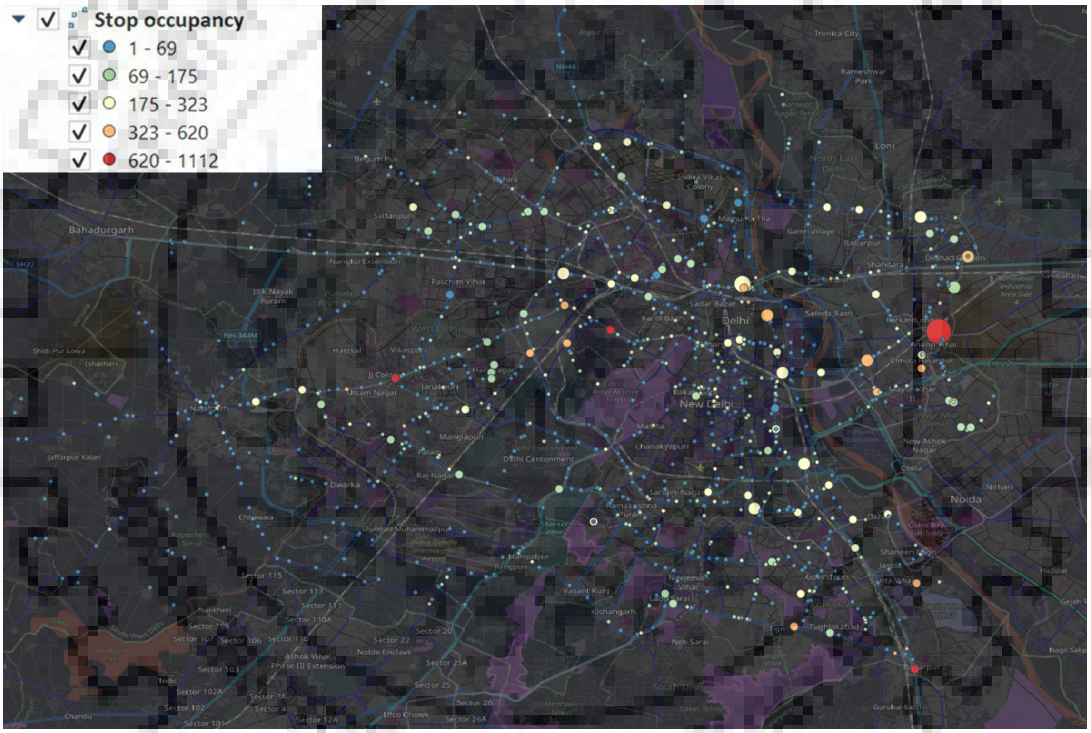


Figure 4.13: Number of passengers going to board as color and number of trips passing at that time bin as size.

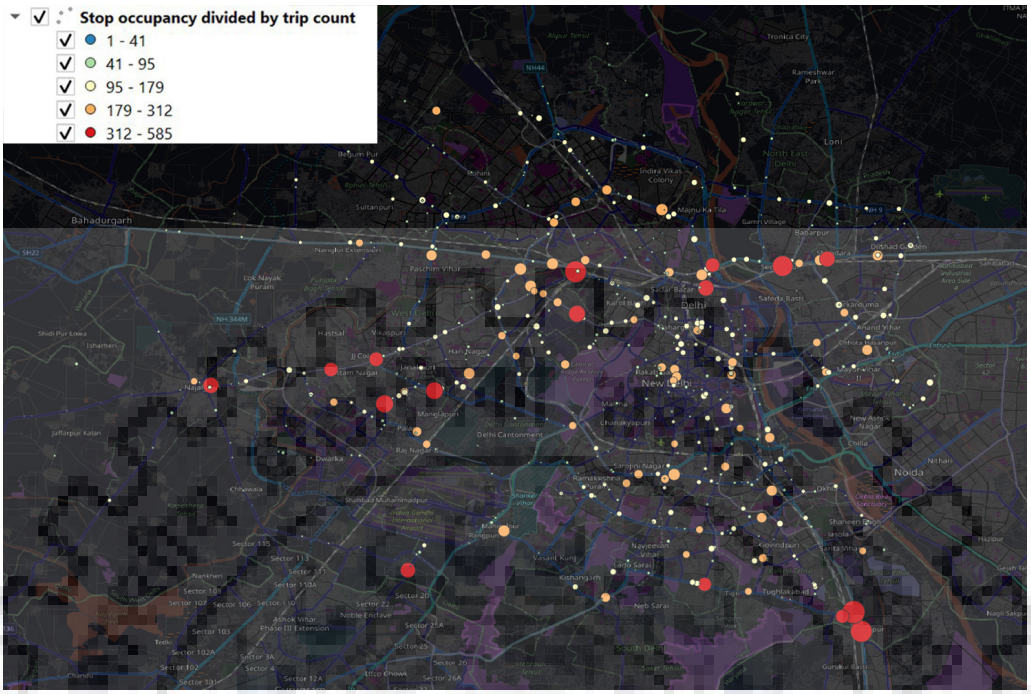


Figure 4.14: Number of passengers going to board divided by the trips in that time bin

Figure 4.15: Crowding index of the high occupancy stops in Fig 4.14

Chapter 5

Conclusions

The current study proposed a real-time stop-level crowding prediction model. The study uses six months of ETM data for buses in Delhi. At first, boarding stops were inferred using the distance and travel time of the bus between two stops, which were calculated by using the time of issuing the tickets. Further, alighting stops were inferred using a combination of explanatory variables, such as point of interest data, population density, green area, industrial area, and residential areas, by applying t-SNE algorithm. The stop crowding is then predicted by using the number of passengers boarding at a stop in a particular time bin to the sitting capacity at that stop. The model MSE value is seen to be decreasing up to ten epochs, and then the loss becomes stationary. Further model training shows the R^2 value of 0.86, which shows the model is well fit on the dataset. From the perspective of transit operators, the proposed stop-level crowding prediction model can be used for the optimal allocation of resources. This can be done by increasing the frequency of trips during the time bin whenever the crowding at stops is high. The operators can also extend the already running route to those stops where node strength is less. The results can also be used for the development of bus stop infrastructure at some of the important locations.

5.1 Future work

1. To obtain a better understanding of passenger flow, one could perform more spatio-temporal analysis on this data.
2. Better ANN architecture could be achieved by tuning hyperparameters or adding/removing neurons or layers.
3. Clustering-based algorithms may also be applied to the data for more generalized training.

References

Alekseev, K. and J. Seixas (2009). "A multivariate neural forecasting modeling for air transport–preprocessed by decomposition: a Brazilian application". In: *Journal of Air Transport Management* 15.5, pp. 212–216.

Anowar, F., S. Sadaoui, and B. Selim (2021). "Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE)". In: *Computer Science Review* 40, p. 100378. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2021.100378 (cit. on p. 20).

Arabghalizi, T. and X. Jia (2021). "A Hybrid Neural Network to Predict Short-term Passenger Flow at Bus Stops". In.

Arabghalizi, T. and A. Labrinidis (2019). "How full will my next bus be? A Framework to Predict Bus Crowding Levels". In: *Proceedings of the 8th International Workshop on Urban Computing. ACM, Anchorage* (cit. on pp. iii, v, 5–7, 11–13).

Bai, Y., L. Sun, H. Liu, and C. Xie (2021). "Using Bus Ticketing Big Data to Investigate the Behaviors of the Population Flow of Chinese Suburban Residents in the Post-COVID-19 Phase". In: *International Journal of Environmental Research and Public Health* 18.11, p. 6066.

Cerqueira, S., E. Arsénio, and R. Henriques (2022). "Is there any best practice principles to estimate bus alighting passengers from incomplete smart card transactions". In: *Transport Research Arena Conference*.

Cervero, R. (1997). "Kockelman., K.(1997) Travel Demand and the 3Ds: Density, Diversity, and Design". In: *Transportation Research Part D* 2.3, pp. 199–219 (cit. on p. 6).

Doi, M. and W. B. Allen (1986). "A time series analysis of monthly ridership for an urban rail rapid transit line". In: *Transportation* 13.3, pp. 257–269.

Giuliano, G. (2004). "Land use impacts of transportation investments". In: *The geography of urban transportation* 3, pp. 237–273 (cit. on p. 6).

Holmgren, J. (2007). "Meta-analysis of public transport demand". In: *Transportation Research Part A: Policy and Practice* 41.10, pp. 1021–1035.

Huang and Herman (1996). "The land-use impacts of urban rail transit systems". In: *Journal of Planning Literature* 11.1, pp. 17–30 (cit. on p. 6).

Kieu, L.-M., A. Bhaskar, and E. Chung (2015). "Public transport travel-time variability definitions and monitoring". In: *Journal of Transportation Engineering* 141.1, p. 04014068.

Kraft, G. and M. Wohl (1967). "New directions for passenger demand analysis and forecasting". In: *Transportation Research/UK/*.

Lv, W., Y. Lv, Q. Ouyang, and Y. Ren (2022). "A Bus Passenger Flow Prediction Model Fused with Point-of-Interest Data Based on Extreme Gradient Boosting". In: *Applied Sciences* 12.3, p. 940 (cit. on pp. 1, 5, 7, 8).

Lv, Y., W. Lv, Y. Ren, and Q. Ouyang (2021). "Optimizing the bus operation plan Based on Deep Learning". In: *Microprocessors and Microsystems*, p. 104042.

Maaten, L. van der and G. Hinton (2008). "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86, pp. 2579–2605 (cit. on p. 20).

Mistretta, M., J. A. Goodwill, R. Gregg, and C. DeAnnuntis (2009). "Best practices in transit service planning". In.

Nguyen, H., L.-M. Kieu, T. Wen, and C. Cai (2018). "Deep learning methods in transportation domain: a review". In: *IET Intelligent Transport Systems* 12.9, pp. 998–1004.

Nwachukwu, A. A. (2014). "Assessment of passenger satisfaction with intra-city public bus transport services in Abuja, Nigeria". In: *Journal of Public Transportation* 17.1, pp. 99–119 (cit. on p. 4).

Pan, Y., S. Chen, T. Li, S. Niu, and K. Tang (2019). "Exploring spatial variation of the bus stop influence zone with multi-source data: A case study in Zhenjiang, China". In: *Journal of Transport Geography* 76, pp. 166–177.

Shen, X., S. Feng, Z. Li, and B. Hu (2016). "Analysis of bus passenger comfort perception based on passenger load factor and in-vehicle time". In: *SpringerPlus* 5.1, pp. 1–10 (cit. on p. 4).

Singhal, A., C. Kamga, and A. Yazici (2014). "Impact of weather on urban transit ridership". In: *Transportation research part A: policy and practice* 69, pp. 379–391.

Talbott, M. R. (2011). *Bus stop amenities and their relationship with ridership: a transportation equity approach*. The University of North Carolina at Greensboro (cit. on p. 6).

Tao, S., D. Rohde, and J. Corcoran (2014). "Examining the spatial–temporal dynamics of bus passenger travel behaviour using smart card data and the flow-comap". In: *Journal of Transport Geography* 41, pp. 21–36 (cit. on pp. 5, 6, 36).

Vasconcelos, V. S., F. Quevedo-Silva, and R. L. Rovai (2021). "Demand Forecasting model based on artificial neural networks for Passenger Transportation Projects". In: *urbe. Revista Brasileira de Gestão Urbana* 13 (cit. on pp. v, 1, 5, 7, 8).

Wang, W., J. P. Attanucci, and N. H. Wilson (2011). "Bus passenger origin-destination estimation and related analyses using automated data collection systems". In: *Journal of Public Transportation* 14.4, pp. 131–150.

Xue, R., D. J. Sun, and S. Chen (2015). "Short-term bus passenger demand prediction based on time series model and interactive multiple model approach". In: *Discrete Dynamics in Nature and Society* 2015 (cit. on pp. iii, 1, 7–10).

Yaakub, N. and M. Napiah (2011). "Public bus passenger demographic and travel characteristics a study of public bus passenger profile in Kota Bharu, Kelantan". In: *2011 National Postgraduate Conference*. IEEE, pp. 1–6 (cit. on p. 4).

Zhang, X. and X. Zhao (2022). "Machine learning approach for spatial modeling of ridesourcing demand". In: *Journal of Transport Geography* 100, p. 103310.

Zhao, J., Q. Qu, F. Zhang, C. Xu, and S. Liu (2017). "Spatio-temporal analysis of passenger travel patterns in massive smart card data". In: *IEEE Transactions on Intelligent Transportation Systems* 18.11, pp. 3135–3146.