Predicting Boarding and Alighting Behaviour of Bus Passengers with Smart Card Data Using Machine Learning Techniques

Tianli Tang

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds

Institute for Transport Studies

March, 2021

Intellectual Property

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

The work in Chapter 2 of the thesis has appeared in the publication as follows:

> Tang, T., Liu, R. & Choudhury, C. 2020. Incorporating weather conditions and travel history in estimating the alighting bus stops from smart card data. *Sustainable Cities and Society*, 53, 101927. DOI: 10.1016/j.scs.2019.101927

The work in Chapter 3 of the thesis has been submitted to *Sustainable Cities and Society* and received an invitation to revise and re-submit:

> Tang, T., Fonzone, A., Liu, R. & Choudhury, C. 2020. Multi-stage deep learning approaches to predicting boarding behaviour of bus passengers.

The work in Chapter 4 of the thesis which is being reviewed by the PhD supervisors prior to submission to a suitable journal:

> Tang, T., Liu, R. & Choudhury, C. 2020. Modelling hourly bus passenger demand with imbalanced data.

The candidate was responsible for the development of the research concept, data collection, data processing, coding computer programmes, development of the

methodology, modelling work, analysis and interpretation of the results, paper writing and correction. Other co-authors contribute to the work of development of the research concept, suggesting modelling work, analysis and interpretation of the results, review & editing.

Acknowledgements

First, I must express my sincere thanks to my supervisors Professor Ronghui Liu and Dr Charisma Choudhury, for their patient guidance and constant support. They are very positive about every idea I had on my research, which enables me to work on my interests. Professor Liu is strict with my studies but always gives me great encouragement when I make progress. Dr Choudhury cares so much about my work and responds to my questions and queries so promptly.

I would like to extend my great appreciation to Dr Achille Fonzone, who hosted me at Edinburgh Napier University. The visit is a beautiful trip that gives me a chance to learn from experts outside our university. Also, his valuable idea and careful editing contribute enormously to the production of the paper in Chapter 3.

I am so grateful for the help from Dr Hongbo Ye, Dr Ying Wang and Dr Weiming Zhao. The topic of this thesis is derived from the discussion with them. They guide me on how to carry out my work at the very beginning. My thank also goes to Mr. Daniel Johnson and Professor Zhiyuan Liu who help me improve the presentation of the thesis.

I am deeply indebted to Professor Shaopeng Zhong, Professor Kai Liu and Professor Shengchuan Zhao. They brought me into the world of research and encouraged me to continue my study.

I gratefully acknowledge the funding and scholarship received towards my PhD from the Future Street project and the RAILS project. Thanks to Professor Greg Marsden

and Dr Zhiyuan Lin. Working with them enriches my knowledge and experience outside my field of study.

Special thanks to the data providers, Hunan Longxiang Bus Co., Ltd. for the GPS data, Changsha IC Card Centre for the smart card data and Changsha Meteorological Bureau for the weather data.

My completion of this thesis could not have been accomplished without the support of my colleagues and friends. Thanks to Zhizhuo, Zihao, Shuo, Yi, Zhuoqian, Yitong, Jipin, Penghui, Fangqing, Jeff, Anna, Tamas, Haruko and other colleagues in ITS and university for their help during varying stages of my study. Thanks also go to Jian, Chen, Yuting, and Pan. I always benefit a lot from learning from them. Many Thanks to Chelegeer, Yuanxuan, Yijun, Duqing, Yifan, Yunhan, Min and Jianxia. I will not forget any time with them. Auld Lang Syne!

I would like to pay my special regards to my family. My parents try their best to provide me with the best learning and living conditions and environment. Their endless love and unreserved support are indispensable to complete the four years of study.

Finally, to my dear Yiwang, who has been by my side throughout this PhD, and without whom, I would not have had the courage to embark on this journey in the first place.

Abstract

Developing an efficient public transport is an important initiative to ease traffic congestion and to reduce energy usage and air pollution. In addition to a well-planned network, an advanced public transport system should offer a comfortable, safe and reliable service for passengers, which requires an appropriate strategy of management and operation. The development of smart infrastructure in public transport ticketing systems has not only improved the operation efficiency and enhanced passenger travel experience, but the development has also made available millions of passengers' daily travel records. This valuable data source can be used to analyse passengers' travel behaviour and travel demand, which in turn can help bus companies offer better public transport services for passengers.

This study mainly aims to understand and predict the boarding and alighting behaviour of bus passengers, from smart card records, using machine learning (ML) approaches. Firstly, a gradient boosting decision tree (GBDT) ML model is trained with features of passengers' boarding records, their travel history, as well as weather conditions and travel history. The model is then applied to estimate the alighting stop for each smart card trips. Secondly, a multi-stage deep learning-based ML framework is developed which utilises the fully connected network, recurrent neural network (RNN) and long short-term memory (LSTM) network, to predict the hourly boarding behaviour (on whether to travel and which bus stop to use) for every smart card user. Thirdly, a deep generative adversarial network (Deep-GAN) is proposed to counter the issue with imbalanced data, where positive instances (in our case a boarding instance in any given

hour) is much less than negative instances, and to improve the prediction on the hourly boarding demand from the smart card data.

The studies indicate that: i) ML techniques are an effective predictive tool to deal with multiple variable and non-linear relations; ii) including weather conditions and travel history can significantly improve the performance of predictive models; iii) the problem of innumerable classes of data fields and imbalanced data records significantly reduce the accuracy of predictive models; iv) In addition to providing good prediction power, GBDT-based ML models provide the ability to rank the relative importance of features; v) RNN and LSTM are capable of capturing the temporal characteristics (i.e. the peak hours) of passengers' boarding behaviour; vi) Deep-GAN models can be effectively used for reducing the problem of data imbalance and enhancing the performance of the predictive models.

Table of Contents

List of Tables

List of Figures

Abbreviations

5G	5th Generation Mobile Networks
ADASYN	Adaptive Synthetic Sampling
AFC	Automatic Fare Collection
AI	Artificial Intelligence
ANN	Artificial Neural Network
APC	Automatic Passenger Counter
AR	Autoregressive Model
ARIMA	Autoregressive Integrated Moving Average
AVL	Automatic Vehicle Location System
Bagging	Bootstrap Aggregating
BCP	Binary Classification Problem
BP-NN	Back-Propagation Neural Network
CBD	Central Business District
CCTV	Closed Circuit Television
CNN	Convolutional Neural Network
CO_2	Carbon Dioxide
Deep-GAN	Deep Generative Adversarial Network

DfT	Department for Transport
DNN	Deep Neural Network
DQN	Deep Q-Network
DSS	Decision Support System
DT	Decision Tree
EMD	Empirical Mode Decomposition
ENN	Edited Nearest Neighbour
EU	European Union
FCN	Fully Connected Neural Network
FG	Feature Group
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GBDT	Gradient Boosting Decision Tree
GDP	Gross Domestic Product
GIS	Geographic Information System
GPS	Global Position System
GRNN	Generalized Regression Neural Network
GRU	Gated Recurrent Unit
HL	Hamming Loss
ID	Identity

IMF	Intrinsic Mode Function
IMM	Interactive Multiple Model
IPTS	Intelligent Public Transport System
ITS	Intelligent Transport System
kNN	k Nearest Neighbours
LR	Linear Regression
LSTM	Long Short-Term Memory Neural Networks
MCCP	Multi-Class Classification Problem
ML	Machine Learning
MLC	Michelson-Like Contrasts Measure
MLCP	Multi-Label Classification Problem
MLR	Multinomial Logistic Regression
MLRM	Multiple Linear Regression Model
NASEM	National Academies of Sciences Engineering and Medicine
MNL	Multinomial Logit Model
MSRBF	Multiscale Radial Basis Function Network
NN	Neural Network
OD	Origin-Destination
OR	Operation Research
PCA	Principal Components Analysis
PEME	Proportion Emphasized Maximum Entropy

RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RENN	Repetitive Edited Nearest Neighbour
RF	Random Forest
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RMSPE	Root Mean Square Percentage Error
RNN	Recurrent Neural Network
RUS	Random Under-Sampling
SARIMA	Seasonal Autoregressive Integrated Moving Average
SMOTE	Synthetic Minority Over-Sampling Technique
SVM	Support Vector Machine
SVR	Support Vector Regression
TIS	Traveller Information System
TN	True Negative
TP	True Positive
U. S.	United States
UN	United Nations
VMS	Variable Message Sign
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

1.1. Smart public transport

Cities are the engines of the growth and improvement of society, economy, culture and innovation, where the power of agglomeration and industrialisation create employment and wealth and promote human's progress (Turok, 2014). Cities provide more national income than their share of the national population. For example, 12% of the population of the Philippines in Metropolitan Manila accounts for 47% of GDP (UN, 2016). The process of global urbanisation continues to move forward in an unstoppable trend. Figure 1.1 shows that more than half of all countries and regions have over 60% of the level of urbanisation in 2015, and most of North America can and Europe are urbanised over 90%. According to the report (UN, 2016), 54% of the world's population resides in the urban area, and this percentage will go up to 68.4% by 2050. The rapid process of urbanisation in developing countries provides a window of opportunity to leap-frog the already developed and mature cities and ensure sustainable development.

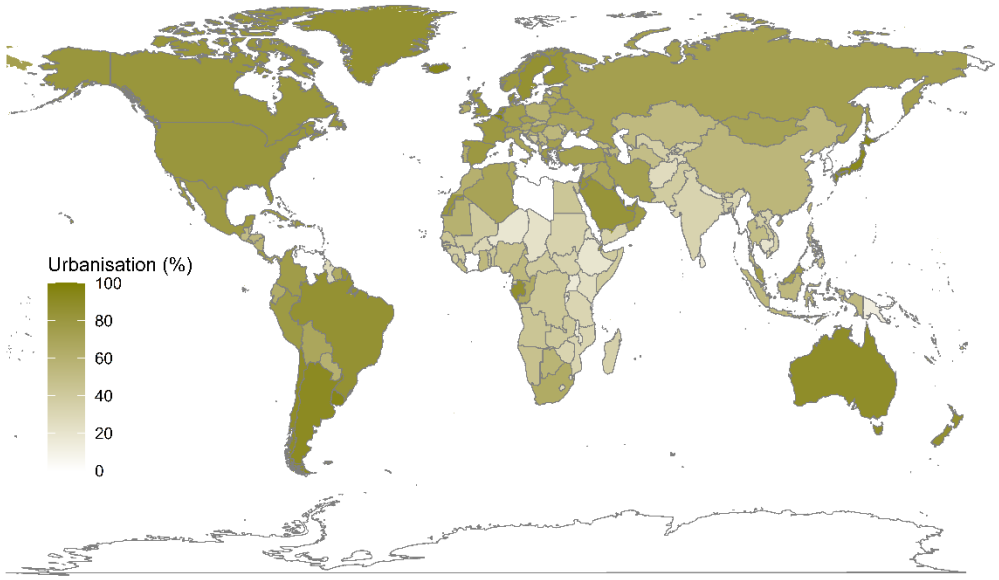

Figure 1.1 The level of urbanization for each country and region in 2025.

Data source: UN (2016).

Although urbanisation has the potential to drive cities more prosperous and countries more developed, many cities have not been prepared for the challenges associated with urbanisation yet. The uncontrolled urbanisation brings more and more people into cities, which results in a surge in travel demand. The tremendous growth of travel demand can be a threat to transport system. For instance, the rapid increase in private cars, coupled with insufficient infrastructures, has caused severe traffic congestion in Seoul (Pucher, 2005). More than 210 million people in the EU are living in an environment full of traffic noise, which is harmful to human health (Jacyna et al., 2017; den Boer and Schroten, 2007). And, the air pollution from vehicle emissions damages to public health (Lim et al., 2012).

Then the question arises, how to manage the transport system facing the constant upgrade of population and urbanisation. As a vital component of a sustainable city, achieving development of sustainable transport with advanced technology is becoming the best solution. It is well known that the public transport system is a green and

sustainable transport mode to solve the issues in cities, e.g. severe air pollution, excessive energy consumption, terrible traffic congestion, and excessive travel cost. Socially, the public transport is environmental-friendly. High patronage will share more traffic demand for private cars and thus result in a decrease in the use of cars, which can save many road-space resources. Further, the popularisation of electric buses has reduced the use of fossil fuels and the resulting carbon emissions and air pollution. The Munich Transport Company, which operates Munich's underground, trams, and most of its buses, saves 160,000 metric tons of CO_2 per year when compared with private vehicles (NASEM, 2009). For the public, public transport offers a cheap, convenient and health services. Taking public transport and living with one less car will help a family save nearly ten thousand U.S. dollar a year (Hughes-Cromwick and Dickens, 2020). Legrain et al. (2015) find out driving is the most stressful transport mode due to the traffic congestion and delay and taking public transport may help commuters avoid this kind of stress. Not only these, but public transport can also provide economic opportunities for cities. Hughes-Cromwick and Dickens (2020) reports that every ten million U.S. dollar in capital investment in public transport will produce five times in economic returns, add 500 new job positions and increase thirty million U.S. dollar in business sale. However, the public transport service is always being criticised owing to its poor image in terms of level-of-service caused by low reliability, accessibility and availability.

In recent decades, the new generation of science and technology has contributed to the rise and spread of the 'Smart Transport' policy. Smart transport, as known as intelligent transport system (ITS), is the key for sustainable cities. The shared idea of smart transport is about managing and developing transport systems from a new angle of digital construction, which is of great significance for urban management and operation (Yigitcanlar et al., 2018). Smart transport armed with artificial intelligence (AI) will lead to the future development of cities (Deloitte, 2018). Since different places have different

aims and requirements for the transport system, the definition of smart transport varies from place to place. Each smart transport project has its own tools for information collection, analysis and delivery. For example, CIVITAS supports a public transport communication system in Tallinn, Estonia, based on the 3G mobile communication standard (Boschetti et al., 2014). This system enables faster share of information and better traffic management. Mobileye's EyeQ® chip (Mobileye, 2020) applies the technologies of computer vision and image processing to deal with the video from the camera on bus vehicles. This product is able to monitor the bus headway and lane-keeping and to warn drivers about forward collision and pedestrian and cyclist collision.

With the development of the digitisation of the city and the intellectualisation of the transport systems, an intelligent public transport system (IPTS) as a subsystem of smart transport is proposed to address the problems and to improve the level-of-service. An IPTS relies on much-advanced communication, information processing, control and electronics technologies, which mainly encompasses five systems (Elkosantini and Darmoul, 2013). Figure 1.2 illustrates the different components of the IPTS. Automatic vehicle location (AVL) system provides real-time information about the vehicle, such as location and speed. Automatic passenger counters (APC) collects the number of passengers on-board and waiting at stops. Geographic information system (GIS) maps the information from AVL and APC to the network. Decision support system (DSS) suggests the control decisions to operators and decision-makers according to the analysis from the monitoring of AVL and APC. Traveller information system (TIS) informs passengers about the real-time situation of the public transport system, such as waiting time and travel time.

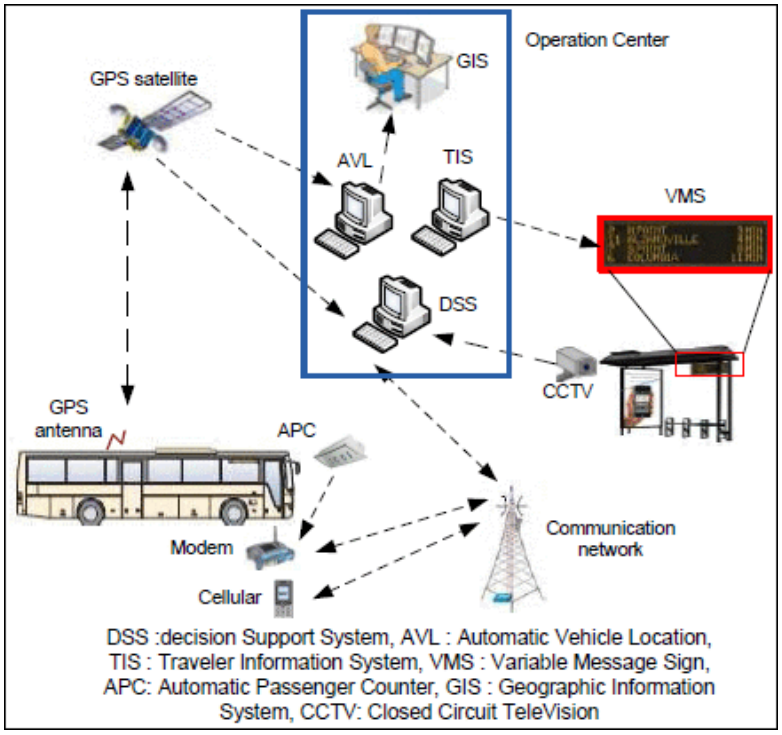

Figure 1.2 Illustration of IPTS (**Elkosantini and Darmoul**, **2013**).

For example, the city of Hangzhou, one of ten demonstrators for ITS in China, has developed an IPTS (Luo, 2020). The system utilises an AI-based platform hosted on Ali cloud computing cluster to analyse and process massive data. This platform can accurately measure the spatial-temporal dynamics of ridership, evaluate the performance of the network and schedule the services in real-time. Those tools will help significantly improve the level-of-service and enhance the attractiveness of public transport. However, the current construction of IPTS in many cities remains at an early stage of development, which independently involves in as many new technologies as possible but does not communicate or share information among sub-systems. An advanced technique, pay-by-face, has been applied to buses for the automatic fare collection (AFC) system in Yinchuan, China. This technique is intended to make it easier for passengers to use buses, and to reduce the boarding time. However, this pay-by-face technique has not been applied to other public transport modes like metro. As a result, the survey from Sorrell (2019) finds that the occupancy of buses are still low,

and the advanced technique shows limited attraction to passengers. Therefore, the concept of novel smart city needs to put human foremost, to upgrade the city with the efforts of innovation and final to achieve the goal of sustainable development (Deloitte, 2020).

1.2. Machine learning in intelligent public transport systems

Since the renaissance of AI in the 1980s, machine learning has greatly improved both in terms of algorithms and available datasets. Machine learning specialises in processing high-dimensional and multivariate data and capturing complex nonlinear relationships in the data (Witten et al., 2016). The advantages of machine learning lead to that thousands of AI applications has already been deeply embedded in the infrastructure of daily production (Kurzweil, 2005). An increasing number of studies has brought machine learning techniques in analysing kinds of data in public transport systems. This section will review the applications and state-of-arts methods of machine learning technique in several domains of public transport systems.

1.2.1. Passenger mobility

Passenger flow prediction is the premise of public transport planning and an important condition to ensure that the planning conforms to future development. One of the most common approaches is to predict the volume of ridership directly. The AFC system, employed in many urban rail networks, records the time and location of passengers' entering and leaving the station. Then an origin-destination (OD) matrix of passenger flows and ridership can be easily obtained from the logs of AFC system. Such historical ridership can then be used to predict future ridership through time series analysis. For

example, the passenger flow of an underground (metro) system has been predicted based on the smart card AFC data using multiscale radial basis function (RBF) network under special events (Li et al., 2017), while long short-term memory neural network (LSTM) has been applied to capture the temporal characteristics of passenger flows (Liu et al., 2019b). Ding et al. (2016) model the ridership entering three underground stations by three independent gradient boosting decision tree (GBDT) models. To deal with the time series of ridership, the empirical model decomposition (EMD) model is a popular tool to decompose the original sequential passenger flow into several intrinsic mode functions (IMFs) and a residual. The EMD model is respectively hybridised with the back-propagation neural network (BP-NN) (Wei and Chen, 2012) and LSTM (Chen et al., 2019) to predict the time series of underground (metro) ridership. When applying the time series to LSTM for passenger flow prediction, Zhang et al. (2019) propose the K-Means clustering model to capture the variation trends and characteristics of ridership and thus to recommend a reasonable time granularity interval to aggregate passenger flows. Liu et al. (2018) compare the performance of the recurrent neural network (RNN), gated recurrent unit (GRU) and LSTM in predicting ridership through a sequences data, and the finding shows that the LSTM is better in learning and remembering over long sequences. The above works require the historical ridership that is based on the complete OD matrix. However, the AFC system in some bus networks does not contains both boarding and alighting information.

For bus networks, the AFC system operates under a range of different regulations: boarding only, alighting only and both boarding and alighting. Alighting-only AFC is the least used system. Correspondingly, few studies have been done on alighting-only AFC. For both-boarding-and-alighting AFC, the smart card is used during both boarding and alighting. Directly from the data of this kind of AFC, we can obtain the boarding and alighting stops and time, so it is easy to measure the ridership, and the

following analysis and forecasting on ridership is similar to the rail system introduced above. For the boarding-only AFC, passengers are required to swipe smart cards during boarding. So, the smart card data records the boarding information for each trip. However, in some places such as Changsha and Shenzhen, there is no direct boarding stop or other geographic information in the smart card data. Therefore, to infer the boarding stops is the first step to analyse the smart card data. Since buses are equipped with a GPS device, it is easy to merge the geographic information from GPS to the AFC records. The research of boarding stop inference focuses on the AFC records with missing GPS data. Ma et al. (2015a) improve the Bayesian decision tree algorithm to calculate the likelihood of each possible boarding stop for the buses without GPS devices. Shalit et al. (2020) take the smart card data, of which we know the boarding stops, as the training dataset to train the linear regression (LR), random forest (RF) and extreme gradient boosting (XGBoost) models for predicting the boarding stops of the smart card data missing GPS. Other than the smart card data, smartphone sensors are also introduced to detect the boarding stops by monitoring the acceleration from support vector machine (SVM)-based predictive model (Chaudhary et al., 2016). After inferring the boarding location, the main task for boarding-only AFC is to determine the alighting stops of bus trips. There is not much research in the area of alighting stop estimation by machine learning techniques. The most classic and popular method is a rule-based method, the trip chain mode (Barry et al., 2002; Barry et al., 2009). To add the alight information into the OD matrix, Yan et al. (2019) use trip chain mode to obtain the alighting stops of a part of smart card data as the input training data, then use Naïve Bayesian, SVM, decision tree (DT), RF, k-nearest neighbours (kNN) and ensemble learning to classify the smart card data into different alighting regions, rather than specific alighting stops. Similarly, Jung and Sohn (2017) get some result of alighting stop estimation by the trip chain and infer the alighting stop from five candidate stops

for the boarding-only smart card data in Soul by using a deep neural network (DNN) model. However, this paper limits the possible alighting stops in five candidate stops. The case does not consider the difficulty of classifying the trips into hundreds of classes (alighting stops). These two studies belong to classification problem. However, either of them do not consider the poor quality of smart card data, such as the data imbalance. The number of instances in each class is different, and the skewed distribution will harm on the performance of machine learning models (Guo et al., 2017). The data imbalance will be discussed in detail in Chapter 4.

The public transport system is a collaboration of multiple transport modes; a trip always involves in the choose and transfer among different public transport modes. Thus, choice modelling is an essential task in travel demand prediction. Many machine learning techniques are developed to be the adequate tools for the mode choice prediction, for example, artificial neural network (ANN) (Cantarella and De Luca, 2003), SVM (Omrani, 2015) and DT (Rasouli and Timmermans, 2014). Celikoglu (2006) uses the RBF neural network and generalized regression neural network (GRNN) to propose a new calibration process for travel mode choice analysis in a transportation modelling framework. Hagenauer and Helbich (2017) present a systematic comparison of seven different machine learning classifier for travel mode choice prediction. This study finds out that the RF is the best way for the mode choice perdiction, which is much better than the commonly-used multinomial logit (MNL) model, and the classifiers are with high sensitivity for public transport. Zhou et al. (2019) use several machine learning techniques, such as kNN, SVM, DT, RF, and some ensemble models, such as AdaBoost, Bootstrap aggregating (Bagging) and gradient boosting decision tree (GBDT) , to model the model choice between sharing bikes and taxis and analysis the impacts of the features on their choice. It shows that the choice of sharing bikes is most impacted by season and weather conditions.

1.2.2. **Service planning and management**

The domain of traffic planning and management is defined as the process of planning and designing for future transportation needs of people and goods (McLeod et al., 2017). Traffic planning includes transport policies, legislative activities, operation, provision, facilities management, financial allocation, investments, economic activity, urban design and many other factors. The goal is to optimise the transport flow according to the resources available. Traffic management refers to all procedures and equipment that enable a proper operation both during regular and special circumstances.

The key to traffic planning and management is to collect and analyse the transport information and to understand the situation of transport systems for appropriate policy and regulation. Therefore, the first research hotpot in this field is to understand the various parameters for describing the public transport system, such as travel time and arrival time. SVM is a useful method for multiple nonlinear data, which can deal with sequential data (Bin et al., 2006) and multiple independent variables (Zeng et al., 2015). Also, ANN is another popular method for the travel and arrival time prediction, which shows a good performance in the presence of a large database (Kumar et al., 2014).To discover the complex patterns of the distribution of bus travel time, Petersen et al. (2019) combine the architecture of convolutional neural network (CNN) and LSTM to use the non-static spatial-temporal correlations for the bus travel time prediction, by which the irregular peaks in bus travel time can be detected. In addition to exploring the feasibility of each machine learning model in forecasting tasks, some studies are focusing on the difference in the performance of various machine learning models. Ranjitkar et al. (2019) compare the performance of multivariate linear regression, decision tree, ANN and gene expression programming models in predicting the bus arrival time based on the real-world data in Auckland and show the input of time series can improve the accuracy of the prediction. The ANN is the best method for the bus travel time

prediction among the mentioned methods. However, it is difficult to explain the nodes and weights in ANN. Even though ANN is the best predictive model, it is of little help in our understanding of the influencing factors. So, it is important to understand and find out the features impacting on the bus travel time. Sun et al. (2019) develop a framework to extract features for capturing local-spatial correlation, global-spatial correlation, recent and distant historical information, temporal periodicity, as well as contextual information such as road network characteristics and other extrinsic factors, and investigate the intrinsic and extrinsic features that impact the bus speed and their significance in special scenarios when predicting the bus speed by ANN.

Scheduling of public transport services is another vital role in public transport planning, and traditionally operation research (OR) methods are widely used for the scheduling problem (Ibarra-Rojas et al., 2015). To solve large-scale scheduling problem, heuristic algorithms, such as genetic algorithm (Yin et al., 2019), ant colony optimization (Wei et al., 2012) and simulated annealing algorithm (Hanafi and Kozan, 2014), are popularly employed. Recently, the development of unsupervised learning brings a new approach to solve scheduling problems, for example, reinforcement learning (RL) approach (Salsingikar and Rangaraj, 2020) and Deep Q-Network (DQN) (Obara et al., 2018). Khadilkar (2019) compares the RL and classic heuristic algorithms in solving the rescheduling problem and concludes that RL could i) utilise the value of objective for the learning processes, (ii) take advantage of the characteristics of specific problem instances, (iii) generalise from one problem instance to another, and (iv) not require a great deal of domain expertise for defining the scheduling rules.

1.2.3. Maintenance and inspection

To keep public transport users safe and ensure the smooth operation of public transport systems, the field of maintenance and inspection is a significant component in the public

transport system. The reliability, safety and vehicle life of public transport systems depend on the maintenance scheme (Haghani and Shafahi, 2002). The tasks of maintenance and inspection are to monitor conditions of equipment and infrastructure and make maintenance plan before the deterioration happens. However, Andrle (1986) point out that skilled experts for correct problem diagnosis are not always available, which requires a smart system to aid mechanics in diagnosis.

Since the fault of train and bus has a different influence on the running statues and safety degree, the research of applying machine learning to the domain of maintenance and inspection in concentrated in the rail system. The first areas of applications in maintenance and inspection are to use machine learning techniques to detect the fault of infrastructures from multi-source data. For example, the health of track is diagnosed based on the signal pattern of dynamic vehicle response (Firlik and Tabaszewski, 2020), and other potential factors such as traffic volume, defect amplitude, track class, and speed (Hu and Liu, 2016). Gao et al. (2018) combine three independent data rouses, ultrasonic B-scans, eddy current testing probes and surface imaging video, for detection of squat-like specific rolling contact fatigue defects. In these studies, machine learning techniques shows a remarkable ability to deal with the multi-source and high-dimension data in pattern recognition. Besides the external features, the deterioration of infrastructures shows some regularity in the temporal-spatial distribution, which can be measured by LSTM (Bruin et al., 2017) and deep convolutional neural networks (Krummenacher et al., 2018). Besides, how to improve the performance of machine learning models is also a focus of research interest. Lasisi and Attoh-Okine (2018) suggest that using principal components analysis (PCA) to reduce the dimension of the track geometry data, which contributes to the accuracy and efficiency of machine learning in rail exception and defects monitoring. These studies contribute to the use of data sources and feature

engineering. They prove that it is an excellent choice to use machine learning techniques in predictive tasks.

Moreover, the current inspection strategy is visual inspection. The second area of applications is to use the techniques of computer vision and image processing for automatic visual inspection of the computer. Aydin et al. (2013) use canny edge algorithm and Hough-transform to obtain the height of the pantograph in the image from the pantograph-catenary system. Among the machine learning techniques, CNN is a popular and effective model to deal with image data. There are many studies based on the CNN model (Giben et al., 2015; Faghih-Roohi et al., 2016) and its improvement (Kang et al., 2019; Han et al., 2020). The studies above are based on image/video data, so improving the quality of data is an essential task for a good performance of machine learning models. Li and Ren (2012) improve the algorithm of the enhancement of image quality and automatic thresholding by the Michelson-like contrast measure (MLC), and proportion emphasized maximum entropy (PEME) thresholding algorithm in detecting discrete surface defects in a vision system. To improve the class balance, Hajizadeh et al. (2016) apply semi-supervised learning techniques such as self-training and co-training to identify and add new positive examples to the training dataset, which can obtain a higher performance in detecting the Squats from image data. There are several studies paying attention to the quality of the database and trying to improve data quality. However, this data issue, for example, the imbalanced data issue, have not been received researchers' attention in the field of public transport studies. The review of imbalanced data issue will be discussed later in Chapter 4.

1.3. Ridership prediction in the bus system

OD matrix is the foundation of spatial analysis of the ridership. It provides the information about the use (boarding and alighting) of bus stops and bus lines in a network. According to OD matrix, operators can easily obtain the distribution of ridership along the network and evaluate the rationality of the network structure, for example, where there is too much or too little demand. The bus lines and their scheduling can be optimised based on an accurate OD matrix. Accurate forecasting of bus OD matrix benefits on both passengers and bus operators. Bus passengers can re-plan their trips by changing transport mode and postponing departure time to avoid the discomfort of being crowded (Brakewood and Watkins, 2019). Bus operators can re-schedule the timetable to improve the reliability and level-of-service of buses. Therefore, the ridership prediction in bus system is of great importance. In this section, I will review the development of bus ridership prediction from aspects of predictive methods and data sources.

1.3.1. Development of bus ridership prediction

The most traditional method for bus ridership prediction is covered by the four-stage model. In the stage of trip generation, the total demand, including all transport modes, is built into a predictive model. Then the ridership of buses will be proportionally separated from overall demand in the stage of mode split. Meanwhile, there are many studies paying attention to how to predict the bus ridership directly. According to the review from Banerjee et al. (2020), the predictive models are classified into three groups: causal model, time series model and AI model. As the AI model is reviewed in Section 1.2.1, the flowing context will introduce the causal and time series model.

1.3.1.1. Causal models

The basic idea of the causal model tries to incorporate the explaining factors to the predictive models of bus ridership and to describe their correlations. Following this idea, we can build up the predictive regression functions, such as LR, based on the relationship between bus ridership and such related factors, e.g. populations, economic attributes, and land-use. Varagouli et al. (2005) fit a multiple linear regression model (MLRM) for the demand forecasting based on the variables of GPD, population, the number of cars and trip characteristics by car of the OD zones. Guo et al. (2007) use the MLRM to analyse the impact the independent weather variables on the bus ridership, which can also be used to predict the bus demand under different weather conditions. Carpio-Pinedo (2014) presents an MLRM to predict the bus demand at stop-level with the impacts of space syntax and other built environment factors. Zhou et al. (2017) model the relationship between bus demand and changes of weather conditions by MLRM and analyse their correlations and significance of the correlation. Khan et al. (2019) produce a bus demand prediction form the MLRM to investigate the potential users when giving the observations of bus passengers and the features of the day. Wei et al. (2019) propose the negative binomial models to model the relationship between bus demand and weather conditions. Unlike using the variables independently in the above research, this study also considers the interplay between the weather conditions. These models can explain the relationship between ridership and influencing factors and help us select the relative features in predictive models. However, most of these models assume that the relationship is linear, which is not consistent with reality. A more accurate prediction requires that the predictive model can measure a non-linear relationship even consider the interaction between variables.

1.3.1.2. Time series model

Bus ridership generally shows regular periodic changes. For example, there are always two peaks in a day in the morning (from about 7 to 9 am) and evening (from 5 to 7 pm). Moreover, the number of commuters is relatively steady on Tuesday, Wednesday and Thursday. There would be more commuters on Monday and Friday due to the impacts of weekends and less demand at weekends (Nurul Habib et al., 2017; Long et al., 2012). Therefore, time series model is a suitable way to reflect the changes in demand over time (Washington et al., 2010). Autoregressive (AR) methods were developed for time series data. Later, the improvement of AR model appears, for example, autoregressive integrated moving average (ARIMA) and seasonal autoregressive integrated moving average (SARIMA) model (Zhang, 2003; Cryer and Chan, 2008). Gong et al. (2014) use a SARIMA model to predict the number of passengers arriving at bus stops from the historical boarding, empty space and GPS data. Xue et al. (2015) employ the ARIMA method to predict the bus demand at the weekly and 15-minute time interval and the SARIMA for daily demand. Zhou et al. (2013) find out that the ARIMA model presents a better performance for the prediction of bus demand at every bus stop when compared with the time-varying Poisson model and weighted time-varying Poisson model. Besides the ARIMA model, Ma et al. (2014) assume that the ridership is related to its hourly, daily and weekly pattern. This study thus suggests predicting the pattern of ridership at the hourly, daily and weekly scales. It then introduces the interactive multiple models (IMM) to capture the relationship and predict the final demand prediction. It is acknowledged that AR-based methods are one of the best ways to deal with the time series data and can receive an accurate result. However, this method needs to model the bus stops and bus lines one by one. The experience in one bus stops cannot be used to others. Considering hundreds of bus stops in a city, it is difficult to widely use this AR-based model to predict the ridership for every bus stop. Additionally, the

AR-based model only captures the regulation in ridership itself but ignores the external factors such as a sudden sport event and rainstorm (see Chapter 2 for details).

1.3.2. Multi-source data for bus ridership prediction

The prediction is always based on the initial conditions that could be the pattern of changes in their conditions and the influence of external factors. So, it is critical to capture the related variables.

In the beginning, without such hi-technology, the ridership is obtained manually. The investigators counted the number of passengers at some bus stops or in a vehicle. We call these methods as site and rider check (Ceder, 2007). Another method is based on the questionnaire survey. The investigators use a survey to collect the characteristics of a small group of passengers and their travel pattern. The overall ridership will be inferred according to the result of the survey. These methods cost a lot of workforce and money for a small sample of demand.

Recently, the Internet of Things connects all the facilities of people's daily life, which enables data from various sensors is available to researchers for the bus ridership prediction. The APC system is developed to collect the direct number of boarding and alighting automatically. APC system uses a passive, non-radiating infrared technology to detect and count people moving through a door or gate (Gerland and Sutter, 1999). However, only 10%-10% of buses in the U.S. are equipped the APC devices (Barabino et al., 2014). Researchers and operators are trying to measure the readership information from indirect data sources. The smart card data is one of the cheapest and most valuable data sources, which has been widely used in the ridership prediction. There are two kinds of designs of AFC system: 'open' AFC system requires passengers to swipe their cards at boarding and records the boarding information, while 'closed' AFC system requires passengers to swipe their cards at both boarding and alighting. The open AFC

is one of the best data sources that can be used to analyse the total bus demand (Bordagaray et al., 2016). Dissimilarly, the closed AFC records the alighting informant, so that is can be used to do deep analysis on the OD matrix (Sun et al., 2016). Additionally, the GPS data provides the location of vehicles for the bus trips when the AFC system does not contain such geographic information (Zhou et al., 2013). The advanced communication technologies, especially 5G, have made mobile phones and the Internet indispensable for people's travelling. Cellular signalling data implies the passengers' mobility and becomes a popular data source to track people's movement (Gundlegård et al., 2016; Xing et al., 2019). The data from social media is also an excellent choice to analyse people's movement (Yang et al., 2019b). As some bus stops and vehicles offer free Wi-Fi to passengers, the accessing actions can be used to follow users' trajectory (Oransirikul et al., 2014).

1.4. Summary of research gaps

Although the previous studies and applications have made a great effort in bus ridership prediction, some research gaps are remaining.

RG 1: Firstly, most of existing studies targeted to predict the total demand of the bus network, i.e. demand along certain bus lines as well as demand at specific bus stops. Although the advanced AR-family methods can achieve the goal accurately, they require building a single model for each bus stop and each bus line. This is computationally burdensome, and the workload is a huge challenge for public transport planner. Thus, it is essential to find a single method that can model the demand for all levels at once.

RG 2: Secondly, not only do we need to know where passengers are coming from (boarding stops), we also need to know where they are going (alighting stop). The OD

matrix tells the planner more details of the passenger flow than the demand at boarding. Nevertheless, the smart card data in most bus-AFC systems do not record alighting information. Although a few studies use machine learning methods to predict alighting stops for bus trips, these studies can only be applied to a limited number of stops. For example, there are only five candidate alighting stop in the DNN model from the study of Jung and Sohn (2017). So, estimating alighting stops for bus trips in network level is still an important task.

RG 3: Thirdly, the impact of environmental factors, like weather conditions, on travel behaviour and ridership is less considered. The bus trip is significantly affected by weather conditions, and passengers have different travelling decisions under different weather events. The related studies will be reviewed in detail in Chapter 2. However, some of the studies investigated the change and rules of passenger flow itself but neglected the impact of environmental factors. Some studies examined the role of a single factor in isolation but ignored the interaction between the factors. When analysing and predicting the OD matrix and passenger flow of bus ridership, considering the impact of weather conditions will make our model more realistic.

RG 4: Finally, whilst the development of smart city brings the techniques of machine learning and AI to the research of bus ridership prediction, there are many difficulties in practice, including noise and missing data and imbalanced data issue (the number of data in different classes is not equal). These data issues result in the inaccuracy and learning errors for machine learning models. The related studies will be reviewed in detail in Chapter 4. Most of the existing studies have ignored the impact of low data quality on the performance of machine learning models. With this, analysis and guidance of how to improve the quality of the practical data are crucial for the application of machine learning techniques in the public transport system.

1.5. Research aims and objectives

The overall objective is to understand and predict the boarding and alighting behaviour of bus passengers at the individual level. The boarding behaviour refers to who gets on a bus at which bus stop, for which bus line and during what time. The alighting behaviour identifies the alighting time and location (specific bus stops) for each bus trip. The smart card data will be the main data source to the predictive models. Since the smart card data does not contain the location of boarding stops, GPS data is used for boarding stop inference. Besides, the hourly weather monitoring data, including visibility, humidity, air pressure, temperature, and precipitation, is used to describe the environmental factors. To accomplish this aim, the following research objectives have been defined:

a) To estimate the alighting stop and time for each bus trip recorded in the smart card data and to analyse the importance of the factors that may impact passengers' alighting behaviour (RG 2);

b) To predict the hourly boarding behaviour of whether to travel and the specific boarding stops for each smart card users (RG 1);

c) To solve the imbalanced data issue in boarding behaviour prediction, that an extremely large percentage of the negative instances results in a poor and inaccurate performance of the predictive model in boarding demand prediction (RG 4);

d) To apply a big public transport data, such as smart card and GPS data, and multi-source data, such as meteorological data, to the machine learning techniques for solving the tasks above (RG 3).

1.6. Thesis outlines and contributions

The thesis comprises of five chapters, and Figure 1.3 illustrates the relationship among five chapters in this thesis. Chapters 1 and 5 are the Introduction and Conclusion of the thesis. Chapters 2 and 3 present machine learning methodology in predicting the alighting stops and the boarding behaviour of smart card users, respectively. These two chapters work on the two aspects of bus demand: boarding and alighting behaviour. Chapter 4 presents an enhancement to the prediction method of Chapter 3: it addresses the imbalanced data issue when prediction boarding behaviour. The work of each chapter is described in detail below.

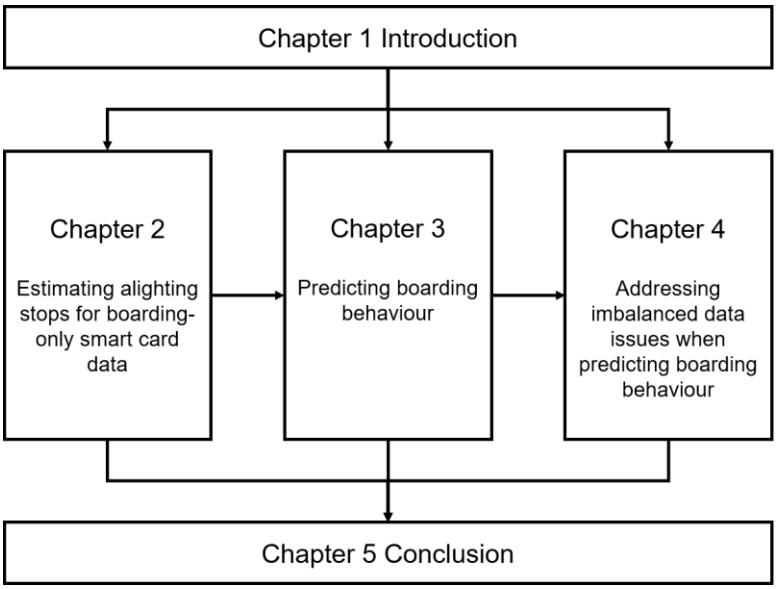

Figure 1.3 The relationship between the five chapters in this thesis.

Chapter 1 introduces the background of the studying topic, the reason why we need to predict the bus demand, including the ridership and OD matrix and reviews the classic and state-of-arts methods on bus demand prediction. It introduces in general terms the machine learning techniques and their broad applications in transport systems, especially public transport systems. The chapter concludes with a summary of the research gaps, the objectives and the thesis outline.

Chapter 2 focuses on applying the novel machine learning technique, gradient boosting decision tree (GBDT) algorithm, on estimating the alighting stops of each smart card transaction that records only the boarding information. Origin-destination flow of passengers in bus networks is a crucial input to the public transport planning and operational decisions. Smart card systems in many cities, however, record only the bus boarding information (namely an open system), which makes it challenging to use smart card data for origin-destination estimations and subsequent analyses. This chapter addresses this research gap by proposing a machine learning approach and applying the GBDT algorithm to estimate the alighting stops of bus trips from open smart card data. It advances the state-of-the-art by including, for the first time, weather variables and travel history of individuals in the GBDT algorithm alongside the network characteristics. The method is applied to six-month smart card data from the City of Changsha, China, with more than 17 million trip-records from 700 thousand card users. The model prediction results show that, compared to classic machine learning methods, GBDT not only yields higher prediction accuracy but more importantly is also able to rank the influencing factors on bus ridership. The results demonstrate that incorporation of weather variables and travel history further improves the prediction capability of the models. The proposed GBDT-based framework is flexible and scalable: it can be readily trained with smart card data from other cities to be used for predicting bus origin-destination flow. The results can contribute to improved transport sustainability of a city by enabling smart bus planning and operational decisions.

Chapter 3 utilises the deep learning approaches to predict the travel pattern of bus passengers and develops a multi-stage framework to solve the imbalanced data issue and many-class problem in the practical dataset that may reduce the accuracy of deep learning models. A clear understanding of the travel pattern of passengers is critical to planning and managing smart public transport systems. Whilst in the past transport

planning rely only on survey responses to study the behaviour of transit passengers, in recent years, smart card data has emerged as a more comprehensive, accurate and cheap source of information. This chapter presents a multi-stage machine learning framework to predict passengers' boarding stops using smart card data. The framework addresses the data challenges arising from the imbalanced nature of the data (e.g. many none-travelling data) and the 'many-class' issues (e.g. many possible boarding stops) by decomposing the prediction for each hour of a daily into three stages: whether a smart card user is expected to travel or not in that one-hour time slot, which line they will use and at which stop they will get on board. I implement a simple neural network architecture, fully connected networks (FCN), and two deep learning architectures, recurrent neural networks (RNN) and long short-term memory networks (LSTM). The predictions also incorporate individual travel histories and weather conditions. The proposed approach is applied to the bus network in Changsha, China, with about 3 million trip-records from 560 thousand card-users. The machine learning models do not show high performance on predicting the decision to travel by bus, and the choice of the line and the specific boarding stops at the individual level. However, aggregated ridership results show that RNN and LSTM are able to measure the temporal characteristics of the ridership but lack the ability to capture the spatial distribution through bus lines, while the opposite holds for FCN.

Chapter 4 addresses the data imbalance issue by a generative adversarial network model to re-balance the number of major and minor instances in the training dataset and to improve the accuracy of the prediction model. Accurate estimation of bus ridership plays an important role in the planning and optimisation of the bus network and bus operations. Recent research has demonstrated that smart card data, using machine learning techniques, can be a promising approach for predicting the spatial and temporal patterns of bus boarding. However, the boarding from a specific stop at a given time

window is a rare event: most of the records denote negative (non-travelling) instances, and only a few are positive (travelling) instances. Such data imbalance in the smart card records can significantly reduce the efficiency and accuracy of machine learning models deployed for predicting hourly boarding numbers from a particular location. This motivates this chapter where I propose a deep generative adversarial network (Deep-GAN) model to generate dummy travelling instances and to produce a training dataset with more balanced travelling and non-travelling instances. The synthesised dataset is used in a deep neural network-based (DNN) model to predict the travelling and non-travelling instances from a particular stop in a given time window. The results show that addressing the data imbalance issue can significantly improve the performance of the predictive model and the dataset with the 1:5 imbalance rate (i.e. ration of travelling to non-travelling) best fits the actual profile of ridership. Further, a comparison of the performance of the Deep-GAN with other resampling methods (e.g. SMOTE and Random under-sampling) denote that the proposed method fits the profile of ridership better than other potential approaches.

Chapter 5 provides an overall summary of the thesis and give recommendations for future work.

Chapter 2

Incorporating weather conditions and travel history in estimating the alighting bus stops from smart card data

2.1. Introduction

'By 2030, provide access to safe, affordable, accessible and sustainable transport systems for all, notably by expanding public transport (UN, 2015)*.'*

The smart public transport system is an irreplaceable part of the 'Smart City' agenda (Ma et al., 2019). A well-planned and efficient bus system is a critical component of sustainable transport eco-system. The benefits of buses can be viewed from a range of different angles: (i) compared to cars, buses offer high capacity and low emission travel (Kwan and Hashim, 2016); (ii) buses are low-cost and quick to implement, relative to rail-based urban public transport systems such as metro; and (iii) bus operations have the flexibility to penetrate and respond to where and when the passenger demand is (Pei et al., 2019). However, many of the urban bus systems suffer from poor images of unreliability, crowding, bus bunching, and generally low level of services (Berrebi et al., 2015; Bordagaray et al., 2013). One of the important factors affecting their level of services and reliability is the temporal and spatial variability in the bus ridership distributions (Liu and Sinha, 2007; Sorratini et al., 2008). Understanding the factors

driving the bus passenger behaviour and accounting for them to accurately estimate bus ridership are therefore the basic foundation for planning and operating a good public transport system (Hollander and Liu, 2008; Ibarra-Rojas et al., 2015; Wu et al., 2017; Wu et al., 2019; Wu et al., 2016).

Bus ridership, or the origin-destination matrix of bus travel demand, is affected by many factors. Existing studies in the literature have tended to focus on the population density and bus service provision of the area (Johnson, 2003; Xie et al., 2019b), the socio-economic-employment characteristics of the traveller such as their car ownership, income, etc. (Paulley et al., 2006; Xie et al., 2019a). Bus passengers are exposed to outdoor weather environment during their travel, much more possible than car drivers and metro train users are. As a result, people may choose destinations and routes differently under different weather conditions (e.g. small but closer shop versus larger but farther supermarket; going straight home versus stopping at an intermediate location to run an errand; route 'without transfer' but long walk versus 'with transfer' but no walking, etc.). In terms of empirical evidence, there have been recent interests in the weather impact on bus ridership on the demand side, and how bus operating strategies should respond to weather conditions on the supply side (see the review by Böcker et al., 2013). For example, adverse weather is found to reduce the level of services of the bus system, while extreme weather (such as rainstorm and flood) could cause significant disruption to bus service (Hofmann and O'Mahony, 2005; Yin et al., 2016). Similarly, passengers' travel behaviour, in terms of whether to travel, trip timing, route, and destination, could also be influenced by the different weather conditions. On the underlying mechanisms of the weather impact on changes in OD trips, my intuition suggests a variety of scenarios where the different weather conditions may have on the choice of passenger's alighting stops. People may make destinations and routes differently under diverse weather conditions (e.g. small but closer shop versus larger but

further supermarket; going straight home versus stopping at an intermediate location to run an errand; route 'without transfer' but long walk versus 'with transfer' but no walking, etc.). Arana et al. (2014) show that wind and rain reduce trip-making, while mild temperature encourages passengers to travel. Aaheim and Hauge (2005) report that heavier precipitation and lower temperature shorten the distance people travel. Sabir (2011) points out that weather may change people's decision in the travel destination, especially for leisure travel. Liu et al. (2015) find that, in Sweden, both commuters and non-commuters are more willing to choose a closer destination in heavier rain. Hereby, we speculate that the passengers may change their alighting stops due to the different weather conditions, and we consider the ambient weather variables in our estimation.

Big data sources from the automatic data collection system can be utilised to support public transport planning and operation (Zannat and Choudhury, 2019; Zhang et al., 2018). For example, the automatic fare collection and automatic vehicle location systems offer new opportunity to understand the behaviour and patterns of bus ridership. With automatic data collection, the methods to estimate the ridership have been gradually shifted from the traditional manual survey, such as point check and ride check (Ceder, 2007), to data mining using readily available and large automatically collected data. There have been remarkable research interests recently in ways to extract the relevant and useful information from automatically collected data. Public transport users' smart card data from the automatic data collection system has been widely used as the most attractive resource to estimate bus ridership (Bagchi and White, 2005). Many of the bus systems, however, operate as a *single-tap* or *open* system, where passengers tap/swipe smart cards only at boarding, and thus we do not have information about their alighting. This raises challenges in using smart card data to directly derive bus origin-destination demand information, more specifically bus passengers alighting stops. Most of the existing research on this topic has so far only been able to estimate the alighting

stops of regular commuter bus passengers, by approximating the alighting stops of their morning commuting bus journey as being the boarding stops of their evening return bus trip. In this paper, we attempt to provide a machine-learning-based framework to estimate the alighting stops for general bus trips, including regular and non-regular bus journeys.

The remainder of this paper is structured as follows. Section 2.1.2 reviews the methods in estimating the bus ridership and introduces machine learning techniques used in mining automatically collected data. A review of the weather factors affecting bus ridership is also presented. Section 2.1.3 introduces the case study network and the open smart card data used in this paper and highlights the limitation of applying the existing methods (trip chaining, for example) to our case. A machine learning approach based on the recently developed gradient boosting decision tree (GBDT) algorithm is proposed in Section 2.2.4 to solve the multi-class classification problem of estimating the alighting stops for the trips. Section 2.2.5 describes the trip features used in the model and designs the experiments whose results are presented in Section 2.2.6. Finally, Section 2.7 summarises our findings and suggests future research interests.

2.2. Literature review, research gaps and proposed improvements

2.2.1. Bus ridership and alighting stop estimation using open smart card data

Passengers' travel history can be tracked by the smart card data and then used for inferring their travel behaviour and ridership (Pelletier et al., 2011). In the literature,

there are two main approaches to estimate bus ridership from the open smart card data: attraction rate and trip-chaining model (see the review by Li et al., 2018).

Briefly speaking, the attraction rate modelling estimates the attractiveness of a bus stop to the passenger, considering its boarding stop, the bus line of travel, and other relevant factors. Dou et al. (2007) propose a method to calculate the alighting probability at bus stops from the travel distance and passenger numbers. Another method in the attraction rate model is the reverse ridership method (Hou et al., 2012), which proposes that the proportion of the boarding passengers is equal to the proportion of the alighting passengers at the same stop in the reverse bus service. The attraction rate model can hence approximate the total bus passenger origin-destination ridership over a day, which is useful for long-term bus planning purposes. It is not, however, suitable to estimate the within-day (such as hourly) ridership which is critical for short-term or real-time bus operation and management. It is also not suitable for application at the individual smart card user level, which can be useful for policy testing purposes (e.g. testing the implication of a policy to provide fare discount for frequent travellers).

The second approach, trip-chaining model (Barry et al., 2002), uses open smart card data to estimate linked trips and uses the results to establish the associated alighting stops. This method has been applied in extensive studies in New York (Barry et al., 2002), Chicago (Zhao et al., 2007) and London (Gordon et al., 2013). The trip-chaining model makes two strong assumptions: (i) each passenger gets on-board at the station where he/she alighted at the last trip; and (ii) each passenger's daily final alighting stop is the same as his/her first boarding stop of the day (Barry et al., 2009). These assumptions put a limit on the applicability of the method. As summarised by Li et al. (2018), such a naïve trip-chaining model is not applicable to the following groups of passengers: (i) who use an untraceable mode of transport, for example taking a taxi on a leg of the journey; and (ii) who do not return to their origin stops. Since then, various studies have

been making improvements to this naïve trip-chaining model. For the unlinked trips (e.g. those which involve a different untraced mode of transport in between bus trips), Trépanier and colleagues (Trépanier and Chapleau, 2006; He and Trépanier, 2015) suggest using passengers' historic travel pattern, and they propose a density-based method using arrival time and distances corresponding to each potential stops to identify the probability of alighting at that stop. For the daily trips which do not go back to the first boarding stop, Munizaga et al. (2014) find that many midnight trips (between 0–2 am) belong to trip chains on the previous day, and they suggest distinguishing the day at 4 am to reduce missed trips in recognising the trip chains.

One of the key processes in trip-chaining based models is to identify the most likely alighting stop among possible stops in close proximation. Trépanier et al. (2007) search the possible alighting stops by minimising the distance to the boarding stop of the next trip. Nunes et al. (2016) define a threshold of distance by the transaction fares system with distance-based fare structures. Munizaga and Palma (2012) replace the distance by a generalised time, while Nassir et al. (2011) combine smart card records with a range of additional data sources, including bus timetable, automatic passenger counter and automatic vehicle location system, to identify the alighting stop of the last trip.

A common feature in these improved trip-chain models is that they rely on historical data to find the next boarding (alighting) stops. Studies using the attraction rate and trip-chaining models have so far been mainly based only on the smart card data, with some incorporating the network characteristics into the studies. In reality, there are many other factors that can affect the ridership choices made by the passengers, such as the effect of weather on passengers' habitual travel behaviour (see section 2.2.3 for details), special events, etc. This paper attempts to incorporate such weather-related factors in the estimation of bus ridership to address this research gap.

2.2.2. Data mining using machine learning technique

Although the development of automatic data collection system offers detailed data on various aspects of the public transport system, the abundance of available data challenges the traditional data mining methods such as classification, clustering, and regression analysis. Machine learning as a data mining method is shown to be able to handle high-dimensional and multivariate data in a complex, dynamic and even chaotic system, and to identify the patterns in the data and the relevant influential factors (Witten et al., 2016; Wu et al., 2016; Wu et al., 2020a).

Recently, there has been an increase in the number of studies trying to bring machine learning to the analysis of public transport data (see examples listed in Table 2.1). For example, Yu et al. (2011) apply several machine learning models: support vector machine, artificial neural network, k nearest neighbours algorithm and linear regression to predict the bus arrival time from the bus running time on different routes. Corman and Kecman (2018) build Bayesian networks to predict train delays in real-time from a live data stream. Meanwhile, how to use the data in the automatic fare collection system is also an interesting topic for many studies.

There are two types of automatic fare collection systems: the closed automatic fare collection system, which records both the boarding and alighting information, and the open automatic fare collection which records only the boarding information. For the closed (mostly metro) automatic fare collection system, there have been extensive studies applying machine learning to forecast the metro passenger flow from smart card data via the networks of hybrid empirical mode decomposition and back-propagation neural network (Wei and Chen, 2012), multiscale radial basis function network (Li et al., 2017) and long short-term memory neural networks (Liu et al., 2019b).

There are relatively limited studies of machine learning application to open automatic fare collection systems. Toqué et al. (2016) infer the alighting stops using the trip-chaining model to predict the origin-destination matrices at stop level in 15-minute windows using long short-term memory neural networks. Jung and Sohn (2017) develop a deep learning model to predict the alighting stops for each transaction, taking account of variables on the land-use near the boarding and candidate alighting stops. The key literature on machine learning applications on public transport research is summarised in Table 2.1.

Table 2.1 Selected literature on the applications of machine learning on public transport research.

Literatures	**Public transport modes**	**Targets**	**Machine learning models**	**Data resources**
Yu et al. (2011)	bus	arrival time	SVM, ANN, k-NN, LR	real-time traffic data
Corman and Kecman (2018)	train	delay	Bayesian network	scheduled and real timetable
Liu et al. (2019b)	metro	passenger flow	LSTM	closed smart card data; weather and holiday events
Li et al. (2017)	metro	passenger flow	MSRBF	closed smart card data
Wei and Chen (2012)	metro	passenger flow	EMD - BPN	closed smart card data; (holiday events)
Toqué et al. (2016)	bus	origin-destination matrix	LSTM	open smart card data
Jung and Sohn (2017)	bus	destination	Deep learning	open smart card data; land-use
This study	bus	alighting stops	GBDT	open smart card data; travel history; weather conditions

Our study proposes a machine learning method to estimate bus passengers' alighting stops from open smart card data where only the boarding stops are observed. Compared to the existing literature on the subject (Table 2.1), we employ an innovative new data mining approach, the gradient boosting decision tree (GBDT) model, to estimate the alighting stop for every bus trip recorded in the smart card data. Furthermore, we incorporate passengers travel history and their travelling environment – in terms of the ambient weather conditions, into the estimation. We examine the impact of these additional variables on the performance of the estimation.

2.2.3. Weather impacts on bus ridership

As noted in the Introduction, there has been existing research that established relationships between the varying weather conditions and overall bus ridership. Table 2.2 summarises the key literature that examines the weather impact on public transport ridership. Generally, precipitation is found to be one of the most important factors affecting bus ridership. Hofmann and O'Mahony (2005) show that the number of smart card trip records decreases with rainfall. Similar conclusion, drawn by Saneinejad et al. (2012), is that commuting trips of all modes, including buses and private cars, is negatively affected by precipitation. In contrast, Singhal et al. (2014) find that urban transit ridership increases on snowy days as the poor driving conditions tend to shift people from private cars. Additionally, Guo et al. (2007) investigate how rain and snow affect the public transport ridership and report that rain and snow tend to reduce the ridership for both bus and rail, but heavy snow might actually increase the rail ridership. The temperature and wind are other important sensory weather variables. Stover and McCormack (2012) report that in a cold climate, the ridership decreases as the temperature dropped and increases on warmer days. Similarly, Guo et al. (2007) discover a significant positive impact of temperature on transit ridership in warm

weather but find no correlation between ridership and temperature in cold weather conditions. They speculate that the impacts are not caused by the prevailing temperature but the temperature changes and human perceptions. Kashfi et al. (2015) report that temperature has an insignificant effect on the daily bus ridership. For the impacts of wind, Arana et al. (2014) show that the wind, together with rain, leads to a reduction in transit ridership. Singhal et al. (2014) also find a negative effect of wind on hourly subway ridership, but the effect is not significant on daily ridership. Guo et al. (2007) note that increasing wind speed reduces bus ridership, but it has a negligible impact on rail ridership. Besides these three weather variables (temperature, rainfall and wind), Zhou et al. (2017) analyse the impact of weather condition on bus and metro system together with relative humidity and air pressure. Their study reports that increase in humidity, wind and rainfall is generally associated with a certain degree of transit ridership decrease while their degree and the significance of the impact vary from one weather variable to another, and between weekday and weekend. Wei et al. (2019) consider the weather impact on weekday and weekend travel by bus, train and ferry. They find that, unsurprisingly, ferry is mostly affected by bad weather. Poor weather conditions do not appear to affect train journeys during the weekdays, but to reduce train journeys made during weekends. They find that bus trips are negatively affected by rainfall, but not affected by temperature.

The most common method to quantify the impact of weather on public transport ridership in these studies is regression modelling, and the most commonly used independent weather variables are temperature and precipitation. Guo et al. (2007) take into account discretised weather variables, such as heavy or light precipitation, and warm or cool temperatures. Zhou et al. (2017) replace the absolute value of weather variables by its deviation to the average condition. Wei et al. (2019) consider the

interplay of different weather variables in the model to formulate the complex impact of weather conditions on the ridership.

Table 2.2 Correlations between public transport travel demand and weather conditions.

↓ : the negative correlation; ↑ : the positive correlation; –: no correlation; ×: not discussed.

'Warm' and 'Cold' means in warm or cold season.

Literatures	**Public transport modes**	**Precipitation**		**Temperature**		**Wind**
		Warm	Cold	Warm	Cold	
Hofmann and O'Mahony (2005)	Bus	↓		×		×
Saneinejad et al. (2012)	All	↓		↑		×
Singhal et al. (2014)	Metro	↑		– (daily) ↑ (hourly)		– (daily) ↓ (hourly)
Stover and McCormack (2012)	Bus	↓		↑ (winter)		↓ (except summer)
Guo et al. (2007)	Bus	↓	↓	↑	–	↓
	Rail		↑ (heavy snow)			–
Kashfi et al. (2015)	Bus	–		–		×
Arana et al. (2014)	Bus and train	↓		↑		↓
Zhou et al. (2017)	Bus and metro	↓		↓		↓
Wei et al. (2019)	Bus	↓		–		–
	Train	– (weekday) ↓ (weekend)		– (weekday) ↓(weekend)		–
	Ferry	↓		↑ (weekday) – (weekend)		–

In this paper, we employ a machine learning method to independently evaluate the relative importance of different weather variables on bus ridership and consequently incorporate the important weather effects in the bus ridership prediction.

2.3. Data sources

2.3.1. Network description and data sources

In this study, we estimate the alighting stops of individual bus trips made using the smart card in the city of Changsha. The city, in the central south of China, is separated by the Xiangjiang River: the city's Central Business District (CBD) lies east of the river, while the west is principally residential zone. There are more than 200 bus lines in Changsha, operated by three bus companies. The study network, shown in Figure 2.1 is a subset of the Changsha bus network and includes seven bus lines, all operated by the same bus company. Despite it being only a sub-network of the city, the case network covers the key public transport interchanges in the city: the three bus terminals and two rail stations, as well as the three river crossings that connect the major geo-economical centres of the city. The seven bus lines are also representative in service characteristics, including long-distance and sparse-stop lines (Line 063 and 168), long-distance and dense-stop lines (Line 147), short-distance and sparse-stop lines (Line 006 and 007) and short-distance and dense-stop lines (Line 123 and 150).

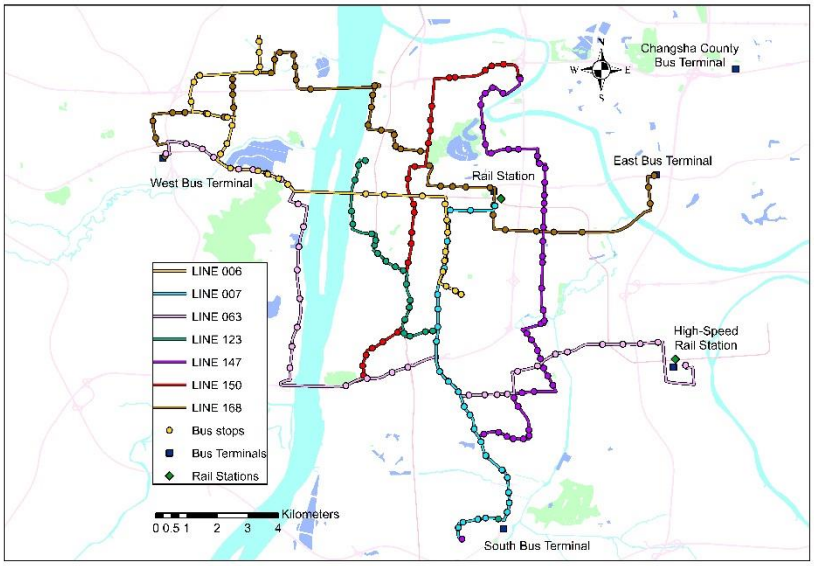

Figure 2.1 The case study bus network in Changsha, China.

Changsha's smart card system is a typical open automatic fare collection system, where passengers swipe cards only at boarding. The system records information on smart card ID, boarding time, line and the bus vehicle boarded. All buses in Changsha are fitted with GPS trackers, which record the vehicle ID, the longitude and latitude of the vehicle location every 10 sec.

Six months (April to September 2016) of smart card data is made available to this study, in which 12 days' data was missing. There are 17,159,076 smart card records in total or roughly 80000 records per day on average. The number of daily smart card records for the study period is shown in Figure 2.2. It shows that the ridership in holidays is markedly lower than that in working days. It is also noticeable that the four weeks between the end of July and mid-August have low ridership when the city typically experiences heatwaves with an average temperature around 35°C and maximum temperature of 40 °C. Besides the high temperature, July and August are also the school summer holiday, which may also contribute to a decrease in bus ridership.

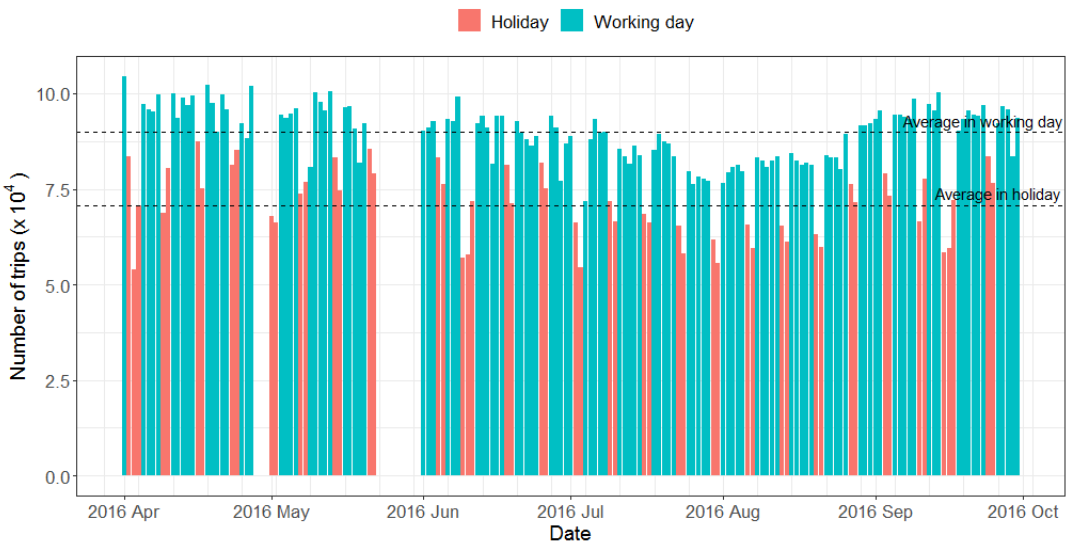

Figure 2.2 The number of smart card trips by day.

As part of a wider study of bus patronage, we are also interested in the effect of weather on passengers' travel demand. We acquired weather data for those six months. The weather data includes hourly measurements of temperature, precipitation, pressure, humidity, visibility, wind speed, and an indicator/register of the type of weather event of the hour. The weather events registered include, for example, clear, cloudy, thunderstorm, rain, etc. Figure 2.3 illustrates the hourly changes in weather events, overlaid with the number of smart card trips during that day (and separated for weekdays and holidays). What is also overlaid is the 'global' daily bus ridership, averaged over all the six months of the study period. It can be seen that, on clear days, the number of smart card trips is smaller than the global average. Combining the pattern in Figure 2.2, the high temperature and blistering sun may reduce trip-making. Before sunrise and after sunset when the temperature is lower, the ridership on those individual days is similar to the global average. On cloudy days, the ridership during the morning and afternoon peaks of the weekday is lower than the global average, while the ridership on the holiday day is delayed by two hours. Fog and haze have little or no impact on the bus passengers' travel behaviour in the working day, while the ridership on holiday is consistently reduced. Rain appears to have reduced bus ridership on both holiday and

working days. In Section 2.3.2, we present a statistical analysis on the significance of the different weather events on ridership.

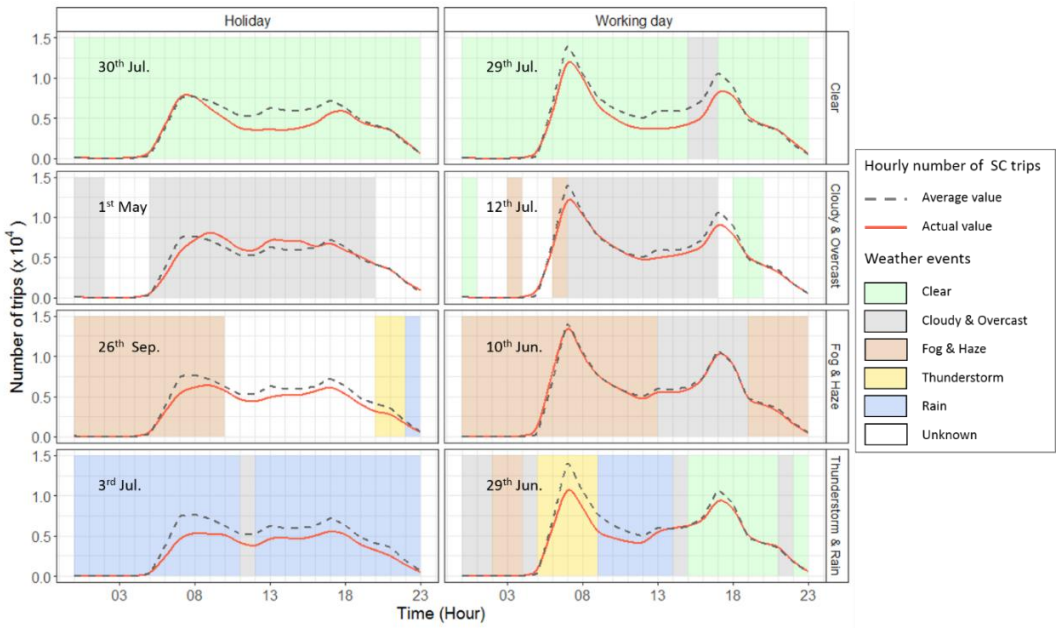

Figure 2.3 Hourly ridership in a week for the typical weather events.

2.3.2. Data pre-processing

We consider a passenger's travel from origin to destination as a journey, and each leg of their journey as a trip. To simplify the problem and clarify the data analysis process, the following assumptions are made: (A1) each passenger owns only one smart card, and each card can be only used by its owner; (A2) a journey that requires transfer among different lines is regarded as separate trips, each with its boarding and alighting stops. Although in practice, the same smart card may be used by family members or friends, by assuming (A1), we take each smart card user's travel history into consideration in estimating his/her alighting stops. With the above definition and assumptions, a trip is composed of a single pair of boarding and alighting stops.

The smart card records are firstly cleaned up based on the assumption (A1). If there are two or more records appear in the same vehicle at the same station in a very short time interval (defined as within 1 minute), the data is registered as repetitive records and

counted only once (the first record) in this study. 6.3% of the data is recorded as repetitive. The remaining smart card records are combined with the GPS tracking of bus vehicles to obtain the passengers' boarding bus stops. 9.9% of trips cannot be matched with the GPS record of the vehicle number, perhaps due to poor quality of the data. Then, we capture the timestamps when a bus enters and leaves a bus stop, and match the boarding time with these timestamps to find out the boarding stops. Based on this boarding-stop inference method, an additional 15.5% of trips whose boarding stops could not be inferred. In total, 31.7% of the original smart card records are not useful, and the remaining 68.3% of the data (with 11.7 million smart card records) is applied in this study.

In our proposed machine learning approach, the training set utilises the alighting stops as the label to teach the model how to do the classification, and the testing set also requires their alighting stops to validate the accuracy and performance of the model. However, with the open automatic fare collection system, the smart card records we have do not contain any alighting information. Here, we use the trip-chaining method (Wang et al., 2011) to synthetically generate the alighting stops for the trips in training and testing datasets and assume these as the 'real' alighting stops for our proposed method. Following this naïve trip-chaining method, the trips are categorised into the following four types:

- *Trips in a chain* (X1);
- *Segments in transfer journey excluding last one* (X2);
- *Last segment in the transfer journey* (X3); and
- *Other trips* (X4).

As presented in Table 2.3, types X1 and X2 trips account for 26.7% of the overall records. This percentage is much lower than the cases in previous studies, for example, 75% in London (Gordon et al., 2013), 70% in Chicago (Zhao et al., 2007) and 90% in New York (Barry et al., 2002). One possible reason for the low share (of X1 and X2 trips) could be that our study network is a subset of the Changsha bus network, covering only seven bus lines. It is quite possible that there are more trips of X1 and X2 types made using other bus lines (operated by different bus companies) that are not counted in our sample. This reflects the practical constraints imposed by the bus operating framework in Changsha, as well as in many other cities, where there is more than one bus company operating different bus lines in the city, and there is no central governing body (such as Transport for London) to combine and share the smart card data generated by the different companies. A consequence of not having full access to all smart card data in a city would lead to breaks during the trip chains and lower percentage of X1 and X2 types of trips.

Earlier, we saw in Figure 2.3 illustrations of the different weather events and their effect on overall bus ridership. Here, we examine statistically the significance of weather events on ridership of each type of trip (chain). We use the one-way analysis of variance to examine the significant differences in the number of trips made under different weather events; the results of the statistical analysis are presented in Table A.1 in Appendix A. We find that the all the p-values are less than 0.05, except those for the X3 and X4 types of trips, which proves that passengers, regardless of taking transfer trips or chaining trips, have a significantly different travel behaviour under different weather events. The results further support the hypothesis that the weather has a significant impact on trip chaining (X1 trips), and on trips that involve transfers (X2+X3, and X1+X2+X3).

Table 2.3 The smart card data records for the study network.

Type		The number of trips	Percentage
Invalid data	Repetitive trips	1085500	6.3%
	No GPS data	1698515	9.9%
	No boarding stop	2661630	15.5%
Cannot infer the alighting stops	X4	6580433	38.3%
	X3	548182	3.2%
Database for this study	X2	746202	4.3%
	X1	3838614	22.4%
Total		17159076	100.0%

Following the trip-chaining method by Wang et al. (2011), the alighting trips can be only inferred for trips of X1 and X2 types. It may be noted that some of the types X3 and X4 trips can also be used by the method proposed by He and Trépanier (2015). For consistency, however, only the trips in X1 and X2 are used in our study with the machine learning model.

2.4. A GBDT-based machine learning approach for alighting stop estimation

In this section, we propose a machine learning classification approach to identify the alighting stop of the trip from an open automatic fare collection system, where the alighting information of the passenger is not recorded. We incorporate each smart card user's travel history and the weather conditions into the machine learning estimation framework.

2.4.1. Notations

The following notations are adopted in this paper.

Table 2.4 Table of notations.

Notations	Description
trip	Vector containing the features and alighting stop of a trip
r	A feature representing a characteristic of the trip
r	The vector including all the features associated with the trip
V	The number of features employed in the model
d	Alighting stop of a trip
k	Index of an alighting stop
K	The number of alighting stops in the network
merror	Estimation error
M_{wrong}	The number of trips that estimate to the wrong alighting stops
M_{total}	The total number of trips in the dataset
t	Index of iteration
T	Maximum iteration
S_a	Training set
m	The number of data in the training set
$h(\cdot)$	Probability function of alighting at stops
R	Disjoint region that collectively covers all the trips
Notations	**Description**
j	Index of the region
J	The number of regions
c	Coefficient corresponding to regions and defining the boundaries of regions
$f(\cdot)$	Boost tree model
$L(\cdot)$	Loss function
$p(\cdot)$	Symmetric multiple logistic transform of the probability of alighting at stops
g	Decent direction

2.4.2. The machine learning estimation framework

Machine learning approach works by training the algorithm to optimise a certain performance criterion using large data samples (Alpaydin, 2014). In machine learning languages, one data record is called an 'instance'. An instance contains many observed (or model) 'features', and one 'target label' (or a class) to be estimated. The set of the observed features is called a 'feature vector'. The observed features considered in our estimations are described in Section 2.5.1. In our study, the target label is the alighting stop we want to estimate. The instance, which in our study represents a bus trip, is mathematically represented as:

$$trip = (\mathbf{r}, d) \tag{2.1}$$

where $\mathbf{r}$ denotes the vector of V observed features of the trip:

$$r = \{r_1, r_2, \cdots r_V\} \tag{2.2}$$

and d denotes the target label, i.e. the alighting stop of the trip:

$$d \in \{d_k \,|k = 1, 2, \cdots, K\} \tag{2.3}$$

where the set of d_k represents all the possible alighting stops (or classes), and K is the total number of stops (which is 306 in our case study). Thus, our problem of predicting alighting stops is a multi-class classification problem.

The first step in machine learning is to separate all the trips in smart card data into three datasets: training, verification and testing datasets. The training set is used to obtain a trained model; verification set is used to evaluate our trained model in the training process, while the trained and verified model is applied to the testing dataset to predict the alighting stops of the trips in the dataset. The performance of a trained model is measured in terms of an estimation error ($merror$), defined as:

$$merror = \frac{M_{wrong}}{M_{total}} \tag{2.4}$$

where M_{wrong} is the number of trips that is estimated to the wrong alighting stop and M_{total} is the total number of trips in the dataset. We calculate the *merror*s for each of the training and verification dataset, which are used as the stopping criterion after each iteration of the training process.

Each model is trained with a set of hyper-parameters of the machine learning model. A range of initial hyper-parameter values is tested, resulting in a range of different trained models and model estimation errors. The final selected trained model is the one with the minimum estimation error, which is then applied to the testing dataset for estimating the alighting bus stops for the individual trips in that dataset.

2.4.3. A multi-class GBDT algorithm

2.4.3.1. General framework

The training algorithm introduced in this study is the gradient boosting decision tree (GBDT) algorithm. Firstly proposed by Friedman (2001), the algorithm is based on the integration of statistical and machine learning methods. More specifically, since a single tree (as used in Friedman, 2001) is too weak to lead to an accurate result, GBDT uses a set of simple trees, in the form of a classification and regression tree, to calculate the results and draws the conclusion (i.e. to estimate the alighting stop of each trip in our model) together.

Unlike most of the applications of GBDT which have only a binary choice, the problem of alighting stop estimation belongs to the multi-class classification problem (MCCP). We outline in Figure 2.4 the main processes for such a multi-class GDBT algorithm. Since there are 306 possible stops in the label, we consider the alighting stops one by one. The multi-class classification problem of estimating the alighting stops is then transformed to a set of regression problems and used to build the classification and

regression tree which calculates the alighting probability at each stop. The bus stop with the highest alighting probability is chosen as the final estimation result, i.e. the most probable alighting stop.

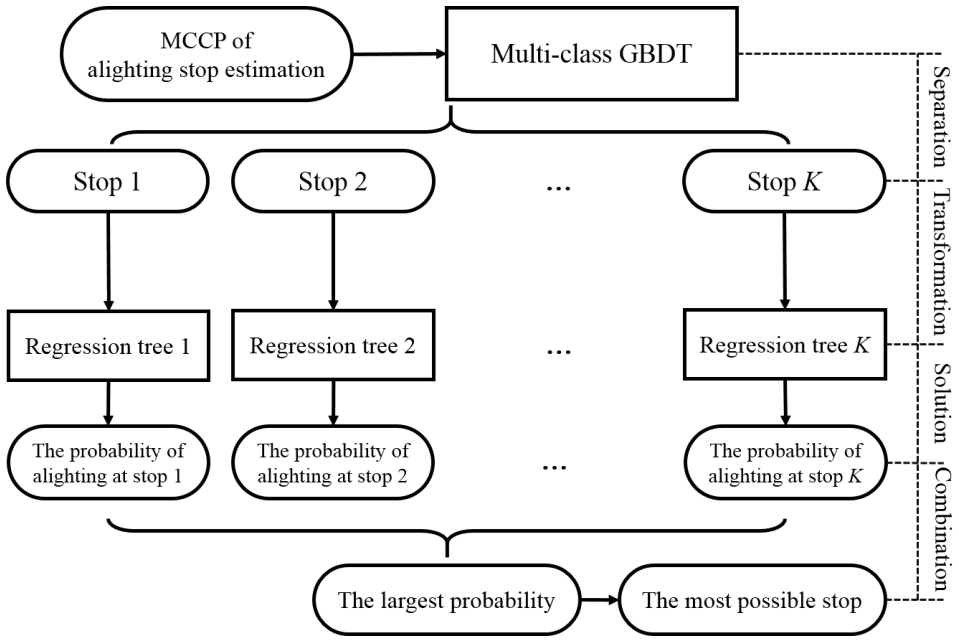

Figure 2.4 The processes of a multi-class GBDT.

2.4.3.2. Gradient boosting decision tree algorithm

The GBDT model combines a decision tree algorithm, a gradient updating algorithm, and a boosting algorithm, in an iterative process (outlined in Figure 2.5) to improve the training results.

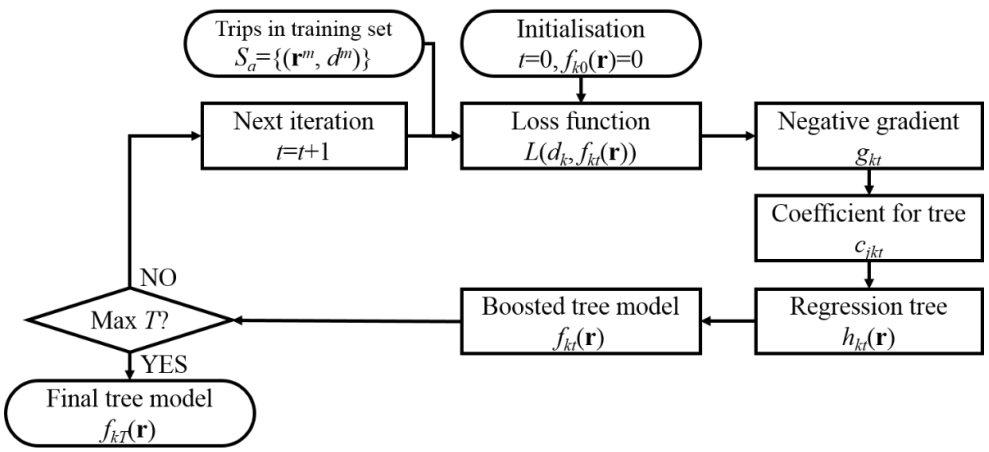

Figure 2.5 The detailed algorithm of GBDT model for a single stop *k*.

The classification and regression tree algorithm in GBDT is the most widely used decision tree model. Each internal node on the tree represents a test on a feature of the trip, while the branch represents the test output (represented in probability terms). The terminal nodes of the tree represent the alighting probability at the bus stop along the branch. Let $h_{kt}(\mathbf{r})$ denotes the estimation result for a trip $\mathbf{r}$ from a simple regression tree for stop k at the t^{th} iteration. The probability of alighting at the stop k at the t^{th} iteration is measured as the additive form:

$$h_{kt}\left(\mathbf{r}\right)=\sum_{j=1}^{J_{kt}} c_{jkt} I\left(\mathbf{r}\in R_{jkt}\right) \tag{2.5}$$

where R_{jkt} is the disjoint region j that collectively covers all the trips for stop k at iteration t, and J_{kt} is the number of regions for stop k at iteration t. These regions are represented by the terminal nodes of the tree. c_{jkt} is a coefficient corresponding to region j for stop k at iteration t, which defines the boundaries of the regions. The indicator function $I(\cdot)$ has the value 1 if the argument is true, and zero otherwise.

The idea of boosting is to identify ways to improve the simple trees. Let $f_t(\mathbf{r})$ denotes the estimation result of the boosted tree model after iteration t. Hence, the boosted tree model, or $f_T(\mathbf{r})$, can be obtained from:

$$f_{kT}\left(\mathbf{r}\right)=\sum_{t=1}^{T} h_{kT}\left(\mathbf{r}\right) \tag{2.6}$$

To increase the accuracy of the estimates, a loss function is being minimised step by step in the iterative GBDT process. A gradient algorithm is used to calculate the direction where the loss function decreases the most, and the gradient of numerical decent. The negative direction of the gradient refers to the direction where the loss function decreases the most. In GBDT, the loss function employs the log-likelihood loss function (Friedman, 2002):

$$L\left(d_k, f_k\left(\mathbf{r}\right)\right) = -\sum_{k=1}^{K} d_k \log_{10} p_k\left(\mathbf{r}\right) \tag{2.7}$$

where d denotes the real alighting stop; $f_k(\mathbf{r})$, calculated from Equation (2.6), is the probability of the trip estimated to alight at stop k; and d_k is the probability of the trip belonging to alighting stop k, where d_k equals to 1 if k is the real alighting stop, otherwise, 0.

Following the method of Friedman et al. (2000), we use the symmetric multiple logistic transform:

$$p_k\left(\mathbf{r}\right) = \frac{\exp\left(f_k\left(\mathbf{r}\right)\right)}{\sum_{l=1}^{K} \exp\left(f_l\left(\mathbf{r}\right)\right)} \tag{2.8}$$

Then the decent direction of trip m at stop k and iteration t can be calculated as:

$$g_{kt}^{m} = -\left[\frac{\partial L\left(d_l^m, f_{kt}\left(\mathbf{r}^m\right)\right)}{\partial f_{kt}\left(\mathbf{r}^m\right)}\right]_{f_{k,t}(\mathbf{r})=f_{k,t-1}(\mathbf{r})} = d_l^m - p_{k,t-1}\left(\mathbf{r}^m\right), l = 1, 2, \cdots, K \tag{2.9}$$

Equation (2.9) states that the error is the difference between the real probability of the alighting stop k that trip m maps and the corresponding estimated probability at iteration t-1.

Next, a new tree can be generated by following Equation (2.5), where the coefficient can be optimised as:

$$c_{jkt} = \underbrace{\arg\min}_{c_{jk}} \sum_{m=1}^{M} \sum_{k=1}^{K} L\left(d_k^m, f_{k,t-1}\left(\mathbf{r}\right)\right) + \sum_{j=1}^{J} c_{jk} I\left(\mathbf{r}^m \in R_{jt}\right) \tag{2.10}$$

Following Friedman et al. (2000), Equation (2.10) is approximated as:

$$c_{jkt} = \frac{K-1}{K} \frac{\sum_{\mathbf{r}^m \in R_{jkt}} g_{kt}^m}{\sum_{\mathbf{r}^m \in R_{jkt}} \left|g_{kt}^m\right|\left(1-\left|g_{kt}^m\right|\right)} \tag{2.11}$$

With Equation (2.11), a new regression tree for each stop can be generated by Equation (2.5), and the boosted tree model can be updated by using Equation (2.6):

$$f_{kt}(\mathbf{r}) = f_{k,t-1}(\mathbf{r}) + \sum_{j=1}^{J_{kt}} c_{jkt} I\left(\mathbf{r} \in R_{jkt}\right) \tag{2.12}$$

The iterative process continues until an empirical stopping criterion is met by comparing the *merror*s of training and verification dataset from Equation (2.4). In our case, a pre-specified number of iterations is reached.

Unlike many other machine learning methods, GBDT is able to evaluate the relative importance of the independent features of the trip. Since the depth of the tree is constrained by the hyper-parameter, the simple tree that is used in each iteration only includes a randomly chosen set of features (as opposed to all features). Hence, the frequency of the features used across all trees can be used to measure the relative importance.

2.4.4. Model evaluation

One measure of the performance of the model is the estimation error *merror* from Equation (4). However, it is a far too simple measurement to evaluate the machine learning models. Generally, a confusion matrix of measures, composed of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), is used to evaluate the performance of the binary classification model (Stehman, 1997). For our multi-class classification model, we introduce a confusion matrix for the estimated results of each alighting stop and then calculate evaluation indexes (Powers, 2011; Zhou, 2016). Table 2.5 presents the confusion matrix for a single alighting stop.

Table 2.5 The confusion matrix for the estimated results of a single alighting stop *k*.

Real alighting stop / Estimated alighting stop	k (positive)	Other stops except for k (negative)
k (positive)	TP_k	FP_k
Other stops except for k (negative)	FN_k	TN_k

Our evaluation indexes then include precision (*macro P*), recall (*macro R*) and F1 score (*macro F*1), as defined in equations (2.13 – 2.15). Precision and recall reflect the quality of the model in terms of the reliability for the results and the applicability for the sample. The F1 score, the harmonic mean of the precision and recall, measures and provides an overall performance of the model. The higher F1 scores indicate more superior models.

$$macro\ P = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_k}{TP_k + FP_k} \tag{2.13}$$

$$macro\ R = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_k}{TP_k + FN_k} \tag{2.14}$$

$$macro\ F1 = 2\cdot\frac{macro\ P \cdot macro\ R}{macro\ P + macro\ R} \tag{2.15}$$

2.5. Feature selection and experiment designs

In this section, we introduce the features selected that characterise the trips, and the machine learning experiments designed to evaluate the relative performances of the algorithms and data features.

2.5.1. Feature selection

Each trip in this study contains 18 observed features, denoted as r_1 to r_{18} in Equation (2), and the one target label, d; these are listed in Table 2.6. The observed features contain three groups of data: (i) the basic bus trip information as recorded by the smart card and the boarding information as inferred from the GPS records of the bus services; (ii) the smart-card user's recent travel history, also extracted from the historical smart card data; and (iii) the ambient weather data for the trips taken in (i). The temporary features and boarding stops are the necessary information from the smart card data and often used in the previous studies for the bus ridership estimation. To investigate the regularity of travels and describe the travelling preference, we introduce features about the passengers' recent travel history. As noted in the Introduction, weather can impact on the travel destinations (Sabir, 2011), we introduce independent weather variables in the estimation of alighting stops.

Table 2.6 The selected model features and the target label.

Feature groups	Features	Types	Investigated range
Basic smart card information	Month	Discrete	4 – 9 [for April to September]
	Day	Discrete	1, 2, …, 31 [day]
	Hour	Discrete	0, 1, 2, …, 23 [hrs]
	Days of week	Categorised	Mon., Tues., Wed., Thurs., Fri., Sat., Sun.
	Holiday	Binary	0: working day; 1: holiday
	Boarding stop ID	Nominal	060101, 060102, etc.
	Boarding line	Nominal	6, 7, 63, 123, 147, 150, 168

Feature groups	Features	Types	Investigated range
Travel history	Number of trips on the previous day	Discrete	0, 1, 2, …
	Number of trips in the same hour on the previous day	Discrete	0, 1, 2, …
	Number of trips on all the previous 7 days	Discrete	0, 1, 2, …
	Number of trips on the same day of the last week	Discrete	0, 1, 2, …
	Number of trips in the same hour on the same day of last week	Discrete	0, 1, 2, …
Weather conditions	Temperature	Continuous	-6 - 40 [°C]
	Precipitation	Continuous	0 - 58mm
	Humidity	Continuous	0 - 100 [%]
	Visibility	Continuous	0 - 10 [km]
	Wind speed	Continuous	0 - 10 [mph]
	Weather events	Categorised	Clear, rainy, misty, cloudy, overcast, unknown.
Model label	Alighting stop IDs	Nominal	060101, 060102, etc.

2.5.2. Experimental design

As GBDT is a relatively new machine learning algorithm, we adopt two other classic algorithms, multinomial logistic regression (MLR) and neural network (NN), to compare their relative performances. MLR and NN are the two most popular algorithms used in machine learning approaches. Both have been used in a variety of transport applications, e.g. traffic forecasting, travel mode choice modelling and trip distribution modelling (Karlaftis and Vlahogianni, 2011). The hyper-parameters of the

GBDT and NN algorithms are set as inputs for the machine learning model. Table 2.7 displays the different initial settings of the hyper-parameters for the algorithms during the training process. The training process tries the values of hyper-parameters in Table 2.7 and applies trained model to the verification data. After several attempts, we find the best performed model with a group of hyper-parameter values.

Table 2.7 The initial setting of the hyper-parameters during the training process.

The next two groups of experiments with GBDT follow this initial setting.

Hyper-parameters	GBDT	NN
Learning rate or step-size	0.0005,0.001, 0.005, 0.01, 0.05, 0.1	0.0005,0.001, 0.005, 0.01, 0.05, 0.1
Maximum depth of each tree	3,5,8,10,12,15	-
Fraction of data for training next tree	0.2, 0.4, 0.6, 0.8	-

We conduct six experiments, with increasing total number of trips in the training and verification set, while keeping the same dataset as the testing set. Table 2.8 lists the details of the training and verification datasets used for the six experiments. All the six experiments employ the same testing dataset, the trips made on 30th September 2016. The other data are combined and used as training and verification data (depending on the sample sizes). In each of the six samples, 30% of the combined training and verification data is chosen randomly as the verification data and the rest 70% as the training data. From Samples 1 to 6, the number of days and data in the combined training and verification data increases. These experiments are designed to illustrate the relationship between the size of the training set and the accuracy of the results.

Table 2.8 Data sample for the experiments.

Sample 4 – 6 exclude days when the data was missing.

Experiments	The training and verification data		The testing data	
	Days	Number of records	Day	Number of records
Sample 1	01-29/09/2016 (23 days)	807136	30/09/2016 (1 day)	22602
Sample 2	01/08 - 29/09/2016 (54 days)	1576132		
Sample 3	01/07 - 29/09/2016 (85 days)	2324989		
Sample 4	08/06 - 29/09/2016 (108 days)	2954072		
Sample 5	08/05 - 29/09/2016 (123 days)	3388117		
Sample 6	08/04 - 29/09/2016 (143 days)	3983788		

As introduced in Section 2.1, in this study, we are interested in the weather conditions and travel history of passengers' behaviour. To test the hypothesis, we set up several GBDT models with four different combinations of feature groups (dubbed as FG 1 to 4). The feature groups and the setting of hyper-parameters are displayed in Table 2.9. All four experiments use the dataset of Sample 6 experiment. FG 1 uses only the basic smart card information, similar to many previous methods. FG 2 adds only the travel history to FG 1, and FG 3 adds only the weather variables to FG1, while FG 4 includes the full set of features proposed in Table 2.6 (i.e. adds both travel history and weather conditions to FG 1). The experiments are designed to help us understand the effect of different groups of features.

Table 2.9 Experimental designs with different feature groups (FG).

Experiments	Basic smart card information	Travel history	Weather conditions
FG 1	✓		
FG 2	✓	✓	
FG 3	✓		✓
FG 4	✓	✓	✓

2.6. Model results

2.6.1. Model comparison

Sample 6 and feature group FG 4 are applied to the GBDT model introduced in this paper, and to the MLR and NN models. For the GBDT and NN model, the final set of the hyper-parameters are displayed in Table 2.10.

Table 2.10 Values of the hyper-parameters in GBDT and NN.

Hyper-parameters	GBDT	NN
Learning rate or step-size	0.005	0.001
Iteration	150	30
Maximum depth of each tree	8	-
The fraction of data for training next tree	0.4	-
Number of nodes in layers	-	(342,333,333,306)
Activation function for hidden layers	-	Sigmoid
Dropout rate	-	0.3
Activation function for output layer	-	SoftMax

The relative estimation power of the three models, as measured by their precision (*macro P*), recall (*macro R*) and F1 score (*macro F1*), are illustrated in Figure 2.6. The

F1 score of a random classification is used as the baseline for comparison, as is indicated as the 'baseline' in Figure 2.6. It can be seen that the F1 scores of all these three algorithms are higher than that of the baseline, suggesting all three models are theoretically acceptable, while GBDT has the best performance according to *macro P* and *macro F*1. Looking at the precision and recall of the model estimations, we can see that the values of *macro P* are always higher than *macro R* for their respective machine learning algorithm. This suggests that the estimation accuracy in all three models is better than their recall power. GBDT has the highest prevision accuracy, while NN has slightly higher recall power than GBDT. Overall, GBDT performs the best, and its prediction power is higher in accuracy than in its comprehensiveness.

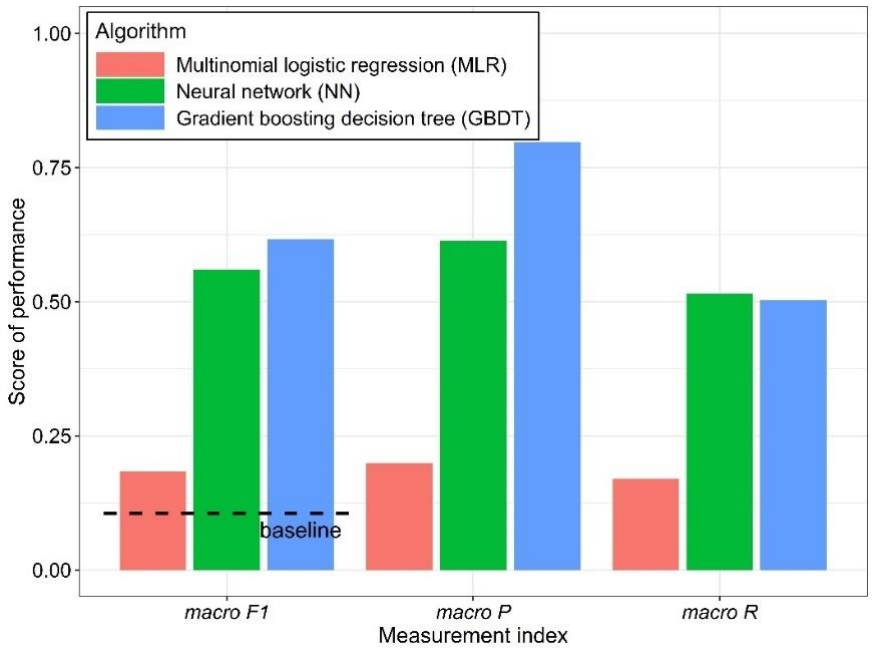

Figure 2.6 Comparison of the performance of the three training algorithms.

2.6.2. Effect of the training data size

The different training datasets, as defined in Table 2.8, are applied to the GBDT model with feature group FG 4. The values of hyper-parameters are displayed in Table 2.11.

Table 2.11 Values of the hyper–parameters in Sample 1 to Sample 6.

Setting	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
Learning rate	0.001	0.001	0.001	0.005	0.001	0.005
Iteration	70	70	100	120	110	150
Maximum depth of each tree	6	8	8	5	7	8
Fraction of data for training next tree	0.5	0.7	0.6	0.7	0.5	0.4

The prediction measures are shown in Figure 2.7. We can see that, in general, increasing the training data size improves the prediction power. The most significant improvement happens between experiment Sample 1 and Sample 2, while the improvements gradually become smaller as the total sample sizes get larger. The level of precision is universally higher than the recall. It indicates that this model does better in the precision estimation than in the extensive estimation.

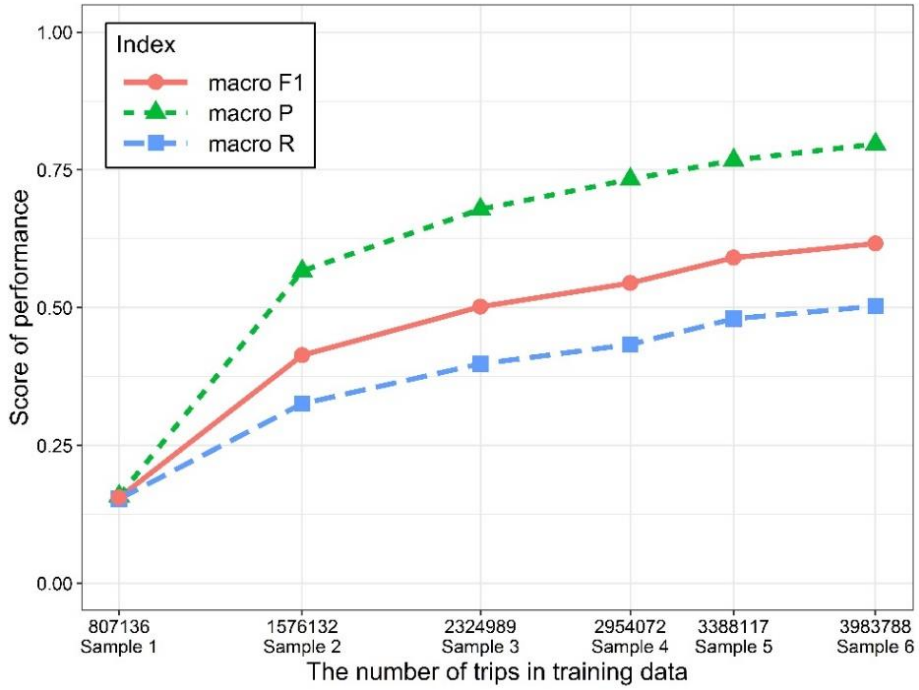

Figure 2.7 The performance measurements of the model in different size of the training set.

2.6.3. Impact of weather condition and travel history

We illustrate the impacts of including the weather variables and historical trips on our models by comparing the results of the four groups defined in Table 2.9. The final value of the hyper-parameters of each group is presented in Table 2.12.

Table 2.12 Values of the hyper–parameters for the experiments with different feature groups.

Experiments	**FG 1**	**FG 2**	**FG 3**	**FG 4**
	Basic smart card information	FG 1 + travel history	FG1 + weather conditions	FG1 + travel history + weather conditions
Learning rate	0.05	0.05	0.005	0.005
Iteration	80	80	120	150
Maximum depth of each tree	3	5	5	8
Fraction of data for training next tree	0.6	0.5	0.6	0.4

In Figure 2.8, from FG 1 to 4, as reflected in the F1 scores, we see improvements in the performances of the models. Between FG 2 and 3, the improvement from adding historical trips is less compared to the improvement resulted from adding the weather conditions. We speculate that this is because the information about historical trips captures the regularity of the behaviour, but the travel behaviour of the passengers are affected more by the changing weather conditions. FG 4 has the best F1 scores indicating that including both travel history and weather leads to the best performance.

Looking at the precision and recall sides of each model, the main increase from FG 1 to 2 is in the recall index (*macro R*). The similar situation occurs from FG 3 to 4. When we remove the historical trips out of the model (from FG 2 to 3), the recall ability of

the model reduces slightly. However, there is a significant increase in precision from FG 2 to 3. The results suggest that the two groups of features can improve the model in different ways. Principally, the features about the historical trips improve the comprehensiveness of the model and the weather variable makes the estimation more accurate.

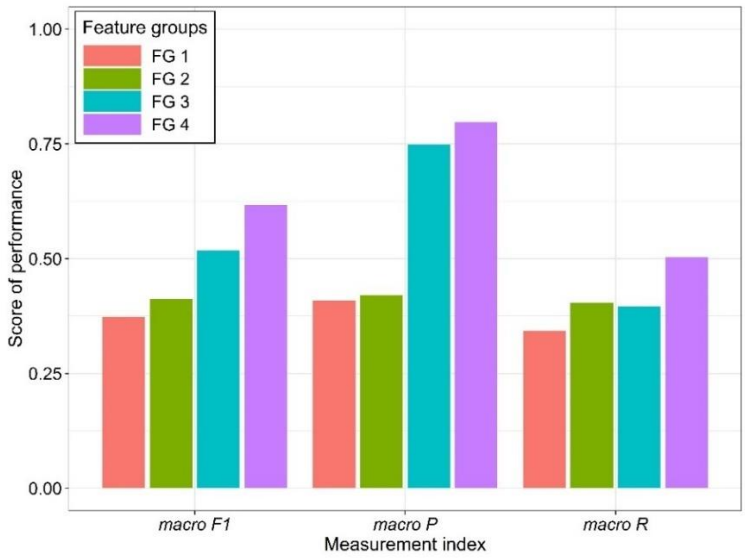

Figure 2.8 Evaluation of the impacts of different feature groups.

2.6.4. Relative importance of feature variables

We capture the relative importance of the features in models FG 1 to 4 in Figure 2.9. It can be seen in Figure 2.9(a) that, for FG 1(when only considering smart card data), the boarding stops is the most significant feature. The month and day are not significant features. The feature of holiday is the least significant feature, for which the day in a week contains the information on holiday to some extent. For FG 2 (as seen in Figure 2.9(b)), almost all the features about historical trips score low on impact. In Figure 2.9(c) for FG 3, the boarding stop scores the highest followed by the weather events with the temperature being the most important of the group. However, the importance of other quantitative weather features, i.e. wind speed, humidity, visibility and precipitation, is not as significant. This may imply that where passengers decide to alight is not

influenced by their qualitative cognition (e.g. whether it is rainy or not) than by the quantitative information (e.g. how much the precipitation is). This is also reflected in practice how bus companies adjust their service frequency under different weather event, i.e. they provide more frequent bus services on a rainy day regardless of the level of precipitation. Even if the bus company cannot respond to individual weather variables, e.g. humidity and visibility, our study suggests that the simple register of a 'weather event' would improve the origin–destination demand estimation and better (re)scheduling of their bus services. In Figure 2.9(d) for FG 4, we can see that the weather condition is a much more important group of features than the travel history in the model. This reinforces the findings drawn in Section 2.6.3. Thus, based on the importance analysis and performance measurement, FG 4 (the model including all the features) is demonstrated as the best model.

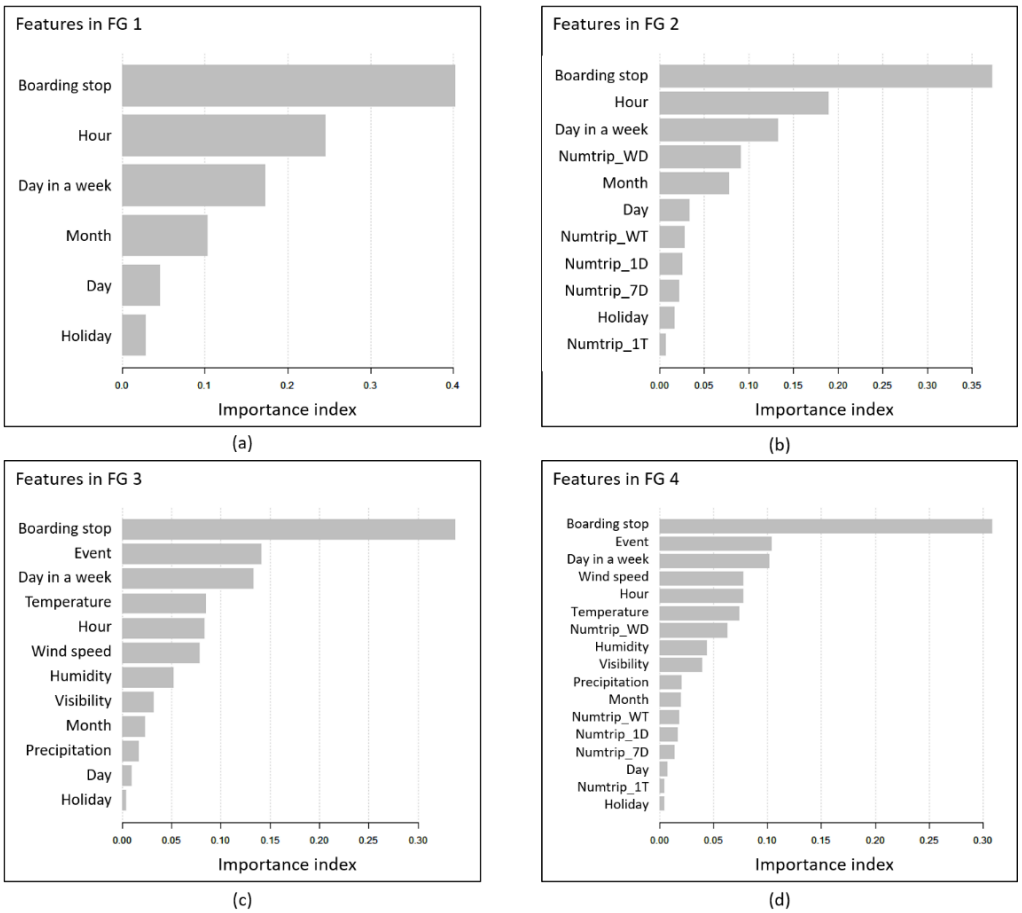

Figure 2.9 Ranking of the feature importance in different feature group experiments.

2.6.5. Ridership estimation

For bus planning, the overall demand (and distributions) of bus ridership is the most critical factor to consider. In this section, we apply our two trained models with and without weather features (FG 4 versus FG 2) to the test dataset, to predict the following aggregated bus ridership:

- The number of alighting passengers at each station,
- The load-profile and max load on each line.

The models with feature groups FG2 and FG4 are used in the prediction, and the predicted alighting stops are compared with the 'true' alighting stops as inferred from the trip-chaining model (Wang et al., 2011). We utilise the *GEH* Statistic (DfT, 1996) to compare the difference between the estimated and true alighting numbers at bus stops, which is formulated as:

$$GEH = \pm\sqrt{\frac{2\left(N_e - N_r\right)^2}{N_e + N_r}} \tag{2.16}$$

where N_e and N_r represent the number of correctly estimated and true alighting stops, respectively. Additionally, we add the signs to represent when $N_e > N_r$ (positive) and when $N_e < N_r$ (negative). Figure 2.10 presents the *GEH* statistics for seven bus lines (in both directions). In general, an absolute value of *GEH* less than 5 is considered acceptable (DfT, 1996).

Overall, 98% of the alighting stops have a *GEH* value less than 5, suggesting that the estimation accuracy is high. There are six stops for FG 2 and seven stops for FG 4 with *GEH* value between 5 and 10, while four stops in both FG 2 and FG 4 have *GEH* values

greater than 10. We find that most of those stops (with *GEH* higher than 5) are the last stop of the bus route. It is possible that the accumulation of the errors at intermediate stops leads to those large errors at the last stops. Besides, in the middle of Line 150 outbound services, there are two stops with high *GEH*s. We take Stop 16 in the model with FG 4 as an example: the true number of alighting is 7, while the weather-included model (FG 4) estimates that the number of alighting at this stop is 36. So, a small base number might cause a higher *GEH* value. Another possibility is that the trips alighting at this stop only make up 1.8% of the total training data. This load imbalance might also cause inaccuracy in the estimation. It is worth noting that the *GEH*s of Line 63, 123, 147 and 168 are near 0, suggesting that both of the models with FG 2 and FG 4 produce accurate matches in the alighting numbers at bus stops along these four bus lines. Furthermore, comparing the two models, 60% of stops have lower *GEH* in FG 4 than in FG 2, suggesting that including weather conditions help our machine learning model estimate more accurate alighting stops.

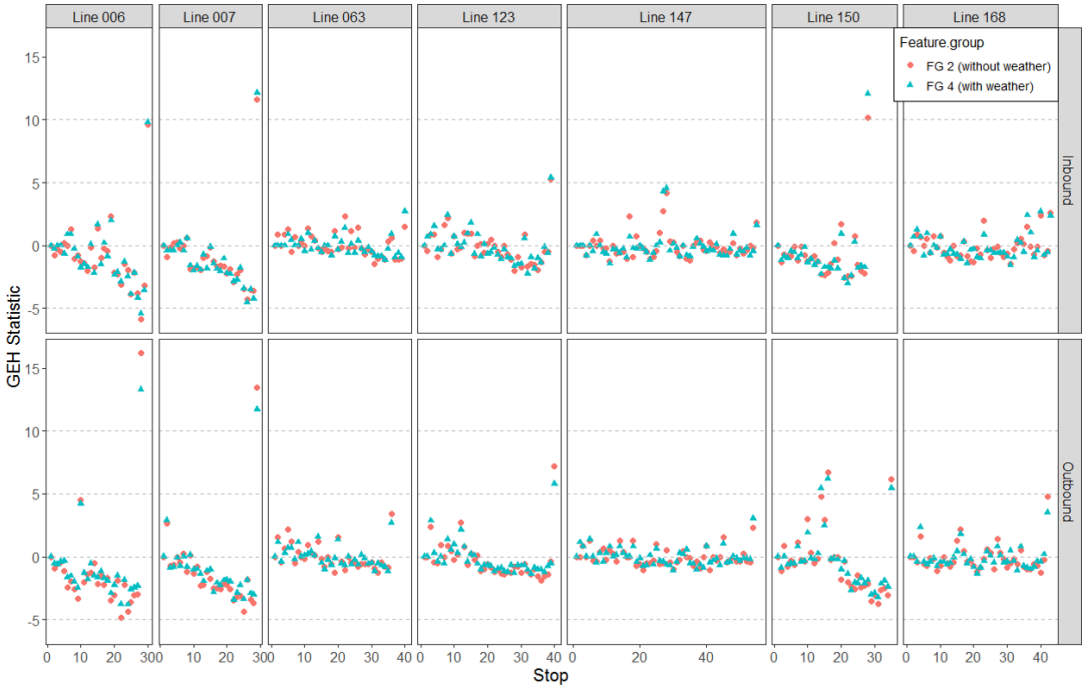

Figure 2.10 The GEH statistic of the alighting number at each station.

Figure 2.11 shows the ground truth and estimated ridership with FG 4 and FG 2, the two trained machine learning models with and without weather variables. As seen in the figure, the estimated load-profile has similar profiles as the ground truth and correctly matches the max-load stop in the ground truth with little differences in the absolute value of the maximum load. This is especially the case for Line 63, 123, 147 and 168, which have fewer passengers and which get an almost perfect matching. Although both models reflect the ground truth reasonably well, the model containing the weather variables (FG 4) is closer to the ground truth. Again, this comparison confirms that including weather variables makes the ridership estimation more accurate.

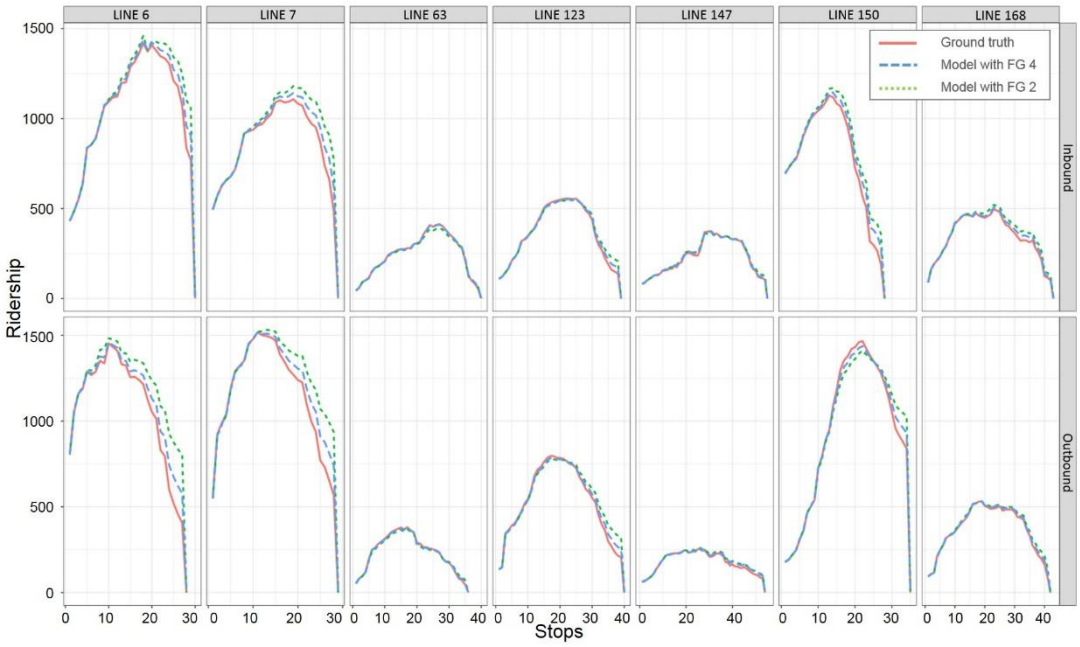

Figure 2.11 The one-day load-profile of each service in ground truth, weather-included model and weather-excluded model.

Additionally, the significant errors occur at the downstream stops of each bus services, which can be attributed to accumulation effects. We take a closer look at the outbound services of Line 6 and 147 to gain better insights about the accumulation errors. Line 6 has the largest error, while there are few errors in the latter one. Having a look at these two services in Figure 2.10, most stops in Line 6 has negative errors, and the only two

positive errors are in the middle and at the end, respectively. With accumulating the negative error, the increasing number of passengers are counted on board. So, the estimated ridership increases by these errors. The difference becomes larger and larger until the final positive error corrects the previous accumulated negative errors. However, the situation in Line 147 is different. The negative and positive errors occur alternately so that the following error can correct the previous errors in time, and the errors are not accumulated.

2.7. Discussion and Conclusion

Developing smart public transport system is a vital task in building sustainable cities. Understanding passengers' origin–destination and travel pattern are of great importance to improve the level of services and attractiveness of buses.

This study proposes a machine learning model with advanced gradient-boosting decision tree (GBDT) solution algorithm to estimate the alighting stops from the smart card logs of an open automatic fare collection system. We explicitly incorporate features that represent weather conditions and information of the individual's travel history in the model, so the estimation is not only based on the characteristics of the trip itself but also referring to the impacts of the ambient environment and the passengers' habitual travel behaviour.

To illustrate the performance of our proposed method, we conduct three comparative studies: (i) GBDT method vs two commonly used machine learning models; (ii) the size of the training dataset; and (iii) the inclusion (or not) of weather conditions and travel history in the estimation model. The results show that the machine learning method can accurately estimate the alighting stops from smart card data and that GBDT performs better than NN and MLR overall, and in particular from the view of precision.

Intuitively, increasing the size of training dataset improves the estimation accuracy. However, we discover that there is less improvement after a certain point (in our case, 3-month data in training dataset). The results also confirm that we can obtain a more accurate estimation when considering more features in the model, although the effect of the features varies. Weather conditions improve the accuracy (in precision), and historical trips improve the comprehensiveness (in recall). Additionally, the high ranking of the feature, weather events, and its significant contribution in increasing the precision of the GBDT model highlights that the effect of this variable is worthy of detailed testing when analysing and predicting passengers' decision of alighting stops.

Whilst the model trained in this paper with smart card data from Changsha is only applicable to this specific study network, the proposed GDBT framework is generic and can be applied to other smart card systems (open or closed): firstly using the smart card data to obtain a trained model, and then apply the trained model to predict bus ridership in the near future where travel conditions (such as weather conditions) can be readily predicted. Even the application in different cities can customise their model by easily adding or deleting the features in the model.

This study can also be used to predict the alighting stops in short-term as opposed to long-term trend prediction. The short-term prediction emphasises on the detailed value and minor changes (dynamics) and leverages the availability of accurate short-term weather forecasts. The target application is to make minor adjustments in the schedule of high-frequency bus services.

Overall, this paper makes new advances in these main aspects. Firstly, we employ a machine learning model with that GBDT algorithm in bus data mining, a novel technique in processing the massive smart card data. Secondly, our method is general and applicable to individual bus trips made by regular and irregular passengers as recorded in smart card data and fills the gap of the trip-chaining model, which requires

the identification of an unbroken trip chain for every smart card user. Third, we incorporate the impacts of weather conditions and travel history in the estimation of detailed origin-destination and ridership. Our model estimates the alighting stops for each smart card log, making it possible to readily compute the origin-destination-based load-profile for each bus line, important baseline information for planning more attractive bus services for the public.

We conclude this paper by critically examining the limitations of the current study. By its very nature, the true alighting stops of the open smart card system are not known, and we did not have access to an alternate source of ground truth data. Rather, we only have access to smartcard data from one bus company and use the naïve trip-chaining method to generate/obtain an estimated 'ground-truth'. Lack of validation from the real data is a limitation of this study. As a result, this study uses the alighting stops inferred by trip-chaining method as the labels to train and test the ML algorithm. Due to the absence of true alighting stops, the model cannot be examined by the ground truth. Lack of validation from the real data is a limitation of this study. If the smart card data with both boarding and alighting stops is accessible, the error between true alighting stops and estimated alighting stops can be tested. Further verification should tell us whether the trips labelled by trip chaining can be used to train the model. Also, validation against true alighting stops collected from surveys and/or inferred from other data sources (e.g. video recordings) will help verify the assumption and support the machine learning model estimation process. Additionally, this paper, in fact, applies the proposed machine learning model on the trip chains (X1) and part of transfer trips (X2) and we are not able to consider travellers who change their travel modes under different weather conditions. This could potentially lead to survivor bias for the model, as a result of the limited availability of data sources. As more multi-source data, such as bus video recordings and automatic passenger counted data, becomes readily available in the

future, they open opportunities for data fusion and new models for estimating bus ridership.

Further, due to the limitation of data accessibility, our case study is a subset bus network in Changsha, which may have caused the underlying self-selection bias. As stated in Section 2.3.2, it is common for more than one bus companies to operate in a city and they do not share their data. In our study, we try to reduce the bias by selecting representative bus lines in the city (discussed in Section 2.3.1). However, this still leaves a certain level of self-selection bias, which we cannot completely avoid. It is worth further investigation to compare the deviation of the models trained by the data from the whole and subset network.

Our model has 306 stops in the study sub-network of Changsha. In larger networks, the number of bus stops can be very big. A large number of stops can cause difficulty in training a good model; this is a general challenge for multi-class machine learning problem. As far as we are aware that machine learning methods have only been applied to cases with limited stops (classes), for example, Jung and Sohn (2017) consider only five candidate alighting stops in their model. One possible advance might be to separate the trips by bus lines and to build machine learning models line by line. This reduces the number of candidate stops (classes) in each model. However, the increased number of models are likely to make it more difficult for bus companies to use them, and to coordinate the different changes in service schedule at a network level.

In the study, we select GBDT as the predictive algorithm and compared to other two machine learning models. There are still various models that are good at different domains. Understanding how these models will perform in this work is a valuable investigation in the future.The current study is only the first step in applying machine learning techniques to estimate bus ridership form open smart card data. We believe that there will be more efforts and gain that can be made from using machine learning

techniques to gather passenger origin–destination demand and ridership information to support developing a sustainable public transport system.

Chapter 3

Multi-stage deep learning approaches to predicting boarding behaviour of bus passengers

3.1. Introduction

The rapid development of urbanisation at one hand brings convenience to people's lives but, on the other hand, causes problems such as traffic congestion and leads to an increase in energy demand and environmental pollution (Kwan and Hashim, 2016; AlRukaibi and AlKheder, 2019). As a sustainable transport mode, a well-planned public transport can play a key role in reducing transport externalities (Yao et al., 2020). With a high degree of accessibility and low implementation costs, bus transport is the most significant mode in urban public transport, accounting for 50% of trips made by all public transport modes in England (Department for Transport, 2019) and 45% in Beijing (Beijing Transport Institute, 2019). However, the bus system suffers from a poor image of unreliable services, crowding, bus bunching, and low level-of-services (Berrebi et al., 2015; Bordagaray et al., 2013). Coupled with the rise of demand-response travel options such as Uber, bus ridership has been in decline in recent years (Department for Transport, 2019). To move towards a smart and sustainable city, an important goal in transport planning is to encourage and attract more people to travel by public transport (Ma et al., 2019; Tong, 2019). One way to sustain or to increase bus patronage is to provide more reliable bus system based on sound service planning and management.

The first and most important factor in bus planning is the bus ridership, which affects reliability and level of crowding (Liu and Sinha, 2007; Sorratini et al., 2008; Fonzone et al., 2015) as well as pricing (Sakai et al., 2017; Xu et al., 2018). Understanding travel patterns of bus passengers and accurately predicting ridership are therefore essential foundations for planning and operating a sound bus system (Hollander and Liu, 2008; Wu et al., 2017; Wu et al., 2019).

It is well-known that the decision-making in car-drivers' driving behaviour is related to their past travelling behaviour and habit formation (Goldenbeld et al., 2000). That daily human mobility can be reproduced by tracking previous trips (Schneider et al., 2013). These theories can also be extended to public transport users who rely on their experience when making decisions, and who are more inclined to take their regular travel pattern. For example, if a typical commuter always travels from home to office in the morning and back from office to home in the evening, he/she may regularly take the same bus line at the same bus stop at the same time. Therefore, tracking their travel history helps us to identify and predict their regular travel pattern, such as when and where the trip is made.

Travel behaviour may be regular, but not static (fixed). Travel behaviour is under the influence of many factors of the public transport system itself, such as travel time, headways, reliability and cost, and external factors such as weather conditions (Sierpiński, 2016). Level-of-service and reliability are the most critical factors. Passenger likes to take high-quality and reliable bus service. However, some other things, such as adverse weather conditions, may lead to irregular headway and longer travel time, which reduces reliability and level-of-service. So, when the travelling scenario changes, passengers are likely to change their travel choice. An increasing number of studies has claimed that weather conditions have a significant impact on the public transport system (Böcker et al., 2013), alongside level-of-service attributes such as travel time, cost,

waiting time, the number of transfers and other network characteristics. For example, adverse weather conditions may result in lower reliability, longer travel time or even disrupting the public transport services (Ma et al., 2015b; Koetse and Rietveld, 2009). Exclusively focusing on the buses, previous research has highlighted the varying impacts of weather on public transport. For example, Wei et al. (2019) report that rain and snow have a negative impact on bus ridership in Brisbane. There is, however, the different impact of the same weather conditions. Stover and McCormack (2012) show that temperature has a positive impact on bus ridership in Washington, U.S., while Zhou et al. (2017) find the influence of high temperature on bus ridership being negative in Shenzhen, China. What is certain, nonetheless, is that weather conditions play a non-negligible role in bus users travel decision, which in turn influences the overall travel patterns and demand levels.

This study aims to predict the boarding behaviour of bus passengers at the individual level from smart card data. The boarding behaviour refers to whether to have a bus trip and which bus stop and bus line to use. As stated above, the regular boarding behaviour can be tracked by their travel history. In contrast, the change in boarding behaviour can be affected by other factors such as weather conditions. Therefore, the prediction in this study is to identify the boarding stops for each smart card user and the predictions are made for each of the operation hours of a day. We propose a three-stage framework to predict: (i) whether a smart card user is expected to travel or not in each one-hour time slot, (ii) which line they will use, and (iii) at which stop they will get on board. The predictions at each stage deploy three different architectures: a simple neural network (fully connected network) and two deep learning networks (recurrent neural network and long short-term memory network). Finally, the result of individual boarding stops will be aggregated to the hour ridership on the stop-, line- and network-level. Unlike calculating the ridership directly from the smart card data, using machine learning

techniques can make the prediction on the ridership. Public transport planning based on the future situation and changes in ridership will be more reliable and scientific.

3.2. Related works

The development in automatic data collection system offers an opportunity to understand travel demand and to plan the public transport system (Zhang et al., 2018). For example, GPS tracking of bus operation has been used to model the bus bunching (Wu et al., 2019); and smart card data (Bordagaray et al., 2016) and GPS data (Yang et al., 2019a; 2020) are used to capture the bike-sharing travel behaviour and to replicate the public transport system (Liu et al., 2019a). There is an extensive literature on observing and capturing the passenger flow from varying data resources. Yang et al. (2019b) use a smart card and social media data to explore the travel purpose and ridership of metro in Shenzhen. Oransirikul et al. (2014) present that the Wi-Fi activity correlates closely with the bus passenger flow at bus stops and on-board buses, and demonstrate the feasibility to measure the passenger flow by monitoring the Wi-Fi transmissions. Zhou et al. (2013) combine GPS data from private mobile phones and buses to measure the bus passenger flow. Sun et al. (2016) analyse and visualise the metro passenger flow in Shanghai directly from a closed automatic fare collection (AFC) system, which records both of the boarding and alighting stops. On the other hand, in many cities, the *entry-only* AFC system is used where only the boarding information is recorded, but the alighting information is unknown. To address this issue, Barry et al. (2002) proposed a trip-chaining method to identify the alighting stop for each smart card transaction. This method has been wildly applied in New York (Barry et al., 2009), Chicago (Zhao et al., 2007) and London (Gordon et al., 2013).

Meanwhile, there are extensive research interests in predicting future passenger flow. Yang et al. (2009) develop the regression equations to forecast the number of boarding and onboard passengers. The forecast is based on the number of smart card transactions. Autoregressive integrated moving average (ARIMA) model is introduced to model and predict the public transport passenger demand, as a time series data (Washington et al., 2010). Zhou et al. (2013) predict the total bus demand using the ARIMA model in conjunction with two other Poisson models. Gong et al. (2014) propose a sequential framework for a short-term prediction of the number of waiting passengers. In their framework, the seasonal ARIMA model is designed to predict the number of arrival passengers and empty space from the historical boarding, empty space and GPS data. With the help of real-time data from GPS and waiting passenger count, a Kalman Filter model follows the seasonal ARIMA model and predicts the number of waiting passengers. Ma et al. (2014) predict the short-term bus demand using time series and interactive multiple models (IMM). They construct the time series of weekly, daily and hourly demand and use IMM to combine the time series estimations and predict the final demand prediction. Following on from Ma et al. (2014), Xue et al. (2015) employ the ARIMA for predicting the 15-minute and weekly demand prediction, and the seasonal ARIMA for daily demand. Meanwhile, working towards the same objective, recent studies use machine learning methods, e.g. support vector machine (SVM) regression (Yang and Liu, 2016) and Shepard model (Jin et al., 2019) to predict bus passenger demand. One key common feature of these existing studies is that they rely on historical passenger flow on bus stops only, and they do not consider other external factors (e.g. weather conditions) in the prediction models.

Machine learning techniques, such as neural networks and Bayesian networks, have been used in predicting the mode choice (Zhou et al., 2019), arrival times (Yu et al., 2011) and train delays (Corman and Kecman, 2018), as well as public transport passenger

flow(Karnberger and Antoniou, 2020). Jiang et al. (2014) forecast the high-speed rail demand using ensemble empirical mode decomposition (EMD) and grey SVM models. Wei and Chen (2012) propose a method combining EMD and back-propagation neural networks for the short-term metro passenger flow prediction. Li et al. (2017) utilise the multiscale radial basis function networks to predict the metro passenger flow. The method can pinpoint the over-crowding stops under special events scenarios. Liu et al. (2019b) achieve the same objective through along short-term memory (LSTM) neural network. Since the AFC system in rail and metro records both alighting and boarding system, their historical passenger flow can be easily extracted from the smart card data. For the *entry-only* AFC system, Toqué et al. (2016) predict passengers origin-destination matrices at stop level over 15-minute windows using the LSTM networks, and they infer the alighting stops by trip-chaining model. Recently, Tang et al. (2020b) incorporate weather conditions and travel history of travellers in the gradient boosting decision tree model to estimate the alighting stop for every smart card trip and give the ranking of the importance of features.

As it is difficult to understand the travel behaviour of each bus passenger, there is an idea to cluster bus passengers and to analyse the group travel behaviour (Faroqi et al., 2018), which can significantly decrease the number of objects in the analysis. Faroqi et al. (2017) use histograms, Pearson correlation coefficients and hexagonal binning to analyse the smart card data in Brisbane. The results show a nonlinear spatial-temporal similarity correlation among bus passengers. Later, Faroqi et al. (2019) compare three methods based on the spatial-temporal characteristics of bus trips on clustering the bus passengers: S-T clusters the spatial matrix firstly and then temporal matrix for each spatial group; T-S clusters the temporal matrix firstly and then spatial matrix for each temporal group; and ST combines both spatial and temporal similarity matrices into one matrix. The study concludes that S-T method is better used in the cases where spatial similarity is

more important, and T-S method is the opposite. ST method is a moderate method that considers the effects of time and space equally. He et al. (2017) propose a method by combining cross-correlation distance and hierarchical clustering to segment the time series of passengers' travel pattern. He et al. (2019) incorporate a sampling method in their previous study to classify the temporal pattern of bus passengers. Based on the development of the clustering methods from the previous two studies, He et al. (2020) uses time-series distance metrics and a hierarchical clustering method to classify the public transport users. However, the boarding behaviour of passengers is similar but not identical. Grouping passengers generalises the typical characteristics of group behaviour and ignores the differences between individuals. On the contrary, a bottom-up model is able to capture the heterogeneity of individual travellers.

Summing up, the existing research focuses mainly on the total passenger demand over various time scales (e.g. only during peak hour or on a day). However, they do not identify the details, or the time-dependent load-profile, of the bus demand. According to the introduction from Ceder (2007), the load-profile and maximum load can help us better understand and plan the bus system and boarding behaviour is an essential part of the load-profile. Hence, this study contributes to predicting the boarding demand at different levels by starting with the prediction of individual boarding behaviour and takes account of the individual's travel history and the weather factors in the prediction.

3.3. Framework for boarding stop prediction

3.3.1. Problem statement

In the proposed model, we take the boarding behaviour of a passenger during a one-hour time slot as the instance of our model. The instance in a machine learning model contains a feature vector and a label vector. The feature vector consists of a set of

elements characterising the passengers in the analysed period. The label vector is a set of 0 or 1 for each stop, with 1 [0] indicating that the passenger has [not] boarded a bus at the concerned stop in the reference period. Each stop is a class in the machine learning model, which also includes 'NONE' to represent the non-travelling instance. The classification process aims to assign each instance to the possible class.

There are three data challenges in our case that affect the prediction accuracy when using machine learning techniques:

- Multi-label problem: passengers may make more than one trip within an hour, for example, starting a journey at stop 1 and transferring to a different bus line in stop 2 within one hour. In such cases, an instance (the boarding behaviour in one hour) may be assigned to more than one label (stop). Our study, therefore, belongs to the multi-label problem.
- Imbalanced data: there are about 98% instances where label vector is all 0, which means this passenger do not travel in this period and/or do not board at that particular stop. Boarding observations in a specific stop in a specific hour is hence a 'rare occurrence'.
- Many-class problem: a public transport system for a city has many bus stops, so there are many classes in the label. Such a many-class problem makes the classification more difficulty, which in turn reduces the accuracy of the model and the computational efficiency of the training and prediction process.

These problems are common for most of the urban public transport systems. We propose a three-stage framework to address these issues.

3.3.2. A three-stage framework for predicting boarding stops

In this section, we propose a framework with three sequential models to predict: (i) whether a smart-card user makes a trip in a given time slot (Stage 1), (ii) the bus lines the passengers used (Stage 2), and (iii) the boarding stop on the predicted bus line (Stage 3). Figure 3.1 illustrates the overall framework.

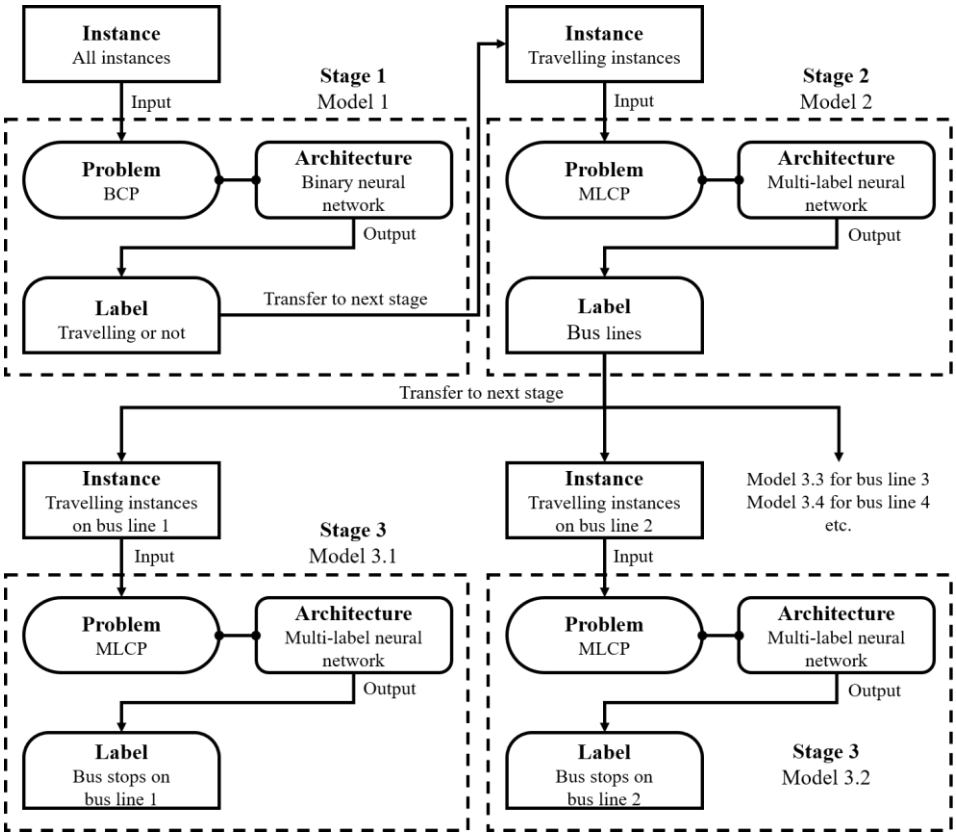

Figure 3.1 The processes of how the framework build up the models for each bus lines.

In Stage 1, we predict whether a passenger makes a trip at a given time (one-hour slot). This is modelled separately from the bus line and boarding stop selection to address the data imbalance issue highlighted in the Problem Statement earlier. As the label of this model is 'travelling' or 'not travelling', this estimation is referred to as a binary classification problem (BCP) in machine learning.

Stage 2 considers only the travelling instances predicted from Stage 1. It predicts the bus line for each travelling instance. More than a single class in the label might cause lower accuracy and longer computing time, and the boarding stops can be disjointly grouped by bus lines. Hence, the boarding stops prediction is divided by bus lines firstly, before considering the bus stops. Through Stage 2, we cut off the number of classes in the label from hundreds even thousands to the tens. Since passenger can make transfers in the same one-hour time slot, there may be more than one bus line for an instance. Therefore Stage 2 is a multi-label classification problem (MLCP).

Stage 3 works on our ultimate target, predicting the boarding stops for each bus trip. Here, we build up a prediction model (e.g. Model 3.1, 3.2, etc.) for each bus line. According to the difference in the number of stops along the bus lines, the number of classes in each model can vary from twenty to fifty. For the same reason in Stage 2, Stage 3 also belongs to the MLCP.

3.3.2. Architectures of neural network

We employ the fully connected neural network (FCN) as the basic architecture to solve the problems in the proposed framework because it has the simplest structure. Since human trajectory is highly related to the temporal regularity (González et al., 2008), another two architectures, recurrent neural network (RNN) and long short-term memory neural network (LSTM), are used to solve the problems in the proposed framework.

3.3.2.1. Fully connected neural networks (FCN)

FCN is a classic architecture of neural networks (Svozil et al., 1997). Figure 3.2 illustrates an example architecture of FCN. It consists of three layers: an input layer, an output layer and some hidden layers. Each layer has a different number of neural cells

(nodes). A node in the input layer represents an input feature. The nodes in the hidden layers are the results calculated by the activation function according to the information in the input layer. A node in the output layer presents the probability of the occurrence of a class. In FCN, the information moves from the input nodes, through the hidden nodes (if any) and to the output nodes. In our context, for Stage 1, there is only one node in the output layer of binary FCN architecture, which presents the probability of travelling. For Stage 2 and 3, the number of nodes in the output layer depends on the number of bus lines and bus stops on each line. The hidden layer is used to discover relationships between features through the activation functions.

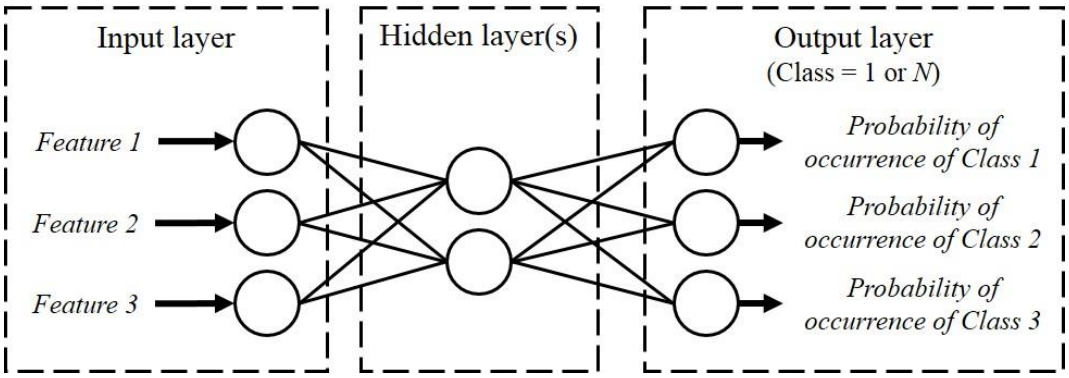

Figure 3.2 An example architecture of FCN.

3.3.2.2. Recurrent neural networks (RNN)

RNN architecture is good at dealing with sequential data (Connor et al., 1994). Like FCN, an RNN consists of an input, hidden and output layer, and each layer contains one or more nodes (as shown in Figure 3.3). Whilst the hidden layer in FCN is only calculated from one input instance, the hidden state in RNN is related to both the current instance and the hidden state of the previous instance.

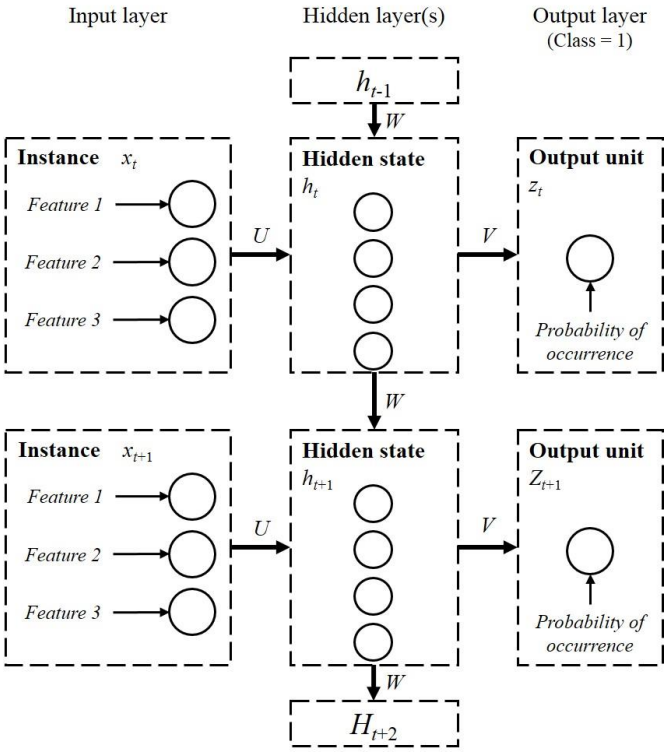

Figure 3.3 The example architectures of RNN.

The processes of RNN is formulated below. The input of RNN requires a sequence of instance:

$$\{x_1, x_2, \cdots, x_t, \cdots, x_T\} \tag{3.1}$$

where x_t denotes the input instance at the position t in the sequence and T is the number of the sequences. In this study, the sequence is the time series of instances, and Section 2.4.3 will introduce how we build up the sequence in this study.

The current hidden state of an instance x_t is measured from the instance itself and the previous hidden state:

$$h_t = \phi\left(Ux_t + Wh_{t-1}\right) \tag{3.2}$$

where h_t and $h_{t\text{-}1}$ denote the current and previous hidden state at the position t and t-1 in the sequence; U and W are the weight matrix from the input layer to the hidden layer and the hidden layer to the hidden layer; $\phi(\cdot)$ is the activation function which is usually a tanh function.

Then, the output unit can be calculated as:

$$z_t = o(Vh_t) \tag{3.3}$$

where V is the weight matrix from the hidden layer to the output layer and $o(\cdot)$ is the activation function which usually uses the sigmoid function for MLCP and BCP.

3.3.2.3. Long short-term memory neural network (LSTM)

LSTM, proposed by Hochreiter and Schmidhuber (1997), is an improvement on the RNN by solving the vanishing gradient problem Hochreiter et al. (2001). The architecture of LSTM is similar to RNN's (shown in Figure 3.4). However, it utilises a new concept, called cell state, which is the horizontal line going through the top of the figure, to memorise the state at the previous position. LSTM is a gate-controlled architecture, including forget gate, input gate and output gate. The forget gate controls how much old information can be inherited from the previous position.

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right]\right) \tag{3.4}$$

where f_t is the forget coefficient between 0 (totally forgetting) and 1 (totally remembering) at position t; W_f is the weight matrix for the forget gate; and $\sigma(\cdot)$ is the gate activation function which always uses the sigmoid function.

The input gate controls how much new information can be inherited from the current position.

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right]\right) \tag{3.5}$$

where i_t is the input coefficient between 0 (barely inputting) and 1 (totally inputting) at position t and W_i is the weight matrix for the input gate.

The output gate controls how much the cell state can be transferred to the next position.

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right]\right) \tag{3.6}$$

where o_t is the output coefficient between 0 (barely outputting) and 1 (totally outputting) at position t and W_o is the weight matrix for the output gate.

The final cell state at position t consists of the previous cell state at position t–1 and the new candidate cell state at position t:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{3.7}$$

$$\tilde{C}_t = \mu\left(W_C \cdot \left[h_{t-1}, x_t\right]\right) \tag{3.8}$$

where C_t and $\tilde{C}$ denote final and new candidate cell state at position t respectively; W_c is the weight matrix; $\mu(\cdot)$ is an activation function which always uses the tanh function.

The hidden state is computed by the output gate and the cell state.

$$h_t = o_t \cdot \varphi\left(C_t\right) \tag{3.9}$$

where $\mu(\cdot)$ is an activation function which always uses the tanh function.

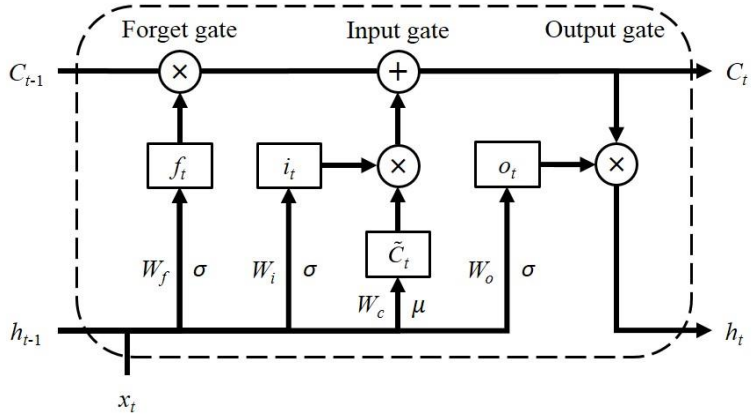

Figure 3.4 The example architectures of RNN.

3.3.3. Feature selection

As noted in the Introduction, travel history and weather conditions are the most significant factors in understanding travel behaviour. We, therefore, define model features into three domains: boarding time, weather condition and travel history. Features concerning boarding time (e.g. time slot, season, etc.) are clearly relevant to

the boarding stop prediction problem. The weather conditions (e.g. temperature, precipitation, etc.) implies the impacts of different weather variables on public transport users' behaviour. The travel history (e.g. boarding stops on the previous day, etc.) describes passengers' regular travel patterns and habits. We take binary encoding for card ID and One-Hot encoding for other categorical features; this leads to a high-dimension vector for representing the categorical features. All the numerical features are normalised. Table B.1 in Appendix B lists all the features employed in machine learning models.

3.4. Case study

3.4.1. Case description

The proposed framework is applied to a small bus network in the city of Changsha, China. Changsha is in the central south of China. Changsha has a subtropical monsoon climate. Temperature changes a lot in spring. The rainy season happens at the beginning of summer. Form the mid-summer to early autumn, the climate is very hot with little rain, and there are 85 days over 30℃ and 30 days over 35℃. In winter, the climate is not very cold, but ice sometimes freezes on the roads due to the freezing rain. The climate is distinctive in four seasons in Changsha. Thus, people's trip is always affected by weather conditions. As one of the major cities in south central China, 8.4 million people are living in Changsha, and the urban area covers 1938 square kilometres. Three bus companies operate over 200 bus lines serving for more than two million trips per day. The whole bus network covers all the roads in six administrative districts. The accessibility of bus network touches every corner of the city. Besides for three tourism bus lines, the headway of other bus lines is normally 10 minutes and sometimes reaches to 5 minutes during the peak hour. The bus is always the main mode of public transport service.

Due to the limitation of data accessibility, data from only seven bus lines are available to our case study (in Figure 3.5). However, these bus lines are still representative. First of all, the study network is composed of three lengthwise and four transverse lines and connects significant centre business districts, high-tech zones and residential zones around the city. Besides, the study network covers the core public transport infrastructures of Changsha such as rail stations and long-distance coach terminals. The three bridges, which are the main routes connecting the east and west city, are included in the network. Finally, the study bus lines cover a variety of service characteristics in terms of length, station density and frequency of the lines.

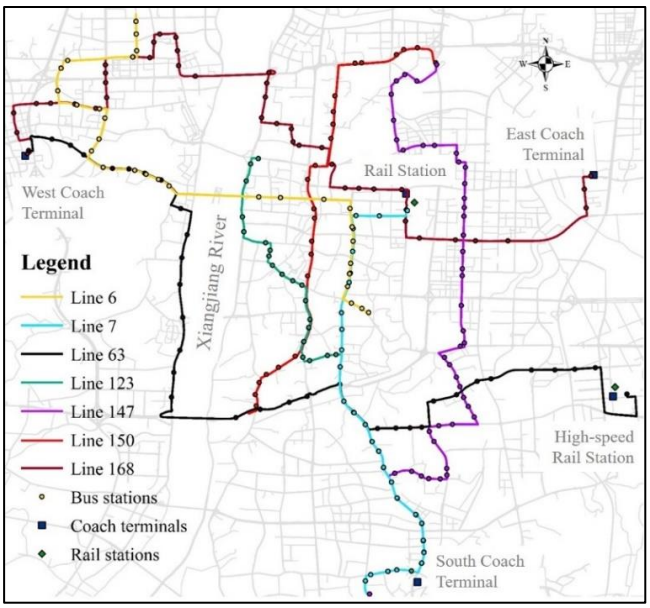

Figure 3.5 The map of the study case network in Changsha, China

The smart card system in Changsha records the boarding behaviour of each trip, i.e. bus lines, vehicle ID, card ID, card type and boarding time, but does not contain the specific boarding stop. For the supervised machine learning model, it requires the boarding stops in training the model and evaluating the performance of the prediction. We hence utilise the GPS data to implement the geographic information and identify the boarding stops. The GPS devices are equipped for all the buses. Each device has its unique ID,

which corresponds to the vehicle ID in the smart card data. The GPS reports the latitude and longitude of the vehicle location every 10 seconds.

3.4.2. Data pre-processing

The available smart card data covers 32 days from 1st August to 1st September 2016. The bus services operate 19 hours a day from 6 am to 1 am on the following day. The raw smart card dataset includes 2,917,272 transactions and 564,803 card users for the selected lines. Processes are taken to clean the raw smart card data and to prepare the training and testing datasets for machine learning models. Figure 3.6 illustrates the data processing procedures and the resulting number of data records after each procedure. To simplify the problem, the following assumptions are made.

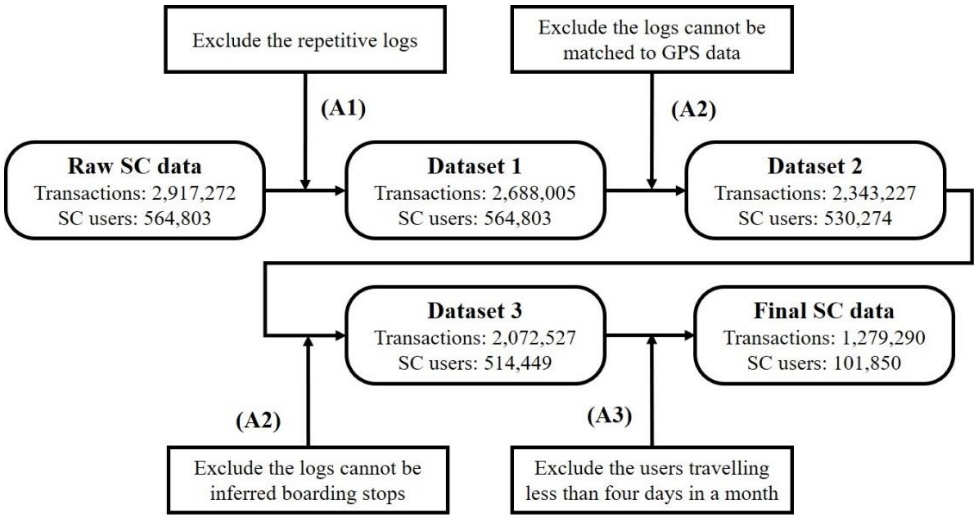

Figure 3.6 Processes to prepare smart card data.

(**A1**) *Each card ID corresponds to a single passenger, and each passenger swipes the card only once at a single boarding.* In a real-life situation, some passengers may (accidentally) swipe their cards more than once for one boarding, which causes two and more transactions during a short period. We consider these data as repetitive data. The first process cleans out 229,267 repetitive logs (8% of the raw data).

(**A2**) *Smart card logs for which the boarding stop cannot be inferred are considered noise.* Due to poor data quality, the GPS data for some vehicles is missing. In total, 344,778 smart card transactions (12% of the raw data) do not have the corresponding GPS data. Then, we extract the boarding stop of each smart card transaction with the data fusion of GPS data. 270,699 smart card transactions (9% of the raw data) are removed from because they cannot be inferred the boarding stops.

(**A3**) *This study only focuses on the regular smart card users who travel at least once a week.* Many card IDs appear only a few times during the 32 days of the study period. For example, 36% of users travelled only in August and not appeared on 1st September. 37% of users travelled less than four times. To simplify the data, we exclude the infrequent users travelling less than four because this kind of users will generate too many non-travelling instances for the model and we have intended to reduce the number of non-travelling instances and to lighten the burden of calculation. Additionally, since there are few travelling instances for the infrequent users, they can provide limited information to the models. This assumption results in a remaining 101,850 IDs for the rest of the study.

Finally, we transform the smart card records to the instances used in our models. There are nineteen time slots (corresponding to the nineteen hours of service operation) in a day so that every user ID has nineteen instances for each day. If there are two or more smart card records for the same person at the same time, including time slot and day, the corresponding instance will have two or more labels which become an MLCP. If there is no smart card record in a time slot, we label such instances as 'NONE', i.e. not travelling.

3.4.3. Experimental environment and setting

The training and testing process is conducted via Keras (Chollet and Others, 2015) with the R programming language. All the experiments are run on a graphics processing unit (GPU) platform with eight NVIDIA® K80 (GK210) and 12 GB GPU memory per unit.

Since we do not know the travel history for the instances for the first seven days (from 1st to 7th August 2016), the data concerning these seven days are excluded in the training dataset. The instances from 8th to 30th August are included in the training dataset; the instances on 31st August are in the verification dataset; while the testing dataset contains the instances on 1st September. Table 3.1 presents the number of instances in the different datasets of models.

Table 3.1 The number of instances in the different datasets of models.

Models		**Bus lines**	**Number of instances**		
			Training dataset	Validation dataset	Testing dataset
Stage 1		All network	44,508,450	1,935,150	1,935,150
Stage 2		All network	593,608	48,885	
Stage 3	Model 3.1	LINE 006	280,907	10,220	
	Model 3.2	LINE 007	290,628	10,544	
	Model 3.3	LINE 063	81,906	3,257	
	Model 3.4	LINE 123	147,917	5,235	
	Model 3.5	LINE 147	112,611	4,164	
	Model 3.6	LINE 150	296,020	11,159	
	Model 3.7	LINE 168	121,771	4,828	

In Section 3.3.2, we introduced three popular architectures of neural network (FCN, RNN and LSTM) to solve the binary and multi-label classification problems. The specific architectures used in this study are presented in Table 3.2. The number of nodes

in the input and output layers are the same for each of the three architectures. The number of hidden layers and nodes in each hidden layer vary from model to model. The activation function from the input layer to the hidden layer and between two conterminous hidden layers is the rectified linear unit (ReLU) function for FCN and the tanh function for RNN. The activation functions used in LSTM are tanh and sigmoid function. The activation function from the hidden layer to the output layer is the sigmoid function for all the three architectures.

Table 3.2 The structure of the specific machine learning models.

Models		**Architectures**	**The number of nodes**					**Activation function**
			In-put	Hidden			Out-put	
Stage 1	Model 1	FCN	133	95	66	–	1	ReLU sigmoid
		RNN		133	128	64		tanh sigmoid
		LSTM		64	32	32		
Stage 2	Model 2	FCN	133	100	75	–	7	ReLU sigmoid
		RNN		133	128	128		tanh sigmoid
		LSTM		64	32	32		
Stage 3	Model 3.1	FCN	169	128	128	64	33	ReLU sigmoid
		RNN		128	64	64		tanh sigmoid
		LSTM		128	128	64		
	Model 3.2	FCN	169	128	128	64	33	ReLU sigmoid
		RNN		128	64	64		tanh sigmoid
		LSTM		128	128	64		
	Model 3.3	FCN	169	128	128	64	44	ReLU sigmoid
		RNN		128	64	64		tanh sigmoid
		LSTM		128	64	64		

Models		Architectures	The number of nodes					Activation function
			In-put	Hidden			Out-put	
Stage 3	Model 3.4	FCN	169	128	128	64	46	ReLU sigmoid
		RNN		128	64	64		tanh sigmoid
		LSTM		128	128	64		
	Model 3.5	FCN	169	128	128	64	63	ReLU sigmoid
		RNN		128	64	64		tanh sigmoid
		LSTM		128	128	64		
	Model 3.6	FCN	169	128	128	64	39	ReLU sigmoid
		RNN		128	64	64		tanh sigmoid
		LSTM		128	128	64		
	Model 3.7	FCN	169	128	128	64	48	ReLU sigmoid
		RNN		128	64	64		tanh sigmoid
		LSTM		128	128	64		

As mentioned in Section 3.3.2, the input of FCN is the individual instance, but the input of RNN and LSTM is a sequence of instances which consists of a time series. Following Han et al. (2019), the input sequence in our study consists of the instances from the previous seven days, which is donated by the set of instances, X. However, the instances extracted in the previous days are from different time slots. For example, the time slot selected for the sequence is illustrated in Figure 3.7. The target time slot, x_t, is 9:00 - 10:00 on 1st September. The sequence firstly contains the instances in all time slots of the previous day (expressed as $x_{t-m-s+1}$ to x_{t-s}), i.e. from 6:00 on 31st August to 1:00 on 1st September. In the previous second to sixth days, we only select the same time slot (expressed as x_{t-dm}), i.e. 9:00 - 10:00 on 26th to 30th August. Finally, we consider that the same day of the previous week may have the similar behaviour, so all

time slots ($x_{t-7m-s+1}$ to x_{t-6m-s}) in that day are included, i.e. from 6:00 on 25th to 1:00 on 26th August. Therefore, the input sequence is formulated below.

$$X^p = \left\{ x^p_{t-7m-s+1}, x^p_{t-7m-s+2}, \cdots, x^p_{t-6m-s}, x^p_{t-dm}, x^p_{t-m-s+1}, x^p_{t-m-s+2}, \cdots, x^p_{t-s}, x^p_t \right\} \quad (d = 2,3,\cdots,6) \tag{3.10}$$

where p represents a smart card user; t is the target time slot to be predicted; s is the position of the target number in that day; m is the total number of time slot in a day which equals to 19 in this study; d indicates the day.

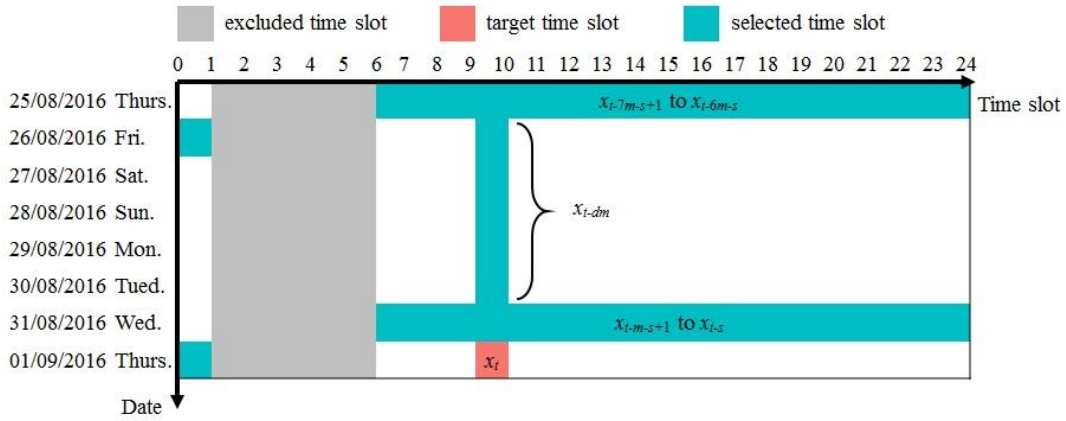

Figure 3.7 Example to illustrate the time slot selected for the sequence.

3.5. Model results and discussion

3.5.1. Performance measurements

The direct result from machine learning models is the boarding stop for every instance. We adopt the confusion matrix (presented in Table 3.3) including precision, recall and F1 score to evaluate the performance of the models (Godbole and Sarawagi, 2004). The precision measures the fraction of correctly predicted instances among the truly positive instances, which reflects the ability to identify only the relevant instances; the recall measures the fraction of correctly predicted instances among the instances predicted positive, which expresses the ability to find all relevant instances; F1 score is the

harmonic mean of the value of precision and recall, which balances the precision and recall of the model. Their formulations are presented below.

Table 3.3 The confusion matrix for the estimated results.

		Observation	
		Positive	Negative
Prediction	Positive	true positive (TP)	false positive (FP)
	Negative	false negative (FN)	true negative (TN)

$$precision = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_k}{TP_k + FP_k} \tag{11}$$

$$recall = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_k}{TP_k + FN_k} \tag{12}$$

$$F1 = 2\frac{precision \cdot recall}{precision + recall} \tag{13}$$

where TP is the number of true-positive instances which are correctly predicted as positive; FP is the number of false-positive instances which are incorrectly predicted as negative; FN is the number of false-negative instances which are incorrectly predicted as positive. K denotes the total number of classes in the models, and k is the index of classes.

Besides generalising the above three measurements, we introduce the Hamming Loss (HL) score to the evaluation (Schapire and Singer, 2000) which measures the fraction of the wrong labels to the total number of labels. Lower HL score indicates the higher performance of models.

$$HL(y_m, \hat{y}_m) = \frac{1}{|M|}\sum_{m=1}^{|M|}\frac{xor(y_m, \hat{y}_m)}{|K|} \tag{14}$$

where M is the total number of instances to be predicted where m is the index of instances; K denotes the total number of classes in the models; y_m and $\hat{y}_m$ respectively denote the ground truth and predicting results of instance m; $xor(\cdot)$ stands the XOR operation in Boolean logic.

3.5.2. Model performance – disaggregated results

Table 3.4 lists the running time (in seconds) of models. RNN is the fastest model. Since LSTM optimises RNN by adding the gate structure, it always takes longer to run. However, while FCN has the simplest architecture, its running time is consistently longer than RNN, and even longer than LSTM in some models, suggesting low computing efficiency of FCN compared to RNN and LSTM.

Table 3.4 Running time (in seconds) of the machine learning models.

Models		FCN	RNN	LSTM
Stage 1	Model 1	37,944	13,185	38,927
Stage 2	Model 2	6,900	9,315	6,240
Stage 3	Model 3.1	4,788	4,488	8,086
	Model 3.2	2,961	2,052	2,376
	Model 3.3	2,405	1,054	2,002
	Model 3.4	2,548	2,592	2,296
	Model 3.5	1,498	864	1,152
	Model 3.6	6,757	4,851	4,068
	Model 3.7	451	868	3,510

Table 3.5 lists the number of instances in four different categories of the confusion matrix for the machine learning models in Stage 1 and 2. We can see that many negative

instances are inaccurately predicted to be positive and positive to be negative. This inaccurate prediction (*FP* and *FN*) decreases the score of precision and recall. The difference in *FN* and *TN* between machine learning and random classification shows that machine learning approaches significantly improve the ability in distinguishing negative observations than the random classification. Also, machine learning approaches slightly improve the ability to find out the positive observations.

Table 3.5 The number of *TP*, *FP*, *FN* and *TN* instances of the confusion matrix in Stage 1 and 2.

Stages	Architectures	*TP*	*FP*	*FN*	*TN*
Stage 1	FCN	28,482	20,039	16,674	1,869,955
	RNN	27,948	20,573	17,208	1,869,421
	LSTM	33,625	14,896	11,531	1,875,098
	Random classification	22,578	944,997	22,578	944,997
Stage 2	FCN	32,332	15,076	13,335	255,349
	RNN	29,181	17,809	16,486	252,616
	LSTM	30,597	17,893	15,070	252,532
	Random classification	22,834	135,212	22,833	135,213

Figure 3.8 shows the performance measures of machine learning models. As reviewed by Tsoumakas and Katakis (2007), a good performance means the value of precision, recall and F1 score is greater than 0.5, and the HL score is less than 0.25.

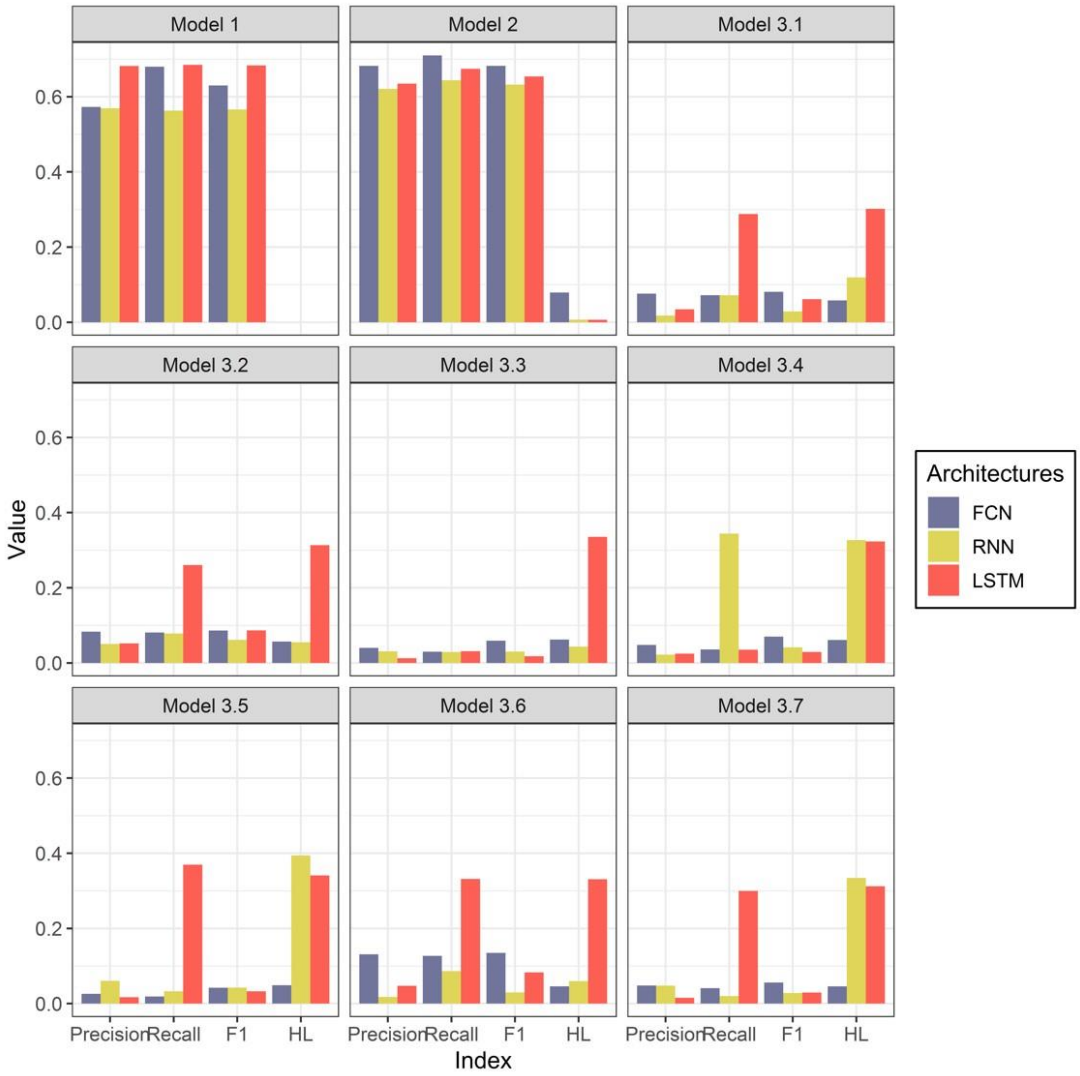

Figure 3.8 The performance measurements on machine learning models: Precision, Recall, F1 score and Hamming Loss.

Looking at all the architectures, precision, recall and F1scores of the models in Stage 1 and 2 are all above 0.5, and the HL score of Stage 2 is only 0.002. The models in Stage 1 and 2 perform well. Whereas the models of Stage 3 show some limitations. Although the HL score in Model 3.1 to 3.7 shows the high ability in the prediction, their precision, recall and F1 score are at a low level. Comparing Figure 3.8 with Table 3.1, one can observe that the performance of the models in Stage 3 is related to the size of the training dataset, with bigger datasets corresponding to relatively better performances. The number of classes, precisely the stops of the bus services in Stage 3, may have influenced the performance of the models. Larger numbers of stops increase the difficulty of the

prediction, a typical issue of over-many classes. Therefore, in Stage 2 and 3, even though we reduce the number of classes in the label to the tens, the prediction cannot reach high performance. Also, the predictor works well when there are a few classes in the label, e.g. Stage 2. Another possible reason is that our prediction models are consecutive. The error from an earlier stage will be transmitted to the next stages. For example, if a travelling instance is wrongly predicted as a non-travelling instance in Stage 1, the result will be wrong in Stage 2 and 3, no matter which stops it is predicted to get on.

Comparing the performance measurements among the architectures, LSTM is good at recall, while FCN is the best architecture fort precision, F1 and HL scores. In models of Stage 1, LSTM is the best architecture in all aspects. The precision of FCN and RNN is similar; however, the recall of FCN is greater than RNN. In Stage 2, the bar charts show FCN is the best and RNN is the worst, but the difference between these three architectures are small. Additionally, the HL score of FCN is much higher than others even though the value of HL score of FCN is still less than 0.25. In Stage 3, LSTM always has a significantly high value of recall with the worst value of the HL score, which is higher than 0.25 for all the models. In Model 3.4, 3.5 and 3.7, RNN also has a bad HL score.

We examine the influence of poor prediction of the previous stage on the next stage. The TP and FP instances of Stage 1 are used in the original testing dataset of Stage 2. However, the FP instances are always negative, no matter which bus line is predicted in Stage 2. These FP instances decreases the performance. Here, we only use TP instances of Stage 1 as the testing dataset of Stage 2 and measure the precision of new Stage 2. So do the models in Stage 3. Figure 3.9 presents precisions calculated in these scenarios. The results show that the precision with new testing datasets in Stage 2 is about 0.8 and in Stage 3 are around 0.5, which is significantly greater than what we

calculate before. This situation proves that the poor performance of Stage 1 decreases the precision of Stage 2 and 3. The error will be transformed and accumulated stage by stage. If we can improve the performance of Stage 1 or if we have already known the travelling instances, the rest of stages (Stage 2 and 3) are able to find out which bus lines and stops passengers use.

Figure 3.9 The precision of the model with original testing dataset (with TP+FP) and new testing dataset (TP only).

3.5.3. Accuracy of ridership – aggregated results

For public transport planning, aggregated behaviour is more important than the individual ones, as the main input of interest to the planners and operators are the predicted ridership in each line during each time slot. Here, we measure the predicted aggregated demand from different architectures and compare them with the true ridership and the predicted ridership by the ARIMA model (Hillmer and Tiao, 1982).

First of all, the total real ridership at the network level is 45,156. The predicted ridership (from Stage 1) is respectively 48,521 from FCN, 49,609 from RNN, 45,667 from LSTM and 49093 from ARIMA. The absolute errors are less than 4,500 and the percentage errors are within 10%. Thus, results from four architectures are all considered close enough to the true ridership, with the result from LSTM falling within 1% of the true value.

To test the sensitivity of the predictive on feature, we make a demonstration by changing the temperature. The range of temperature in testing data is from 21 to 29 degree. It is a nice day in summer. We manually increase and decrease the temperature by 10 degree respectively. The network ridership is presented in Figure 3.10. The change in temperature has caused a change in ridership. Higher temperature reduces bus trips, while cooler weather in summer motivates bus trips. However, these changes are subtle. The increment of ridership is only within 2%, and the reduction of ridership is within 4%. As can be seen from Chapter 2, temperature is not a very important feature. Moreover, due to the large number of features in this study, the sensitivity of the model to a single variable is not strong.

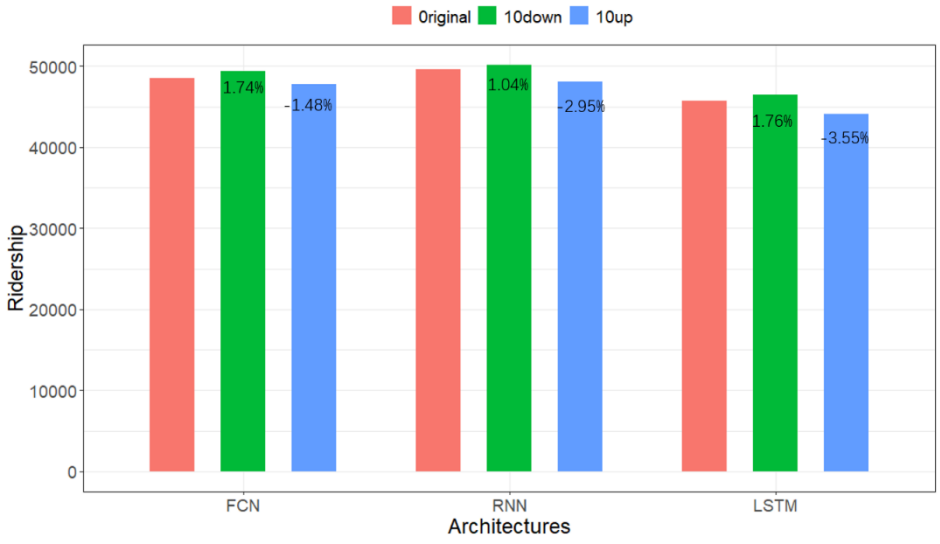

Figure 3.10 The sensitivity of predictions of model on temperature.

The ridership distribution over the day is presented in Figure 3.11. The true ridership has two clear peaks around 8:00 and 18:00 respectively. LSTM, RNN and ARIMA have all accurately predicted when the peaks occurred. FCN also accurately predicted evening peak at 18:00, however, it predicted morning peak is later than the true one by two hours. Looking at the value of the ridership, LSTM matches the true ridership best where two distributions almost overlay on each other perfectly. The ridership predicted by RNN and ARIMA is similar, and both are higher than the truth before the evening peak. The ridership predicted by FCN is much lower than the truth during the morning peak and significantly higher during the evening peak. During the off-peak time, i.e. from 11:00 to 17:00, the ridership from FCN is close to the actual ridership and not worse than other architectures. Therefore, as LSTM, RNN and ARIMA consider the time series in their model, they have the ability to capture the temporal characteristics of the data. In contrast, FCN takes a poor performance in the time-dimension because it only uses independent features.

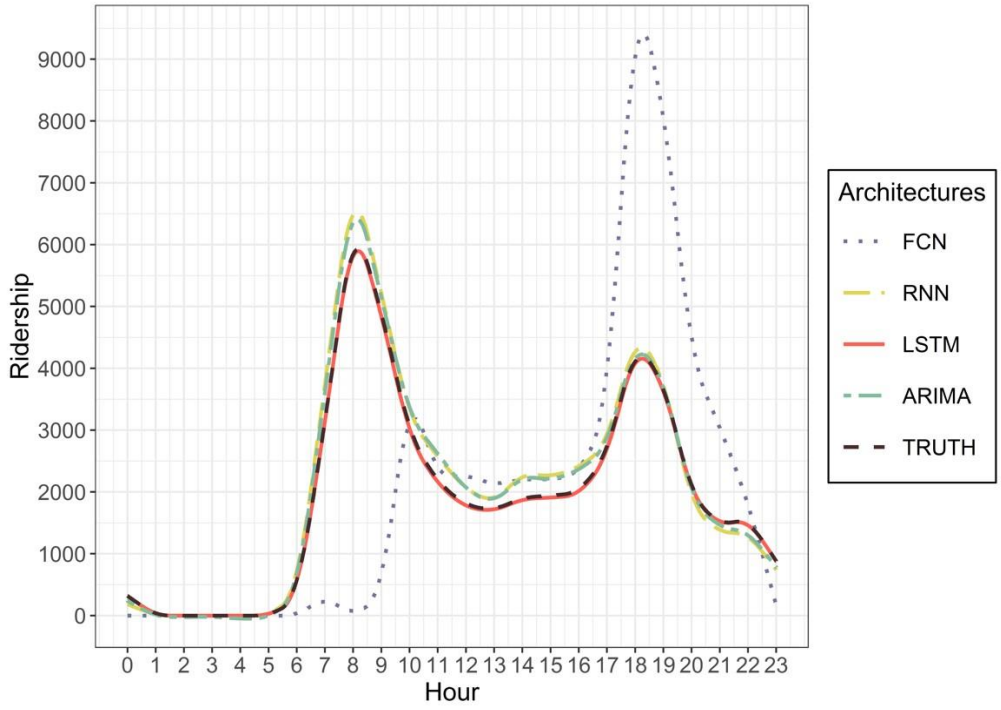

Figure 3.11 Ridership at the network level in truth and predicted by different architectures and models in Stage 1.

Figure 3.12 presents the true and predicted ridership for bus lines from the results of Stage 2. The overall picture is similar to those in Figure 3.12 that: for each bus line, RNN, ARIMA and LSTM all accurately predicted the timing and the level of the morning and evening peaks, whereas FCN did not produce so accurate predictions. Errors concerning Line 006, 007 and 150 that have large numbers of instances are greater than errors for other lines with fewer instances.

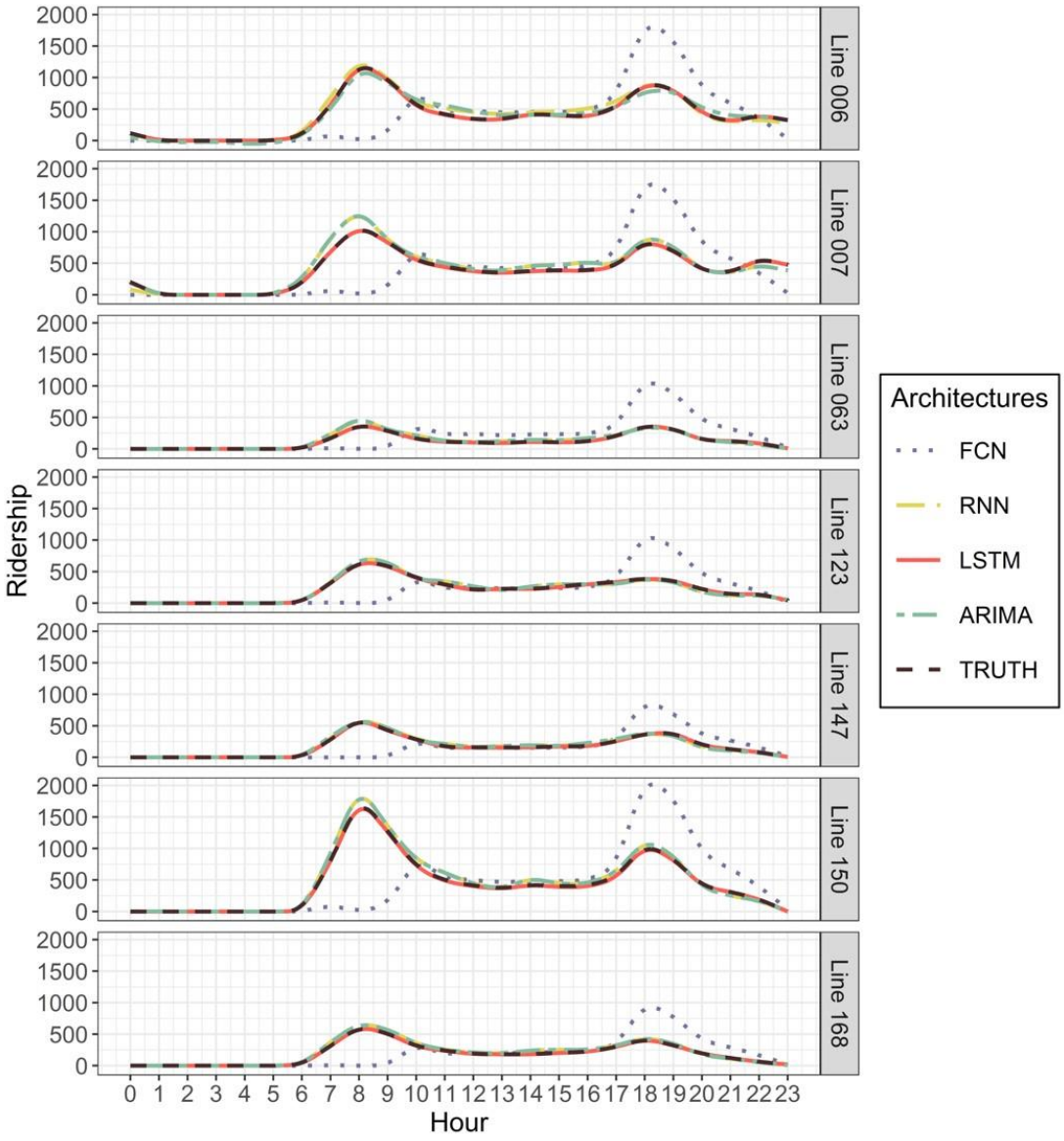

Figure 3.12 Ridership delivered to bus lines in truth and prediction by Stage 2.

To investigate the demand distribution in detail, we compute the ridership for each bus stop from the results of the models in Stage 3. Figure 3.13 shows the true and predicted

ridership at stop-level from different machine learning architectures. The interpretation of the results in Figure 3.13 is complex. For all the seven bus lines, FCN is able to capture busy stops, which have more boarding passengers, even for the bus lines whose machine learning model does not perform well. The results concerning Line 006, 007, 150 and 168 fits the observed ridership almost perfectly in terms of the position of busy stops.

However, there is still an error between the real and predicted ridership from FCN. As for the other three lines, FCN captures the busy stops in reality but incorrectly predicts more other busy stops. For example, on Line 123, observed busy stops are the stop No. 27, 43 and 46; besides, the model predicts the other six busy stops. Due to the poor performance of the models in Stage 3 (see Model 3.1 to 3.7 in Figure 3.8), we speculate that some instances are incorrectly labelled to other bus stops in addition to the actual stops. Hence, there are more predicted busy stops, and the total demand of bus lines is higher than the reality. For RNN and LSTM, their prediction differs from real observation. The indication from Figure 3.13 is opposite to that from Figure 3.11 and 12. On most of bus lines, RNN points out some busy stops but they are not correct. For example, RNN suggests that the busy stops on Line 147 are No. 7, 26, 43 and 56 but observed busy stops are No. 38 and 42. Looking at the results in LSTM, the distributions of the ridership along bus lines are predicted to a balanced result, where LSTM results are basically horizontal lines. It is hard to distinguish the busy and free stops according to the results from LSTM. Although the number of non-travelling instances is already reduced in Stage 1, the instances of travelling at each stop in the models of Stage 3 are still imbalanced. The three architectures we tested have different ability to deal with the issue of imbalanced data: FCN can replicate a part of the peak of data's real distribution; the predictions of RNN are not reliable for many busy stops; the number of instances in each class is balanced from the results of LSTM.

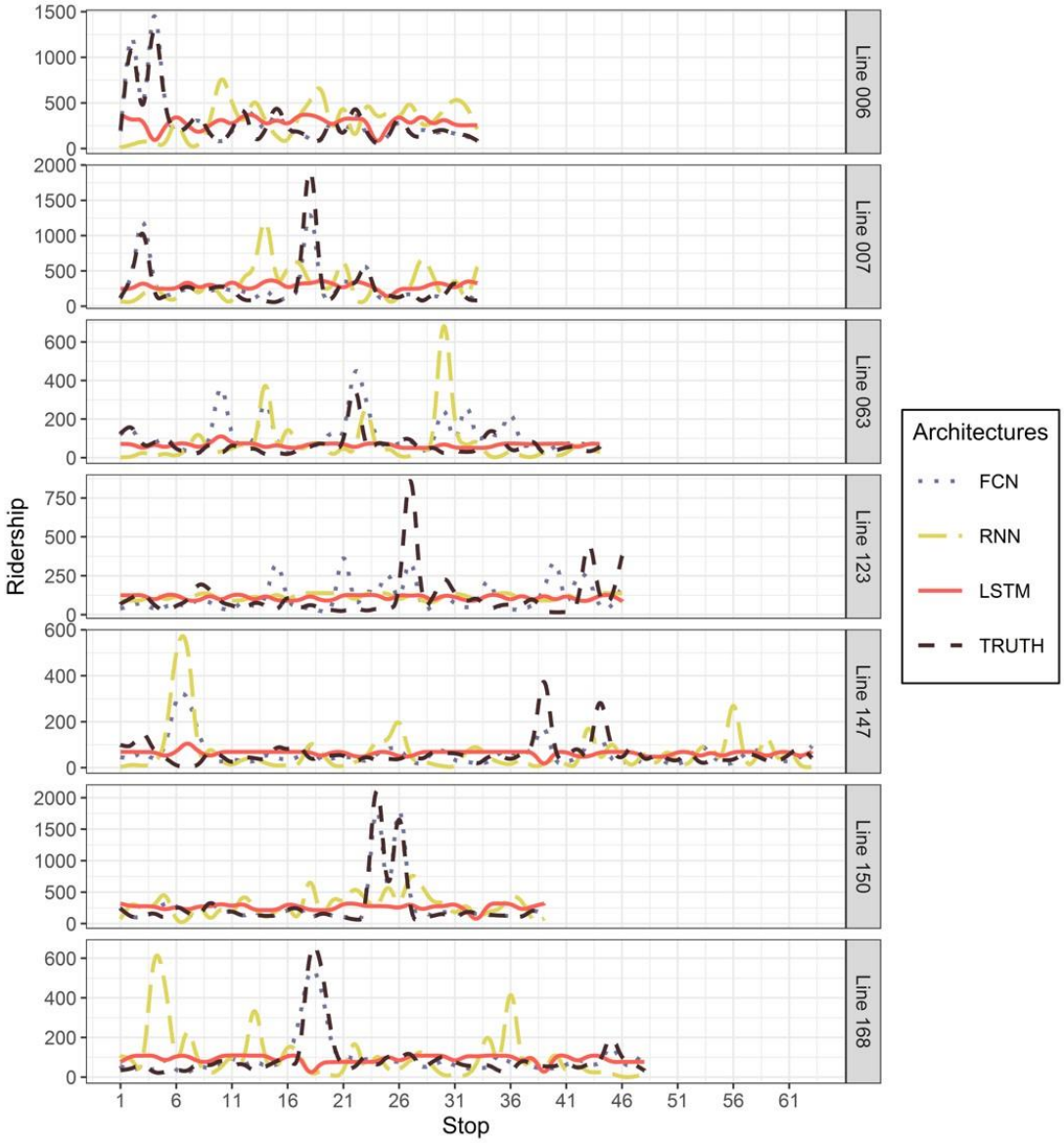

Figure 3.13 True and predicted ridership at stop-level from different architectures.

We show the temporal distribution of the demand at the stops with the largest ridership in Figure 3.14. For operational and planning reasons, it is important not just to identify the busiest stops but also to understand the peak demand. Furthermore, we want to check whether errors are less common when we have more instances. For lines 007, 063 and 123, the predictions of LSTM and RNN have the same temporal trend, although the predicted demand is significantly low. The approach fails to reproduce the temporal distribution of the demand on Line 006, 147 and 168 the number of instances

assigned to peak stops is smaller than the real one due to the error discussed in the previous paragraph. The demand pattern on Line 150 predicted by LSTM and RNN differs from the observed one. As FCN assigns more instances to stops, it is much clearer to analyse their temporal distribution. FCN captures the pattern of the distribution of ridership of most bus lines, i.e. Line 006, 007, 123, 147 and 168. However, FCN fails to replicate the morning peak pattern; for example, there is a significant gap from 6 to 9 am. For Line 150, the result of FCN is similar to LSTM and RNN, with no peak hours at all.

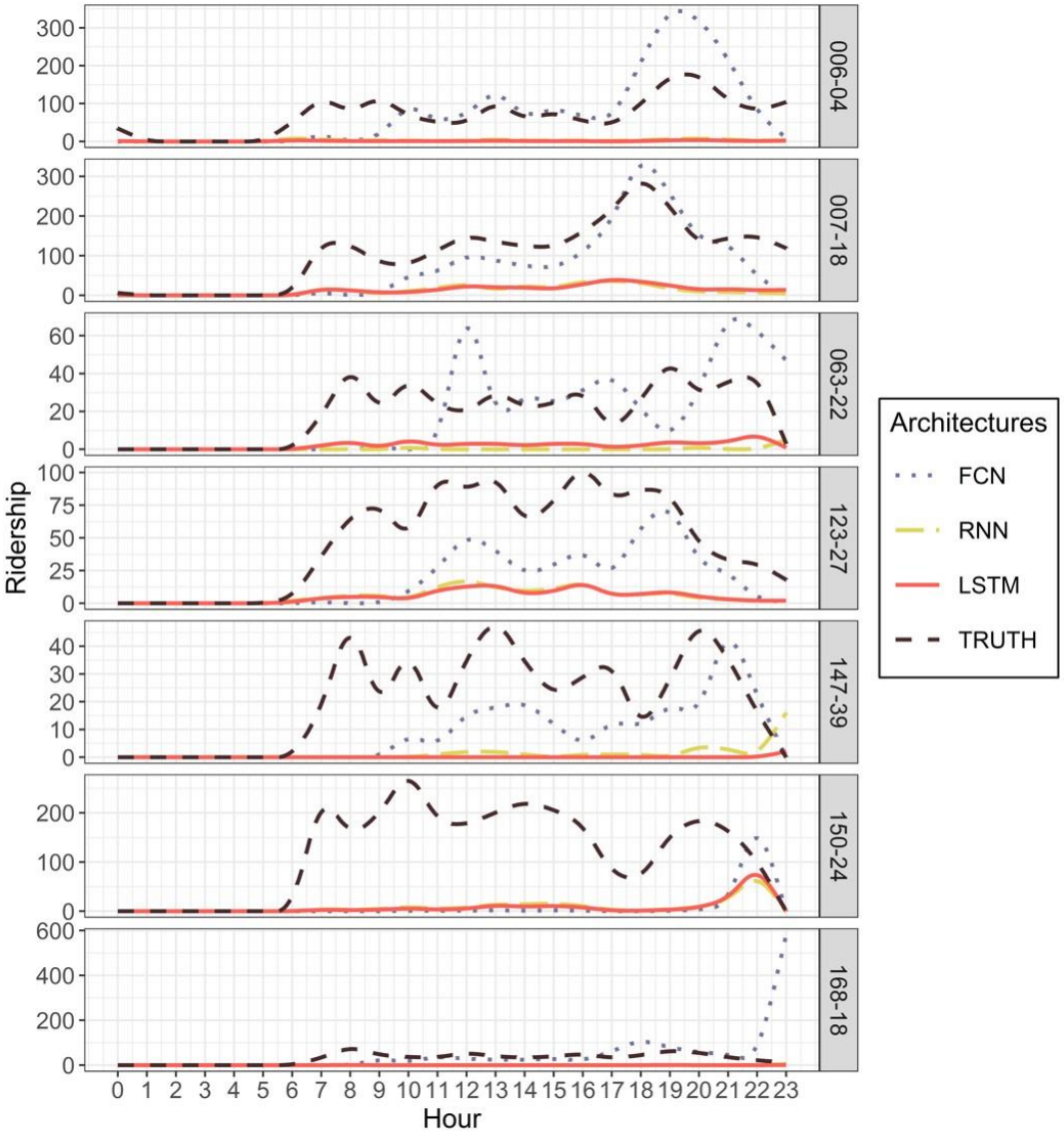

Figure 3.14 True and predicted ridership at the largest–ridership stop from different architectures.

Besides, we are aware that Figure 3.14 presents a different result from Figure 3.13. In Figure 3.13, FCN is the best model to predict the busy stops while FCN is the worst model in Figure 3.14. It is because Figure 3.13 presents a whole-day ridership in different bus stops, and Figure 3.14 present the hourly ridership at one bus stop. This difference proves again that FCN is able to capture the spatial distribution (along with bus stops) of ridership but lacks ability on temporal characteristics (over time).

Overall speaking, the aggregated results predicted from LSTM and RNN match well with the true ridership at the network- and line-levels and the results are better than those predicted from the classic ARIMA model. However, all models lack the ability to make the prediction at the stop-level. The accurately capturing of the temporal distribution of ridership is perhaps due to the consideration of the temporal relationships between the data. FCN can predict the distribution of ridership at stops better than the other architectures, but the results are not satisfactory in absolute terms. We believe that two key causes contribute to reducing the accuracy of the predictions:

- A large number of classes (stops) in the system makes it difficult to predict at the stop level, even though we adopt the multi-stage framework to reduce the number of classes in each model.
- LSTM and RNN do not work well with imbalanced data, which affects the quality of the stop-level predictions.

3.6. Summary and Conclusion

Understanding the travel pattern is important for improving the level-of-service of public transport systems and capturing passengers' choice of boarding stops is the first step to predict the travel pattern. Predicting boarding behaviour tells planner the

situation and changes of ridership in the public transport network. An accurate prediction of ridership is the data basis of long-term scientific planning and short-term operation. It, therefore, improves the level-of-service of public transport systems. Thus, working on predicting the boarding behaviour and making the prediction more accurate will benefit in improving the attraction and patronage of public transport systems, which contributes to the sustainability of cities.

This paper presents a framework to predict the boarding stops for each smart card user in every one-hour time slot. The proposed framework consists of three sequential stages. First, we predict the states, travelling or not, for each instance because 98% of instances are labelled to 'non-travelling' which causes the issue of imbalanced data. At the second stage, we only look at the travelling instances and predict the bus lines they travel on. This is to reduce the number of classes in machine learning models. Finally, we predict the boarding stops on every bus line. FCN, RNN and LSTM are separately used as the architectures in the framework, and weather conditions and travel histories are incorporated in the features of models.

The direct output of the machine learning framework is the boarding stops of individual passengers at different hours. Given that the aggregated ridership is more important in public transport planning, we calculate the hourly ridership at stop-, line- and network-level. Different from the direct predictions of ridership at the stop level (which has been the focus of most existing literature), this paper deals with the prediction at the individual smart card user level. The instance of our models is the boarding behaviour of a passenger in a one-hour time slot. The result of the method is a specific boarding stop for each bus trip. Using results concerning individual users, it is easy to obtain the aggregated ridership at stop-, line- and network-level. Moreover, when prediction the ridership at stop-level, the classic model, such as ARIMA, needs to build up a single model for each bus stop. Starting at the individual level, the results from our method

can produce the aggregated ridership at stop-, line- and network-level. Thus, it reduces the number of models required. LSTM and RNN are shown to produce the time-dependent distribution of the line-level ridership accurately.

Looking at the machine learning models, the accuracy of the predictions of the machine learning models is not high as one would like. The reasons are i) for Stage 1, the valid travelling instances are only 2% of the whole training dataset which is an extremely imbalanced data for machine learning models; and ii) for Stage 2 and 3, the error is transferred from Stage 1, i.e. the non-travelling instances that were wrongly predicted to travelling instances by Stage 1 are always wrong in Stage 2 and 3, no matter what bus stops or lines are predicted. Nevertheless, in some cases, our machine learning approach is able to predict correctly 2/3 instances. Comparing the different architectures, FCN is better on precision, F1 score and HL score, while LSTM is better on recall. Conclusively, FCN is better than other two architectures in the comprehensive ability (F1 score) and the ability to deal with MLCP (HL score), and LSTM has a more powerful ability to find all the possible instances in each class. On the ridership side, all of the architectures accurately predict the total number of travelling instances in the network and on each bus line but lack the ability to make the prediction at the stop-level. To see the temporal distribution of the ridership, LSTM and RNN can predict the accurate ridership in each hour, which is more accurate than the classic time series ARIMA model. However, FCN has a poor performance to predict peak hours. It is because RNN and LSTM consider the temporal relationship in their learning process, while FCN only uses the features independently. On the side of the distribution along with bus stops, FCN is able to capture the pattern of boarding at the stop-level and find out busy stops, but LSTM and RNN do not perform well. Examining in details of ridership at busy bus stops, FCN has a good prediction on the absolute value and trend of ridership, especially after the morning peak of a day. Although LSTM and RNN

have a poor prediction on the absolute value of ridership, they can still reflect the changes of the ridership. The reason, we think, causing the poor prediction is because the training and testing datasets for the models in Stage 3 are still full of imbalanced data and these three architectures have different ability to deal with the imbalanced data.

The predictive models are trained by the data in Changsha, so the trained model is only responsible for the data we used in the study. However, the prediction framework can be easily transferred to other bus systems trained. Operators can collect smart card data and train the model with local smart card datasets, which are generally available to operators. Also, operators can customise the predictive models by adding or removing features according to the situation in different cities. A trained model for a new bus system is easily obtained. We note that the data issues described in Section 3.1 are common to other networks. We expect our method can also deal with these data issue in other networks. Clearly, the quality of results in other bus networks should be tested.

This study incorporates weather conditions and travel history in the prediction models. However, we have not examined how these features impact on prediction models. In the future, we will address the efforts of these features in different domains and attempt to rank the importance of these features. Secondly, we only use one-month data as the training dataset and one-day data as the testing dataset. This short period in this study limits the variability of weather conditions and subsequently limits the prediction of weather impacts on boarding behaviour. For example, the boarding pattern in winter has not been learned because the data do not contain such information. It needs to be brought into the model when more data is available. Due to the limitation of computer performance, the size of the data far exceeded computation capacity. We have to delete a part of the infrequent passenger in the case study. Moreover, the infrequent passengers can provide limited information about their boarding patterns. However, ignoring infrequent passengers leads to a lack of their boarding pattern in the models and also

results in a bias of model. It is worth to investigate how to predict the boarding behaviour of infrequent passengers and what the long-term impact of weather is. Thirdly, the number of bus stops and lines can be very big in a large network. Even if the framework proposes three successive stages to avoid the many-class issues, too many stops in a bus line still challenge machine learning models. Fourthly, this study is based on only seven bus lines. There should be many ignored trips made by those passengers on other bus lines. So, the dataset does not include all kinds of travel behaviour. This data limitation may cause the bias that the model only learns the boarding behaviour in the data but miss the boarding behaviour travelling outside the seven-line bus network. When smart card data from the whole network is available, the model can learn more complete boarding behaviour. Finally, the error accumulation in the multi-stage framework is a problem. In the future, we need to clearly understand how the error comes and provide a more accurate predictive models in every stage. When the error in each stage reduces, the total error will drop down.

Next, the machine learning models, especially in Stage 3, have poor performances. Reflecting the aggregated ridership, FCN lacks the ability to capture the temporal characteristics while LSTM and RNN have a bad ability to assign the total ridership to stops. As we discussed in Section 3.5.3, we speculate that the imbalanced data and many-class issue still decreases the accuracy of the model. Although we adopted a multi-stage framework to avoid such problems, further investigation needs to work on dealing with these data issues in machine learning models. Last, this study only focuses on predicting boarding behaviour. However, OD matrix that contains both the boarding and alighting information is more important to describe the ridership and to guide the public transport planning. Therefore, combining the alighting stop prediction (Tang et al., 2020b) to this study will be a way to understand the travel pattern clearly.

Chapter 4

Modelling hourly bus passenger demand with imbalanced data

4.1. Introduction

The rapid progress of urbanisation leads to the expansion of population in the urban area. At the same time, people's need to travel is also growing up, which causes an increase in traffic congestion, energy consumption, and environmental pollution (Guo et al., 2016; Wu et al., 2020b). Public transport has been widely recognised as a green and sustainable mode of transportation to relieve such transport problems effectively. As a conventional mode in the public transport system, buses have always played a dominant role and are one of the most critical transport modes (Kwan and Hashim, 2016). However, a low image of unreliable services, crowding, bus bunching, and low level-of-services for buses has also been criticised (Berrebi et al., 2015; Fonzone et al., 2015; Schmöcker et al., 2016) and with the advent of ride-hailing services, the bus ridership has experienced a decline in many cities (Nelson and Sadowsky, 2019). In order to increase bus patronage, the bus system is required to find a way to enhance its attraction. Advanced operation and management for bus system can significantly improve those backwards and enhance the level-of-service (Liu and Sinha, 2007; Sorratini et al., 2008). It is not only necessary to understand the characteristics of the bus network, but also to understand the passenger behaviour for a sound service

planning and management (Hollander and Liu, 2008; Wu et al., 2016; 2017; Wu et al., 2019).

The smart card system is initially designed for automatic fare collection, but it also records the boarding information, for example, who gets on buses at where and when. In this way, smart card data has been a cheap and valuable data source for public transport planning (Bordagaray et al., 2016; Tang et al., 2020b; Zhang et al., 2018). From the smart card data, we can easily observe the flow at bus stops and on bus lines and reflect the spatial and temporal characteristic of bus trips (Yang et al., 2019b; Sun et al., 2016). Meanwhile, the big data also poses challenges to our analysing approaches, and the machine learning techniques have been approved to be an efficient and fast approach to deal with a lot of smart card data.

In our recent research (Tang et al., 2020a), we have demonstrated that smart card data, using machine learning techniques, can be a promising approach for predicting the spatial and temporal patterns of bus boarding. However, the boarding from a specific stop at a given time window is a rare event: most of the records denote negative (non-travelling) instances, and only a few are positive (travelling) instances. Such data imbalance in the smart card records can significantly reduce the efficiency and accuracy of machine learning models deployed for predicting hourly boarding numbers from a particular location. This motivates this study where we propose an over-sampling method, Deep-GAN model (developed in the context of image generation) to address the data imbalance issue in the context of predicting the boarding demand of travelling or not based on the synthetic dataset. The performance of the proposed approach is compared with other resampling methods (e.g. SMOTE and Random Under-Sampling).

The rest of the paper is organised as follows. Section 4.2 reviews some key resampling methods to balance datasets and their application in transport studies. Section 4.3

describes the data imbalance issue in predicting the hourly boarding demand. Section 4.4 uses a Deep-GAN model to over-sample the training data and a DNN to predict the boarding conditions (boarding or not) for every smart card user. Section 4.5 applies our methods in a real-life case, Changsha and the results are discussed in Section 4.6. Finally, Section 4.7 summaries the main findings and contributions of this paper and suggests future investigation.

4.2. Literature review

The data imbalance is a common issue of real-world data in terms of fault diagnosis, anomaly detection and many others (Ali et al., 2015). In this section, we will review the methods to re-balance the datasets and their applications in transport systems (summarised in Figure 4.1).

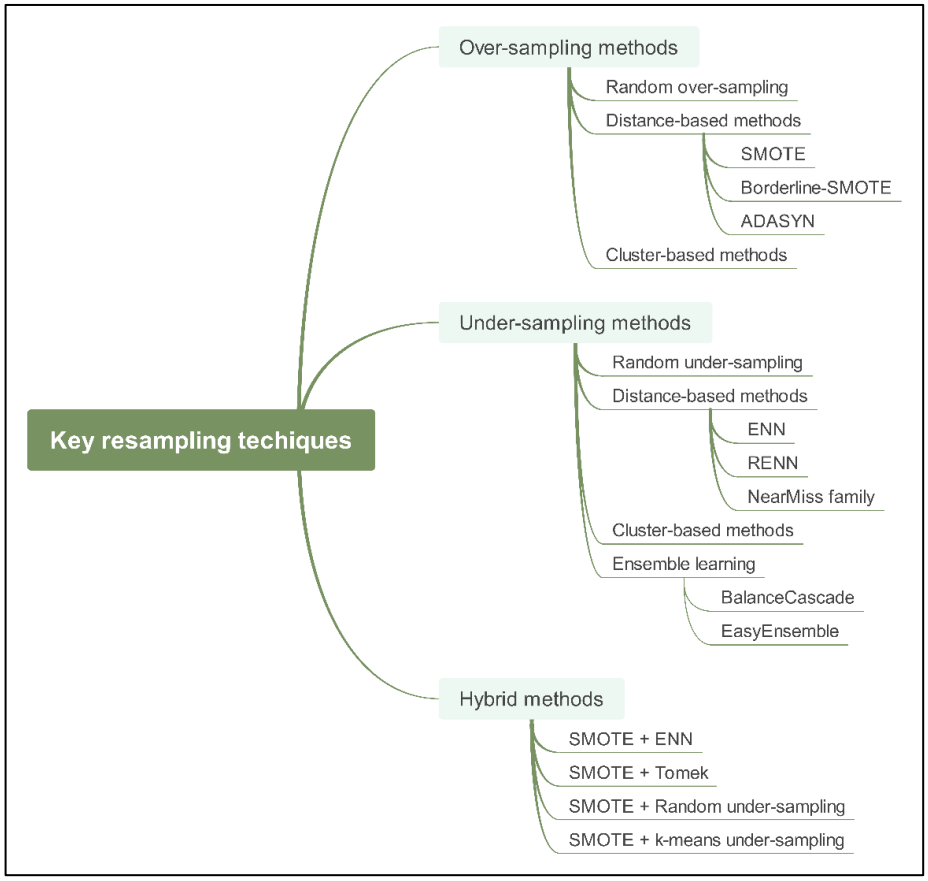

Figure 4.1 Reviewed key resampling techniques.

4.2.1. Resampling methods to balance datasets

Classic machine learning models tend to deal with problems where the number of instances in every class are roughly the same. It is the case for many standard datasets commonly used to test models, including the MNIST data for hand-wiring recognition (LeCun et al., 2010), Iris Plants Database for pattern recognition (Dua and Graff, 2019) and ImageNet data of image recognition (Russakovsky et al., 2015). In many practical problems, however, the data is not all as good as those standard datasets. A particular issue is data imbalance, where there is a minority of positive instances, while the majority is negative instances. For example, when detecting dangerous behaviour and rare activities (Azaria et al., 2014; Gao et al., 2016), the event (dangerous) instances is much less than the normal instances. The skewed distribution of classes, or imbalanced data, challenges the traditional machine learning models. Some possible reasons for the drawback of data imbalance on machine learning models are summarised: i) rare minor instances may be treated as noise and vice versa (Beyan and Fisher, 2015); ii) minor instances overlap with other regions where the prior probabilities of classes are almost equal (Denil and Trappenberg, 2010); iii) it is challenging to detect rate patterns for the minor instances with high feature dimensions because of the lack of density (López et al., 2013).

There are many methods developed to cope with the data imbalance. The basic idea to address this data issue is producing a new balanced dataset, called the resampling method. There are three main resampling methods: over-sampling, under-sampling and hybrid method (Guo et al., 2017). The over-sampling method is to create some imitated instances belonging to the minority class, while the under-sampling method is to remove some existing instances from the majority class. The hybrid method is to use the over-sampling for the minority and the under-sampling for the majority.

The simplest over-sampling method is random over-sampling (Batista et al., 2004), which randomly duplicates those minor instances. The new dataset by random over-sampling emphasises the existing minor instances which cause the risk of overfitting. Many repetitive instances may lead to the specification of the classifier on the dominant instances. Also, if some of the instances are mislabelled or noisy, the error can easily be multiplied. Therefore, a distance-based over-sampling method, SMOTE (Chawla et al., 2002), and its modified version are developed. SMOTE stands for synthetic minority over-sampling technique. It uses k-nearest neighbours (kNN) algorithm to calculate the closest instances of each instance in the minority, randomly selects several neighbours according to the imbalanced rate and randomly generates a new instance between the central instance and neighbour instances. SMOTE does decrease the risk of overfitting. However, this method increases the possibility of overlapping, and it cannot provide new instances with useful information. Han et al. (2005) propose an improved SMOTE method, Borderline-SMOTE. Unlike the random selection of minor instances in SMOTE, Borderline-SMOTE does the over-sampling only for the minor instances which are close to the border of minority category. He et al. (2008) developed an adaptive synthetic sampling (ADASYN) method, which adds weight on minor instances according to the number of nearby major instances and then generates a different number of new instances. These methods improve the quality of generated instances comparing to the SMOTE. However, it cannot deal with some common failing of SMOTE: i) The spared data distribution, feature redundancy and feature irrelevance in high-dimension data challenge the algorithm to identify minor instances, and ii) The SMOTE method is not suitable for the categorised features (e.g. seasons) because their distance cannot be calculated. Besides distance-based methods, clustering can also be used for over-sampling. Jo and Japkowicz (2004) make use of the K-means

clustering to categorise the imbalanced dataset and does the over-sampling process for each category of data.

As opposed to over-sampling, under-sampling method is another technique in resampling. Like the random over-sampling, the method of random selection is also applied in the under-sampling, which randomly removes some major instances. This method abandons a lot of data and information and may result in bias and overfitting in learning. Similarly, the distance-based and clustering-based approach can also be used in under-sampling. In distance-based methods, Wilson (1972) proposed the edited nearest neighbour (ENN) to balance the data. This method looks for and removes the major instances that are surrounded by the instances in the minority. Repetitive edited nearest neighbour (RENN) applies the ENN repeatedly until all neighbours of the major instance are within the majority class (Tomek, 1976). Besides, Mani and Zhang (2003) propose four NearMiss-family methods that use the kNN algorithm to select major instances. NearMiss-1 selects the major instances with the smallest average distances to three closest minor instances; NearMiss-2 selects the major instances with the smallest average distances to three farthest minor instances; NearMiss-3 selects a predefined number of the closest major instances for every minor instance; MostDistance selects the major instances with the most massive average distances to three closest minor instances. For the clustering-based method, Yen and Lee (2009) partition all the data into several clusters and randomly select the instances from each cluster. Zhang et al. (2010) apply k-means clustering in the partition process and use the number of major instances in every cluster as a weight to decide the number of selected major instance. As stated before, the under-sampling method may miss some information. To overcome the weakness of missing information, two methods, EasyEnsemble (Liu, 2009) and BalanceCascade (Liu et al., 2009), are developed. EasyEnsemble uses the idea of ensemble learning. It under-samples the majority with

replacement and generates several independent, balanced training datasets. Then, these datasets are trained for their base-classifier, and these base-classifiers are combined with ensemble learning approaches such as Bagging. BalanceCascade uses the idea of boosting learning. It generates a new balanced training dataset by the under-sampling method and trains a base-classifier. Then, the method only puts back the major instances that are wrongly classified for the next-round under-sampling, and so on. These ensemble methods contain most of the information from the majority from a global perspective. However, Ha and Lee (2016) evidence that the under-sampled data in the majority do not follow the original distribution, which tends to build a biased decision boundary. According to the data size of the minority class, Zhou (2013) and Loyola-González et al. (2016) suggest that the under-sampling is a more appropriate method for the training data with more minor instances while the over-sampling is more suitable to deal with the training data with less minor instances.

If the size of the training dataset is too large, there are some research using the hybrid method to apply the over-sampling and under-sampling together. For example, the integration of SMOTE with ENN and SMOTE with Tomek links are developed by Batista et al. (2004). Jian et al. (2016) use SMOTE over-sampling and multiple random under-sampling with ensemble learning in support vector machine classification. Song et al. (2016) combine SMOTE over-sampling and k-means under-sampling method to resample the dataset. The hybrid method applies the over-sampling to the minority and the under-sampling to the majority, which always yields a better result than either in isolation (More, 2016). Although the hybrid method shows a greater ability on rebalancing dataset than using a single method, the performance of the different combination of methods also varies (see the examples of Ramentol et al., 2012; Liu et al., 2017; Gazzah et al., 2015).It will takes much time to test which combination is the best selection for our case.

4.2.2. Application of data imbalance issue in transport domain

We typically face data imbalance in real-life applications where the minority class is the usually more important one (Krawczyk, 2016). In the field of transport, the problem of data imbalance is especially acute in accident detection, where the data representing accidents is in a rare minority while the usual scenario is in the majority. For example, Park and Ha (2014) prove that the over-sampling by the data mining tools, Hive and MapReduce, can improve the precision in prediction traffic accidents. Parsa et al. (2019) compare the performance of the SMOTE, Borderline-SMOTE and SVM-SMOTE used in over-sampling the minor accident instances. They find that all three methods have similar accuracy, but SMOTE tends to have a higher detection rate and lower false alarm rate. Sharifirad et al. (2014) enhance the SMOTE method for over-sampling the accident data, which weights the distance used in the kNN algorithm by the information entropy of attributes. Cai et al. (2020) use GAN-based model to generate the matrix of describing the car crash, which can provide a smoother distribution than other SMOTE and random over-sampling.

Besides accident recognition, other fields in transport also face problems with data imbalance, such as recognising vehicle types (Dabiri et al., 2020) and identifying commercial vehicle activities (Low et al., 2020) Hajizadeh et al. (2016) use semi-supervised techniques such as self-training and co-training to identify and add minor instances for detecting the rail defect from rail image data. The results by semi-supervised techniques are better than other classic over-sampling methods like SMOTE and random over-sampling. Similarly, Mohammadi et al. (2019) apply the ADASYN method to overcome the imbalanced data issue when predicting rail defects by track geometry measurement dataset. Rahaman et al. (2017) predict the queue context in the airport through the imbalanced taxi and passenger queue contexts. The conclusion of their study suggests that the balanced dataset with any resampling method is much better

than the original dataset in every evaluation index and the random over-sampling performs best.

4.3. Passenger boarding instances from the smart card data

4.3.1. Description of the data imbalance issue

The target of this study is to predict the hourly boarding demand for bus systems. We model the passengers' boarding status, travelling or not, as the measure of demand. Thus, the instance in this study is the trip made by a passenger during a predefined time slot. Here, we select one hour as a time slot in this study. The state of a trip in this study is characterised as travelling or non-travelling. The instance consists of a feature vector describing the passenger and the time of travel, and a label identifying the state of trips. The instance is expressed as follows:

$$r_t^p = \left(x_t^p, y_t^p\right) \tag{4.1}$$

where $r^p{}_t$ denotes the trip r of passenger p during the time slot t; $x^p{}_t$ is a feature vector describing the trip $r^p{}_t$; and $y^p{}_t$ is the label to estimation on the travel behaviour of trip $r^p{}_t$. Feature vector, x, contains several features that characterise the trip, such as boarding hour, temperature, the total number of trips on the previous day, etc. The individual features are denoted as v_1, v_2, etc. Features selected in this study will be introduced in the next section. The label, y, represents the state of the trip: 1 denotes the travelling trips, and 0 denotes the non-travelling trips.

$$x = \left(v_1, v_2, v_3, \cdots\right) \tag{4.2}$$

$$y = \begin{cases} 1, & \text{travelling} \\ 0, & \text{non-travelling} \end{cases} \tag{4.3}$$

Since the non-travelling trip is much more than travelling trips, the dataset is imbalanced. For example, for a typical 19-hour operation time of a day, we find that in the dataset used in this study, there are just 43 thousand travelling instances but about two million non-travelling instances in a day. The ratio of minority (travelling) class to majority (non-travelling) class is 1:44 in the case-study dataset, or only 2% is travelling instances while the remaining 98% represents non-travelling instances. As stated before, the skewed distribution can lead to bias in learning towards the pattern of non-travelling instances and significantly decrease the accuracy of the machine learning model. In the next section, we use the deep generative adversarial network to balance the data.

4.3.2. Data cleaning and pre-processing

The smart card data is initially designed for automatic fare collection. Each log in the smart card data records a bus trip, and more specifically, a boarding record.

At first, we make the following assumptions to clean the data:

- Each card ID corresponds to a single passenger, and each passenger swipes the card only once at a single boarding. In a real-life situation, some passengers may (accidentally) swipe their cards more than once for one boarding, which causes two and more transactions during a short period. We consider these data as repetitive data.

- The number of bus trips made by each smart card users is reasonable. There is a situation in the smart card data that an ID appeared over 50 times in one day and never showed up on any other day. We, therefore, recognise the ID appearing over 19 times (at most once an hour on average) a day as

the testing smart card from the bus company and remove these IDs from the database.

- This study only focuses on the regular smart card users who travel at least once a week. Many card IDs appear only a few times in the data. To simplify the data, we exclude the infrequent users as those who travelled less than five times during the study period, because this kind of users will generate too many non-travelling instances increasing computational burden but providing limited information to the models.

After the initial data cleaning, we transform the remaining smart card data into instances described in Eq. (4.1) and label them according to Eq. (4.3). The cleaned and transformed dataset is used in data generation to balance the dataset and in bus boarding demand prediction, described in the following sections.

4.4. Hourly boarding demand prediction coping with an imbalanced dataset

This section will present how to predict boarding behaviour and method developed to cope with an imbalanced dataset. As shown in Figure 4.2, the data resources are the smart card data that offers the raw travelling data, GPS data that complements geographic information to smart card data for the boarding stop inference, and weather data that provides the impacts of weather conditions on the boarding behaviour. Through the preparing processes (explained in Section 4.3.2), we can obtain the original imbalanced dataset. Then, all the instances in the original dataset will be divided into three sub-datasets: training, validation and testing dataset. We will apply a deep generative adversarial network (Deep-GAN) to produce artificial travelling instances

for balancing the training dataset. The details in the Deep-GAN model will be presented in Section 4.2.1. Next, the synthesised training dataset will be used to train the DNN-based predictive model to predict the state, travelling or not, and the validation dataset will be used to examine the situation of the training process. The architecture of the predictive model will be described in Section 4.2.2. Finally, we apply the testing dataset to the well-trained model to measure the performance of the model.

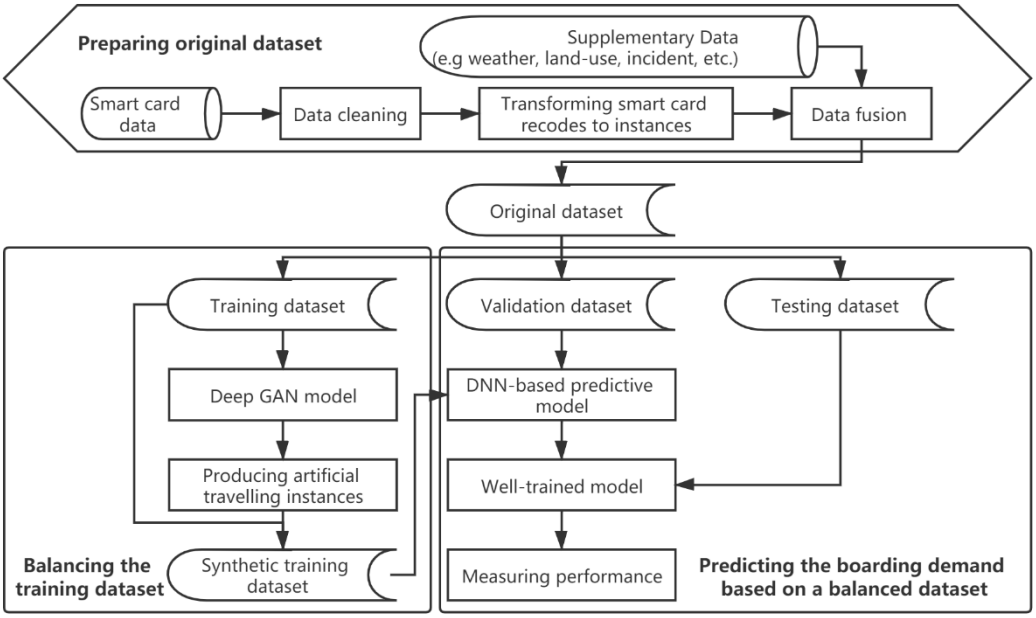

Figure 4.2 The flow chart on predicting the boarding behaviour from an originally imbalanced dataset.

4.2.1. Deep generative adversarial network to balance the dataset

The generative adversarial network (GAN), firstly introduced by Goodfellow et al. (2014), is a deep learning architecture which consists of two multi-layer perceptrons (a generator and a discriminator). Figure 4.3 illustrates the basic architecture of the GAN model. The generator utilises the noise vector, which is made of random numbers, to produce a fake data in the target format. The discriminator tries to distinguish between real data and fake data.

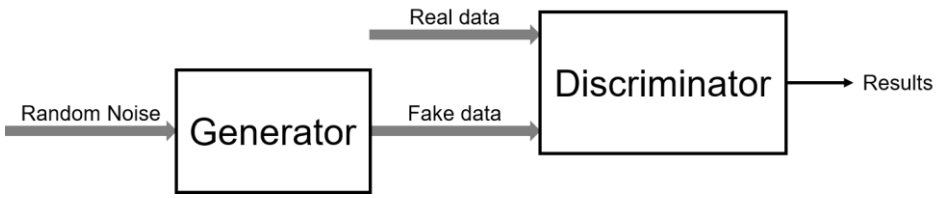

Figure 4.3. The basic structure of the GAN model.

Deep generative adversarial network (Deep-GAN) is one of the updated GAN-based models. The framework of Deep-GAN is illustrated in Figure 4.4. The Deep-GAN replaces the perceptron used in naïve GAN by a deep neural network (DNN). DNN is an extension of the artificial neural network (ANN) in the field of deep learning. The 'deep' in the name means DNN has deep layers in the architecture. In the generator, the activation function between every two layers is the Rectified Linear Unit (ReLU) function except for the output layer that uses the tanh function. The Leaky ReLU function is used on each layer in the discriminator.

$$\tanh(a) = \frac{\sinh a}{\cosh a} = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} \tag{4.4}$$

$$\mathrm{ReLU}(a) = \max(0, a) \tag{4.5}$$

$$\text{Leaky ReLU}(a) = \begin{cases} a, a > 0 \\ \lambda a, a \le 0 \end{cases} \tag{4.6}$$

where a is the value of the nodes in DNN layers and λ is a random parameter between 0 and 1.

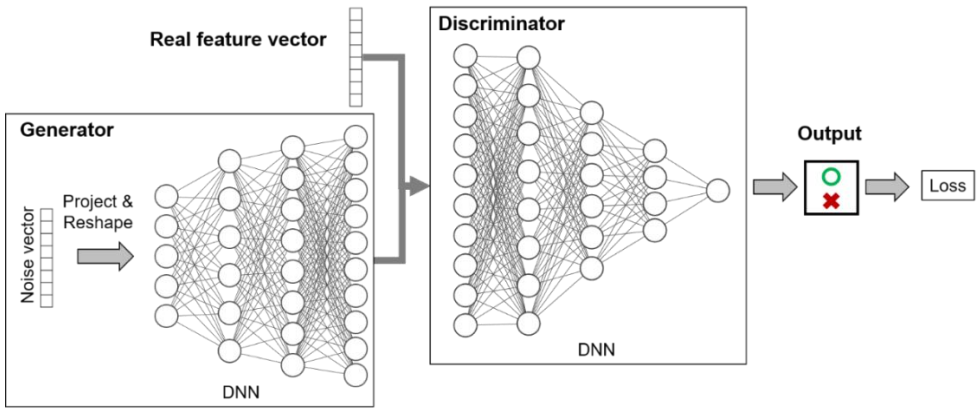

Figure 4.4 The architecture of Deep-GAN model.

We define a discriminator D to identify if a data is sampled from the distribution of real travelling data $p_{data}(x)$. The performance of the discriminator is measured by a logarithmic loss function of the positive instances that data are recognised as the real travelling data:

$$\mathrm{F}_D = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] \tag{4.7}$$

where $\mathbb{E}[.]$ denotes a calculation of the expectation. Maximising Eq. (4.7) means that D can correctly predict $D(x)=1$ when x follows the probability density of real travelling data, which is expressed as:

$$D(x)=1, x \sim p_{data}(x) \tag{4.8}$$

On the other side, the role of generator G is to deceive D by generating fake data. Here, we build up the loss of the generator using a logarithmic loss function of negative instances so that data cannot be recognised as the real travelling data.

$$\mathrm{F}_G = \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \tag{4.9}$$

where we premise that $p_z(z)$ is the prior distribution of random noise z used in the generator. The objective of the GAN model is formulated as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \tag{4.10}$$

D and G play two-player minimax game with value function $V(D,G)$. D tries to maximise the value, which represents the best ability of discrimination. At the same time, G wants to a minimum the value when the generated fake travelling data is much closer to the real travelling data.

4.2.2. A deep neural network (DNN) for predicting the boarding demand

After applying the resampling techniques, the dataset would be relatively balanced. Using the balanced dataset, we predict boarding behaviour using a DNN-based predictive model. Comparing to a simple ANN model, DNN may have more hidden layers, as illustrated in Figure 4.5.

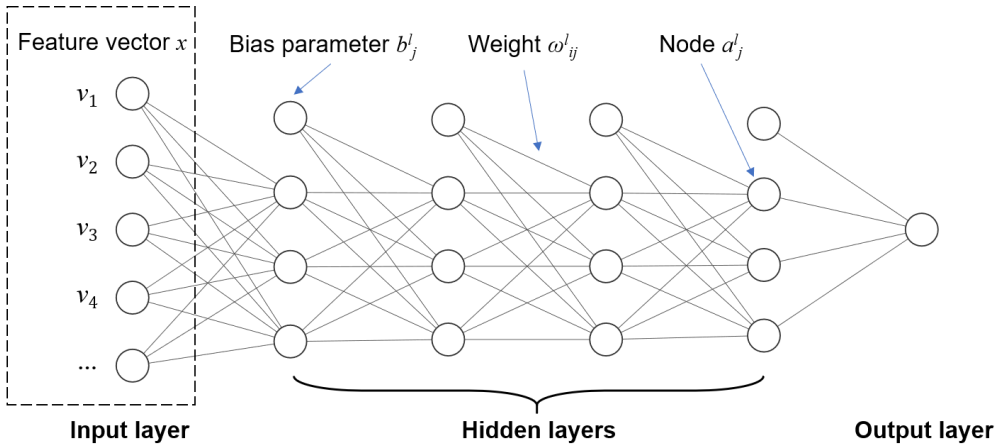

Figure 4.5 An example of DNN's architecture.

The feature vector x comes into the DNN model via the nodes in the input layer, where each node represents a dimension of features. Then, we calculate the nodes in the next layers:

$$a_j^l = \sigma\left(z_j^l\right) = \sigma\left(\sum_{i=1}^{I} \omega_{ij}^l a_i^{l-1} + b_j^{l-1}\right) \tag{4.11}$$

where a^l_j represents the value of the node j in layer l (input layer is the first layer, l=1); z^l_j is the weighted accumulated results from the nodes in the layer l-1 and the bias parameter b^{l-1}_j; $\sigma(\cdot)$ represents the activation function. The weighted accumulation process is based on the value of each node 1 to I in layer l-1 and the weight ω^l_{ij} from node i in layer l-1 to node j in layer l. Following the architecture of DNN, the

information in the feature vector will be revised and transferred through the hidden layers and to the output layer.

4.2.3. Evaluating the prediction results

Confusion matrix (CM) is one of the most used measurements for the classification problem (Godbole and Sarawagi, 2004). The CM for binary classification problem is demonstrated in Table 4.1. CM has two dimensions: real and predicted travelling behaviour, and each dimension has two situations: positive (travelling) and negative (non-travelling). So, each instance can be assigned to only one of the following four situations:

- True positive (*TP*): travelling instance is correctly predicted as travelling instance.
- Ture negative (*TN*): non-travelling instance is correctly predicted as a non-travelling instance.
- False negative (*FN*): travelling instance is wrongly predicted as a non-travelling instance.
- False-positive (*FP*): non-travelling instance is wrongly predicted as travelling instance.

Table 4.1 Confusion matrix for binary classification problem.

		Real travelling behaviour	
		Travelling (Positive)	Non-travelling (Negative)
Predicted travelling behaviour	Travelling (Positive)	True positive (*TP*)	False positive (*FP*)
	Non-travelling (Negative)	False negative (*FN*)	True negative (*TN*)

According to CM, we calculate the precision and recall performance of the model. Precision is the fraction of TP instances among all the predicted travelling instances, which reflects the ability to identify only the relevant instances; Recall is the fraction of TP instances among all the real travelling instances, which expresses the ability to find all relevant instances. The precision and recall describe the two side of the model. Thus, the F-measure in Eq. (4.14) has been proposed in order to have a comprehensive consideration of precision and recall. When β=1, we can obtain the most common and classic performance metrics, F1-measure, to evaluate the overall performance of machine learning models.

$$precision = \frac{TP}{TP + FP} \tag{4.12}$$

$$recall = \frac{TP}{TP + FN} \tag{4.13}$$

$$F_\beta = \frac{\left(\beta^2 + 1\right) * precision * recall}{\beta^2 precision + recall} \tag{4.14}$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{4.15}$$

4.5. Case study

4.5.1. Smart card data resource

The smart card data used in this study records the trips on seven bus lines from the bus network in Changsha, China. The dataset covers the period from 1st August to 1st September 2016. The operation time is nineteen hours between 5 am to 1 am of the next day. The raw dataset includes 2,917,272 transactions with 564,803 unique smart card IDs. Following the screening criteria of Section 3.2, 1,279,290 transactions from 101,850 smart card ID are retained. As shown in Table 4.2, the smart card data records

eight fields: bus company, bus line, vehicle, engine ID of vehicle, smart card ID, smart card type, data, and boarding time. There are no specific boarding stops in the data. In this study, we are concerned only with whether travelling or not; we do not consider (or estimate) the bus line and stops they used.

Table 4.2 An excerpt from smart card data.

Company	Line	Vehicle	Engine	Card_ID	Type	Date	Time
200	6	202111	159804	18725002	1	2016/8/1	11:43:15
200	6	202311	161502	18725002	1	2016/8/1	14:32:59
…	…	…	…	…	…	…	…
200	147	201674	128150	18729273	1	2016/8/1	16:14:51
200	123	201869	145477	17991759	1	2016/8/1	16:14:51
…	…	…	…	…	…	…	…

4.5.2. Feature selection

We choose the features from three domains: boarding time, weather conditions and travel history, because these three domains all have an impact on passengers' decision-making during bus trips (Tang et al., 2020b). All features are listed in Table 4.3. In the domain of boarding time, we use the season and day of the week to describe the date, and a binary feature, holiday, to distinguish between holidays, including weekends and working days. Additionally, we use the time slot to restrict the time of travelling behaviour. To avoid multiple trips in a time slot, we determine that a time slot is one hour so there are 19 time slots in a day. For the domain of weather conditions, we include a range of independent weather indices in features listed in Table 4.3. Also included as a weather feature is the air pollution index (AQI) as a potential influencing

factor on travelling behaviour. Travel history describes the passengers' regularity of using the bus services. This study considers two time points: the previous day (expressed as day-1 in the table) and the same day in last week (expressed as day-7 in the table), and the period between these two time points.

Table 4.3 Investigated domain of features employed in machine learning models.

Feature types: C- Categorical; Nom-Nominal; Num-Numerical.

Feature domains	Features	Dimensions	Feature types	Explanation
Boarding time	Season	4	C	Spring; summer; autumn; winter.
	Day of the week	7	C	Mon., Tues., Wed., Thurs., Fri., Sat., Sun.
	Holiday	2	C	Holidays and working days.
	Time slot	1	Num	One-hour time slot from 6 am on a given to 1 am on the next day
Weather condition	Temperature	1	Num	The average temperature during the time slot
	Precipitation	1	Num	Total precipitation during the time slot
	Humidity	1	Num	Average relative humidity during the time slot
	Visibility	1	Num	Minimum visibility during the time slot
	Wind speed	1	Num	Maximum instantaneous wind speed during the time slot
	Weather events	6	C	Clear, Cloudy, Fog, Overcast, Rain, Unknown
	AQI	1	Num	Air quality index

Feature domains	Features	Dimensions	Feature types	Explanation
Travel history	Card ID	17	Nom	Unique ID to identify the card users
	Total number of trips on day-1	1	Num	Number of trips made by the passengers on the previous day
	Total number of trips on day-7	1	Num	Number of trips made by the passengers on the same day last week
	Total number of trips from day-7 to day-1	1	Num	Number of trips made by the passengers on all previous seven days
	Total number of trips in the same time slot on day-1	1	Num	Number of trips made by the passengers in the same time slot on the previous day
	Total number of trips in the same time slot on day-7	1	Num	Number of trips made by the passengers in the same time slot on the same day last week
	Total number of trips in the same time slot from day-7 to day-1	1	Num	Number of trips made by the passengers in the same time slot on all previous seven days

Features are described in two data types: numerical and categorical. Numerical features can be used directly for the calculation. However, different features have different dimensions and units, which results in non-comparability between features. Here, a min-max normalisation on all numerical features is carried out, as follows:

$$v = \frac{v - v_{\min}}{v_{\max} - v_{\min}} \tag{4.16}$$

where v is a numerical feather in feather vector x and v_{min} and v_{max} respectively represent the minimum and maximum value in the feature v. After the normalisation, all the numerical features are converted to a dimensionless value between 0 and 1.

Categorical features are each assigned a unique value simply to register their categories; there is no direct relation or comparison can be made between categories. Here, we use One-hot Encoding to present a categorical feature using a sparse vector. For example, the feature of the holiday has two categories: 'holiday day' and 'working day'. We use a vector with two dimensions to describe this feature. The vector (0,1) represents the category of 'holiday', and the vector (1,0) represents the category of 'working days'. Moreover, the nominal feature, Card ID, is a special kind of categorical feature. The process of One-hot Encoding can generate sparse vectors with extremely high dimensions for this nominal feature. Thus, we use the feature hashing (Weinberger et al., 2009) to represent such categorical/nominal feature. There is a total of 18 features and 49 dimensions in the feature vectors.

4.5.3. Experimental design

The resampling and prediction processes are conducted via Keras (Chollet and Others, 2015) with the Python programming language. All the experiments are run on an Aliyun cloud graphics processing unit (GPU) platform with an NVIDIA® V100 Tensor Core GPU and 32 GB GPU memory.

As the features include the travel behaviour on the previous seven days, we use data from 8th to 31st August as the combined training and validation dataset and 1st December as the testing dataset. 80% instance of the combined dataset is randomly selected to be the training dataset, and the rest of 20% instance is used as the validation dataset. After the data pre-processing, it remains 101,850 smart card users and 1,935,150 instances in

a day (19 instances per smart card user). The number of instances in the original datasets is listed in Table 4.4.

Table 4.4 The number of instances in the original datasets.

Types of instances	Training dataset	Validation dataset	Testing dataset
Travelling instances	819,278	204,820	45,159
Non-travelling	36,335,602	9,083,900	1,889,991
Imbalanced rate	1:45	1:45	1:42
Total	37,154,880	9,288,720	1,935,150

Table 4.5 The synthetic training datasets with different imbalanced rates (by Deep-GAN).

Experiments	Real travelling instances	Synthetic data	Non-travelling instances	Imbalanced rate	Total
BL (original)	819,278	0	36,335,602	1:45	37,154,880
$E_{1:1}$		35,516,324		1:1	72,671,204
$E_{1:2}$		17,348,523		1:2	54,503,403
$E_{1:5}$		6,447,842		1:5	43,602,722
$E_{1:10}$		2,814,282		1:10	39,969,162
$E_{1:20}$		997,502		1:20	38,152,382

The resampling method is applied to training dataset only. As the baseline (BL), the original data without any resampling method is directly used to predict travelling behaviour. The dataset is the same as the data using in Stage 1 in Chapter 3, but the predictive models are different. To analysis the impact of the balanced rate (minority to

majority), we design experiments $E_{1:20}$ to $E_{1:1}$ with different imbalanced rates. Table 4.5 records the number of synthetic data and their imbalanced rate of training datasets in $E_{1:20}$ to $E_{1:1}$. The synthetic data in $E_{1:20}$ to $E_{1:1}$ is generated by Deep-GAN.

In order to compare the different performance of Deep-GAN model to other existing over- and under-sampling methods, we select two the most commonly used over- and under-sampling methods, SMOTE method and random under-sampling (RUS) method respectively. Since the categorical features cannot be directly inputted in the SMOTE model, they are transferred to the numerical variable by LabelEncoder. The imbalanced rate is decided by the best performance of $E_{1:20}$ to $E_{1:1}$ above. So, the experiment with SMOTE method (E_{SMOTE}) has the same training dataset with $E_{Deep-GAN}$, and E_{RUS} with RUS has its own training dataset. Table 4.6 displays the number of synthetic training datasets in $E_{Deep-GAN}$, E_{SMOTE} and E_{RUS} with the imbalanced rate of 1:5.

Table 4.6 The synthetic training datasets resampled by different methods (imbalanced rate = 1:5).

Experiments	Method to balance data	Travelling instances	Non-travelling instances	Total
BL (original)	None	819,278	36,335,602	37,154,880
$E_{Deep-GAN}$	Deep-GAN	7,267,120	36,335,602	43,602,722
E_{SMOTE}	SMOTE	7,267,120	36,335,602	43,602,722
E_{RUS}	RUS	819,278	4,096,390	4,915,668

4.5.4. Model configuration

There are two DNNs in the Deep-GAN model for generation and discrimination. Table C.1 in Appendix C displays the configurations of the generator and discriminator in Deep-GAN model for $E_{1:20}$ to $E_{1:1}$. There are six layers in the generator, including the input layer. The generator is to reshape and transform the noise vector with eight dimensions sampled from the uniform probability distribution and to produce a 49-dimension tensor following the distribution of real travelling data. We use the ReLU function for the activation function between two layers and the tanh function for the activation function of the last layer. Moreover, we use a layer after the generator to normalise a batch of instances. The discriminator receives the tensor from both the generator and the real data and uses a five-layer deep neural network to distinguish whether the tensor is from the generator or the real data. In the discriminator, the Leaky ReLU function ($\lambda = 0.2$) is the activation function between two layers while the sigmoid function is the activation function for the output layer. The learning rate for both generator and discriminator is 0.0005; the batch size is 512; loss function is the binary_crossentropy function.

Table C.2 in Appendix C displays the configurations of the DNN-based predictive model with six layers. The input of this model is a 49-dimension tensor. The ReLU function is used to be the activation function after the hidden layers, and the sigmoid function is the activation function between the last hidden layer and the output layer. The learning rate for the predictive model is 0.0005; the batch size is 512; loss function is the binary_crossentropy function.

4.6. Results and discussions

In this section, we analyse the performance of the predictive models for the set of experiments designed above. We first examine the level of imbalanced rate in the training dataset, which increases the accuracy of the predictive model the most and which resampling method produces the best synthesised dataset. We will also analyse the aggregated results on the distributions of bus ridership.

4.6.1. Performance of predictive models

We use the same setting in our Deep-GAN model to generate different training dataset with different imbalanced rates and apply these training datasets to the same predictive model. Figure 4.6 shows the performance metrics, on the precision, recall and F1 of the predictive models trained by different training datasets with different imbalanced rates. The BL uses the original imbalanced training dataset of which imbalanced rate is up to 1:45. From $E_{1:20}$ to $E_{1:1}$, we gradually increase the number of synthetic travelling instances in the training dataset, and hence reducing the imbalanced rate of the training dataset. Overall, we can find that model precision is lower than the recall in all the experiments. The main issue of the imbalanced dataset is that the trained model learns more on major negative instances, which predicts more FN instances and fewer FP instances ($FP<FN$). This learning bias leads to a higher recall than the precision score. The predictive model based on BL has the worst performance, where all three metrics are measured around 0.55. This result of BL is only slightly better than that from a random classification. This suggests that using an imbalanced training dataset can result in very poor predictive models, and with extremely imbalanced data, the predictive model is no better than a random classification. As noted in Section 4.2.1 earlier, the reasons of poor performance on imbalanced data are: i) few travelling instances may be

recognised as the noise and ii) a large number of non-travelling instances leads to learning the pattern from non-travelling instances.

With reduced imbalance rates, the performance of the predictive model improves. Figure 4.6 shows that as the imbalanced rate reduces (from $E_{1:20}$ to $E_{1:1}$), the performance metrics increase. With an absolutely balanced training dataset ($E_{1:1}$), the values of all three metrics are over 0.88. Thus, a more balanced training dataset will get a more accurate prediction result. However, a more balanced training dataset with a significantly increased number of instances requires a higher-performance computer and significantly longer time to train. We note in Figure 4.6 that there is a significant improvement in the performance of the predictive models between $E_{1:10}$ to $E_{1:5}$, and the improvements from $E_{1:5}$ to $E_{1:1}$ is relatively small. On the balance of computational burden and model prediction accuracy, we consider $E_{1:5}$ with an imbalance rate of 1:5 as an acceptable choice.

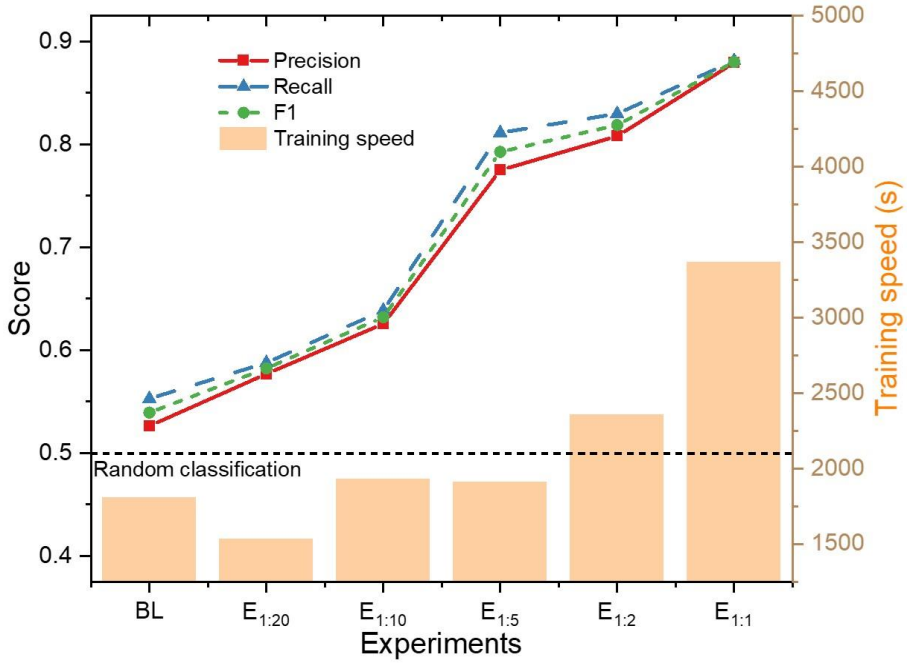

Figure 4.6 The performance metrics (Precision, Recall and F1) and training speed for the training datasets with different imbalance rates.

Next, we compare the performances of three different resampling methods. For the experiments $E_{Deep\text{-}GAN}$, E_{SMOTE} and E_{RUS}, we keep the same imbalanced rate (1:5) as suggested above. $E_{Deep\text{-}GAN}$ and E_{SMOTE} use the over-sampling methods Deep-GAN and SMOTE respectively, and E_{RUS} uses the under-sampling method RUS. Figure 4.7 displays the performance metrics of the predictive models for these experiments.

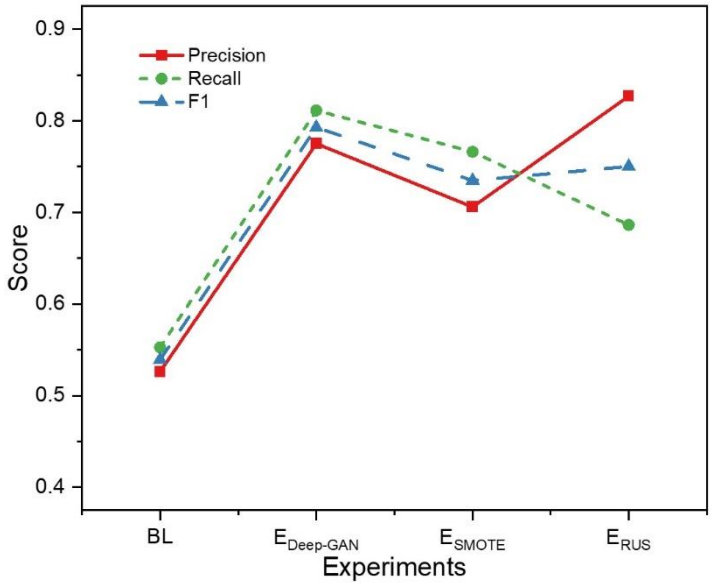

Figure 4.7 The performance metrics (Precision, Recall and F1) for the training datasets generated by different resampling methods.

Overall, the prediction results in $E_{Deep\text{-}GAN}$, E_{SMOTE} and E_{RUS} are much better than in BL and random classification. This result suggests that the accuracy of the predictive model will be improved as long as the imbalance rate can be reduced by any resampling method. Comparing $E_{Deep\text{-}GAN}$ and E_{SMOTE}, both use the over-sampling method to generate synthetic travelling instances in the travelling data. Both $E_{Deep\text{-}GAN}$ and E_{SMOTE} produce good prediction metrics. Using any performance metric, $E_{Deep\text{-}GAN}$ has a more accurate prediction than E_{SMOTE}, suggesting that the synthetic training dataset produced by Deep-GAN is better than that by SMOTE. E_{RUS} uses a different resampling method that randomly removes some non-travelling instances from the original dataset. According to F1 metric, E_{RUS} is also an acceptable method to produce a reliable and

balanced training dataset. The precision of E_{RUS} is much higher than $E_{Deep\text{-}GAN}$ and E_{SMOTE}, at 0.83; however, its recall only scores 0.69.

We note in Figure 4.7 that the precision scores of $E_{Deep\text{-}GAN}$ and E_{SMOTE} are greater than their respective recall scores, while for $E_{RUS,}$ the opposite is true. This is due to the different principles of the sampling methods. The number of FN instances is less than the FP instances in $E_{Deep\text{-}GAN}$ and E_{SMOTE}. It is because the two over-sampling methods artificially add extra information to the travelling instances, so the models are more likely to predict actual non-travelling instances as positive. On the contrary, the under-sampling method used for E_{RUS} deletes some non-travelling instances, which also reduces the information redundancy. Thus, the number of true negative instances increases, and the number of FP instances decreases, which contributes to the improvement of the precision score.

4.6.2. Distribution of estimated bus ridership

From the analysis on the perspective of the performance of machine learning models, we can see that the Deep-GAN model and DNN-based predictive model can accomplish the goal of predicting the boarding behaviour, travelling or not, for every smart card use at every hour from an imbalanced training dataset. An aggregated result of bus ridership is useful to public transport planners and operators, to guide a clear understanding and appropriate planning in the public transport system. In this section, we calculate the root mean square percentage error (RMSPE) and root mean square error (RMSE) to measure the accuracy of hourly ridership and analyse the distribution of bus ridership based on the individual estimation results of machine learning models.

$$RMSPE = \sqrt{\frac{1}{H}\sum_{h=1}^{H}\left|\frac{Rider_h - Rider'_h}{Rider_h}\right|^2} \tag{4.17}$$

$$RMSE = \sqrt{\frac{1}{H}\sum_{h=1}^{H}\left|Rider_h - Rider'_h\right|^2} \tag{4.18}$$

where $Rider_h$ and $Rider'_h$ represent the predicted and observed ridership at hour h and H is the total time slots, which equals 19 in our case.

Table 4.7 presents the RMSPE and RMSE of the first group of experiments (E1:1 to E1:20). After the resampling process, RMSPE and RMSE are significantly lower than BL. From the view of ridership, our method (Deep-GAN) can improve the performance of machine learning models on predicting boarding demand. E1:5 has the lowest RMSPE and RMSE. E1:5 contains more positive (travelling) instances than E1:20 and E1:10 and less than E1:2 and E 1:1. We will analyse this situation with the help of the profile of hourly ridership.

Table 4.7 The RMSPE and RMSE of hourly ridership by different imbalanced rate.

Experiments	BL	$E_{1:20}$	$E_{1:10}$	$E_{1:5}$	$E_{1:2}$	$E_{1:1}$
RMSPE	0.74	0.49	1.09	0.56	0.38	0.18
RMSE	2483.44	712.35	1114.84	281.26	912.15	785.62

Figure 4.8 shows the profiles of hourly ridership observed from smart card data (ground truth) and predicted by BL and $E_{1:20}$ to $E_{1:1}$. The observed ridership has two peaks: the morning peak from 7 to 9 am and evening peak from 6 to 7 pm. The prediction based on BL (totally imbalanced data) produced a delayed morning peak to 10 am, and very poor prediction on the amplitudes of two peaks: a much lower morning peak and a much higher afternoon peak, than the ground truth. $E_{1:20}$ and $E_{1:10}$ with minor improvements in balancing the dataset also predicted a delayed morning peak and underestimate the magnitude of the morning peak and overestimate the magnitude of the afternoon peak. By contrast, the prediction with $E_{1:1}$ (absolutely balanced data) and

$E_{1:2}$ accurately identified the timings of the two peaks. However, both $E_{1:1}$ and $E_{1:2}$ significantly overestimate the magnitude of the morning peak and to a less degree underestimate the magnitude of the evening peak. By comparison, using dataset $E_{1:5}$, the model accurately predicted both the timing and the magnitude of the peaks. It is understandable why BL performs poorly compared to $E_{1:20}$ to $E_{1:5}$, as imbalanced data leads to inaccuracy in machine learning models. We speculate that the errors in $E_{1:2}$ and $E_{1:1}$ estimation may be caused by information redundancy and repetition. The synthetic data follows not only the distribution of features but also the distribution of travelling instances. That is to say, the generated data has more data representing the travelling instances in two peaks. It emphasises the peaks and therefore causes bias in the hourly ridership. Even though $E_{1:1}$ has the best performance metrics, it does not lead to the best profile of ridership. It is because that the profile of ridership is produced by the positively-predicted instances including *TP* and *FP* in Table 4.1.

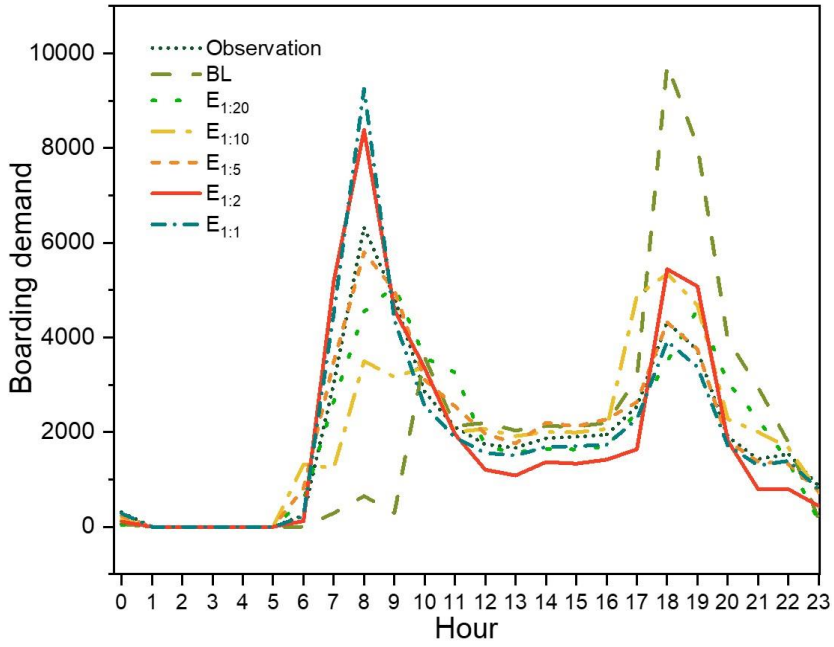

Figure 4.8 The profile of hourly ridership observed from smart card data and predicted by BL and E1:20 to E1:1.

Table 4.8 presents the RMSPE and RMSE of hourly ridership by different resampling methods. All the resampling methods benefits on the improvement of the accuracy of

predicting boarding demand. Our methods, Deep-GAN, is better than the SMOTE and RUS method.

Table 4.8 The RMSPE and RMSE of hourly ridership by different resampling methods.

Experiments	BL	$E_{Deep-GAN}$	E_{SMOTE}	E_{RUS}
RMSPE	0.74	0.56	0.70	0.84
RMSE	2483.44	281.26	483.77	650.85

Figure 4.9 presents the profile of hourly ridership predicted based on $E_{Deep-GAN}$ to E_{RUS} using different resampling methods and with imbalance rate 1:5. All three profiles of $E_{Deep-GAN}$ to E_{RUS} follow closely to the observed one and are significant improvements from BL. The results of Deep-GAN ($E_{Deep-GAN}$) and SMOTE (E_{SMOTE}) are very similar, while the prediction of ridership based on the RUS (E_{RUS}) is less than the observation during the bottom period and evening peak. It suggests that both over-sampling and under-sampling are valid methods for re-balancing the data and improving the accuracy of machine learning models.

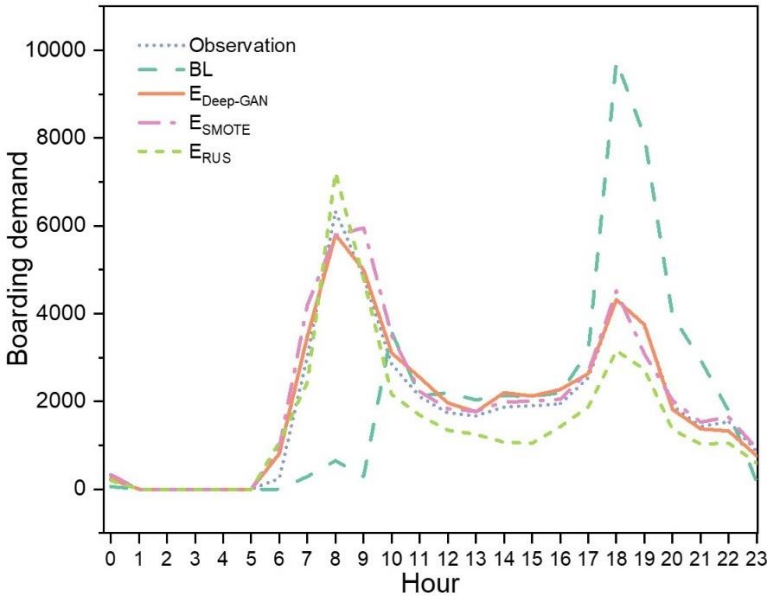

Figure 4.9 The profile of hourly ridership observed from smart card data and predicted by BL, $E_{Deep-GAN}$, E_{SMOTE}, and E_{RUS}.

4.7. Conclusions

In this research, we propose a Deep-GAN model to over-sample the travelling instances and to re-balance the rate of travelling and non-travelling instances in the smart card dataset in order to improve a DNN model of boarding behaviour. The performance of Deep-GAN is evaluated by applying the models on real-world smart card data collected from seven bus lines in the city of Changsha, China. Comparing the different imbalance rates in the training dataset, we find out that the performance of model gets better with the decrease of imbalance rate and there is a significant improvement in the performance when imbalanced rate decreases to 1:5. Comparing the different resampling methods, both over-sampling and under-sampling benefits to the performance of the model. Deep-GAN has the best recall score while RUS has the best precision score but the worst recall score. From the perspective of profile of ridership, the high imbalanced rate will cause the misleading of the profile and the absolutely balanced data may over-predict the ridership during peak hours.

It may be noted that despite the great performance of Deep-GAN and DNN models, there are still some limitations. Firstly, in this research Deep-GAN model is solely applied for the over-sampling. However, there is also a hybrid variant of Deep-GAN where positive instances are over-sampled and negative instances are under-sampled. The promising results of the Deep-GAN oversampling serve as a motivation to test the performance of the hybrid Deep-GAN in future research. Given that the RUS shows the excellent performance on the precision, one potential direction can be to combine the Deep-GAN with RUS. Secondly, this study makes the prediction at the individual level, which creates an explosion of information and makes the computation more difficult. Grouping the passengers may be useful in terms of reducing the size of the

dataset. Finally, the proposed Deep-GAN model used the features and instances independently.

However, even in its current form, the research demonstrates the extent of improvement offered by the Deep-GAN method in addressing the data imbalance issue in modelling boarding behaviour. By better predicting the boarding behaviour, the findings can help the public transport authorities to improve the level-of-service and efficiency of the public transport system. It can also be extended to other components of the public transport usage behaviour – better prediction of the alighting behaviour, for instance.

Chapter 5

Conclusion

5.1. Summaries

Buses, as an effective, economical and conventional public transport mode, have been catering to a major share of the travel demand not only from the public transport system but often the whole transport system as well. However, factors like poor level-of-service and low reliability of the bus system have led to the loss of passengers and impeded it from achieving its full potential. Accurate predictions of bus travel patterns (boarding demand, boarding pattern, alighting pattern etc.) enable bus operators and/or public transport authorities to make better planning, operational and management decisions that can contribute to the efficiency and patronage of the bus system. The availability of large-scale smart card data and advancements in machine learning techniques have opened up the potential to improve the prediction of bus travel patterns and motivates this dissertation. The main focus and findings are listed below:

a) Estimating the alighting stops of bus trips from their boarding information as recorded in smart card data and GPS records of bus journeys (Chapter 2)

For the automatic fare collection (AFC) system in many cities, smart card data records only the boarding information but no information about alighting behaviour. Estimating the alighting stops for such 'open' smart card data is the first task to produce the OD matrix for bus ridership. In this chapter, I propose a new data-driven framework

for the estimation of the alighting stop from the 'open' smart card data. A machine learning algorithm, gradient-based decision tree (GBDT) algorithm, is used to model the multi-class classification problem and the model incorporates the features of weather conditions and travel history to measure the regularity and variance of passengers' choice at alighting. The results show that GBDT outperforms traditional machine learning algorithms, e.g. MLR and ANN, on estimation precision. The inclusion of weather features further improves the precision of the estimation, while the inclusion of travel history enhances the recall ability of the model. Moreover, the GBDT-based model is able to rank the features by their importance to the prediction, and the results show that the boarding stop is the most important feature, followed by the weather event and the weather-related features are more important than the features about the travel history.

This work provides a tool to understand the alighting stops, especially for the trips that cannot be linked to a trip chain. It helps to have a comprehensive and accurate understanding of the OD of bus trips, which will contribute to more appropriate planning and operation for the bus system.

b) Predicting the hourly boarding behaviour for smart card users (Chapter 3)

Since there are more than one classes (candidate boarding stops) in the model, the problem belongs to the multi-class classification problem. Moreover, passengers may travel more than once (i.e. transfer trips), so there will be more than one boarding stop for one instance. Thus, directly predicting the boarding stops is a multi-label multi-class classification problem, and it faces the difficulty that many classes and many negative (non-travelling) instances in the data set will reduce the accuracy and effectivity of the model. In this chapter, I propose a multi-stage deep-learning-based framework to predict the hourly boarding behaviour for each smart card users. To avoid the data issue of many-class and data imbalance, the proposed framework disassembles the aim to three stages: whether to travel, which bus line to use and which bus stop to get on. FCN,

RNN and LSTM architectures are used to solve the multi-label classification problem. The results show that as a set of historical boarding records are combined into a time series to be the input vectors, LSTM and RNN is able to capture the temporal characteristics of the ridership (e.g. the peak hour and the demand at the peaks), which is better than the other machine learning model FCN and the classic method ARIMA, while FCN is better than RNN and LSTM to find the busy stops that have the most boarding demand.

This work provides a bottom-up model that minimises the traditional methods to model bus lines and stops. It also offers a more accurate prediction for the future condition of bus ridership, which guides the planning and operation to meet the travel demand.

c) Predicting the hourly boarding demand coping with the data imbalance issue (Chapter 4)

Although Chapter 3 uses a disassembling framework to avoid the data imbalance issue, the dataset for its Stage 1 is still imbalanced. In this chapter, I propose a novel resampling method, Deep-GAN, to improve the positive (travelling) instances in the dataset. The Deep-GAN employs a DNN to generate dummy travelling instances from random noise vector and another DNN to discriminate whether the generated travelling instances is like the real travelling instances. Then, the synthesised training dataset (original training data plus generated travelling instances) is used to train a DNN-based model for the boarding demand prediction. The results show that the dataset with the 1:5 imbalanced rate has the best performance when considering both the accuracy and efficiency of the predictive model. Comparing to the most used over- and under-sampling methods (SMOTE and RUS), the datasets generated by Deep-GAN can guide more accurate prediction on the boarding demand.

This work optimises the accuracy of the boarding demand prediction based on the work in Chapter 3. It also explores the usefulness of machine learning methods beyond predictive models (e.g. resampling the data).

5.2. Contributions

This thesis explores the application of various machine learning techniques in predicting the boarding and alighting demand for smart card users. This study shows the machine learning model is an effective technique to analyse the high-dimensional and massive smart card data. At the same time, there are still various issues to apply machine learning models such as data imbalance issue. This thesis also focuses on how to utilise algorithms and modify datasets for a best use of machine learning models. Comparing to the descriptive analyse of smart card data, this study supplements the details of the bus demand by estimating the alighting stops that do not appear in the raw smart card data. Also, this study extends duration when we can focus on the ridership by developing the predicting of ridership rather than analysing historical ridership in the past. Besides, models used in this these can incorporate more impacting factors and display how significant the impacts of factors are at the same time. The main contributions of this thesis are in the following aspects.

- Two novel frameworks are proposed to predict the boarding and alighting behaviour of smart card users. The first framework addresses the problem of how to estimate the alighting stop of bus trips with the absence of real alighting stops. The second framework addresses inaccuracy and poor performance of machine learning models caused by many candidate classes in models and imbalanced original dataset.

- The frameworks present the ability of different machine learning models. For example, the tree-based model (GBDT) can rank the importance of every feature, which helps understand the relationship between features and the boarding and alighting behaviour. LSTM and RNN is a promising approach for the sequential data (e.g. time series data), which can measure the temporal characteristics of boarding behaviour.
- The research demonstrates that a GAN-based model (Deep-GAN), developed and typically used in the domain of image processing and transport safety can be successfully used to deal with data imbalance issues in the context of predicting hourly bus boarding.
- The predictive models incorporate the impacts of travel history and weather conditions in predicting the boarding and alighting behaviour and present the importance of those features in the prediction.

Overall, machine learning, one of the most effective and useful techniques for predictions, is yet to reach its full potential in the domain of public transport planning. The dissertation contributes to achieving this by utilizing the remarkable ability of a wide variety of machine learning techniques to deal with the high-dimensional and multivariate data from smart cards.

The predictive models developed in this thesis are based on individual smart card users. The bottom-up model preserves heterogeneity of behaviour between individuals, which is ignored in the traditional prediction at the aggregated level.

5.3. Suggestions and future directions

The research applying machine learning techniques on bus ridership prediction has gained much attention in recent years. Although the field is developing very fast, it is far from mature. Further investigation of both machine learning techniques and their applications in transport are discussed below.

The accuracy of the prediction depends on the quality of data sources. Due to the absence of real alighting stops, the estimated alighting stops cannot be verified with reality. Calibration using data from cities where the boarding and alighting stops are known should be carried out. Since the case study contains only seven bus lines, the data does not adequately describe all the boarding and alighting behaviour of smart card users in the whole city of Changsha. The processes of data pre-processing may remove some smart card trips and users that are normal in the whole bus network but appear abnormal in our data. For example, a passenger is used to travelling on the bus lines other than these seven lines. An occasional ride on the studying bus lines is recognised as an anomaly record. There may be some potential bias in the trained model in this study. Therefore, it is necessary to re-examine the proposed approaches in the context of a more comprehensive bus network and more adequate data.

Moreover, the data covers only six months from April to September that belong to warm season. Thus, the data and trained models can only represent the boarding and alighting behaviour in the warm weather. The cases in this study ignore the boarding and alighting behaviour in the cold weather and cannot be applied to predict the demand in cold season. Therefore, I expect the data during a longer period, such as one year. A one-year data contains both warm and cold weather, which can obviate the bias in the model caused by the lack of data.

The data source used in this study is mainly the smart card data. This data ignores passengers who prefer to use cash or pre-paid monthly tickets. With the development of IoT and 5G, there will be more data sources, such as cellular signalling and Bluetooth data. The idea of multi-source data fusion can make up the limitation of smart card data. Also, some socioeconomic data, such as land-use, point of interest and population, can impact the choice of using bus stops. However, these data are confidential data controlled by the government. I cannot access them now. Therefore, a thorough consideration of multiple data sources to bring the prediction closer to reality will be the next research if the data is available.

This thesis introduces many kinds of machine learning techniques. These methods can accomplish different predictive aims well. However, the employed methods consider only the relationship of various features. The topology of the bus network and spatial-temporal information of ridership are ignored in these models. Combining such information may improve the performance of the models. Thus, it requires some more advanced machine learning techniques to process data about network, graph, and spatial-temporal characteristics.

All the predictions (on the alighting stops or the boarding demand) studied in this thesis take one hour to be the length of the time slot. The future work can try 15-minute and shorter time slot. The shorter time slot will facilitate more advanced planning and operation. Moreover, the predictive models in this thesis are one-step prediction, which only predicts the situation in next day. It is because one-step prediction is the foundation. In the future, the longer-step prediction is worth to be studied, for example, the bus ridership in next week and month.

5.4. The impacts of COVID-19

I encountered the break of COVID-19 while completing the research for Chapter 4. The global pandemic has kept me in China, and I cannot return to the UK to carry out my research. So, I continue my research at home in China. However, due to the loss of preliminary research results and the lack of access to computing facilities, my work has been greatly slowed down. I need more time to run the computer program; I need to wait for my colleagues to send me the data; I need to repeat the experiments in which the data was lost.

List of References

Aaheim, H.A. and Hauge, K.E. 2005. Impacts of climate change on travel habits: a national assessment based on individual choices. Oslo, Norway: Center for International Climate and Evironmental Research.

Ali, A., Shamsuddin, S.M. and Ralescu, A. 2015. Classification with class imbalance problem: A review. *Intertiol Jourl of Advances in Soft Computing and its Applications.* **7**(8), pp.176-204.

Alpaydin, E. 2014. Introduction to Machine Learning. 4th ed. Cambridge, Massachusetts, USA: MIT Press.

AlRukaibi, F. and AlKheder, S. 2019. Optimization of bus stop stations in Kuwait. *Sustainable Cities and Society.* **44**, pp.726-738.

Andrle, S. 1986. Summary: Research need in transit bus maintenance. *Transportation Research Record.* **1066**, pp.1-3.

Arana, P., Cabezudo, S. and Peñalba, M. 2014. Influence of weather conditions on transit ridership: a statistical study using data from Smartcards. *Transportation Research Part A: Policy and Practice.* **59**, pp.1-12.

Aydin, İ., Karaköse, E., Karaköse, M., Gençoğlu, M.T. and Akın, E. 2013. A new computer vision approach for active pantograph control. In: *IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA),* Albena, Bulgaria. IEEE, pp.1-5.

Azaria, A., Richardson, A., Kraus, S. and Subrahmanian, V.S. 2014. Behavioral analysis of insider threat: a survey and bootstrapped prediction in imbalanced data. *IEEE Transactions on Computational Social Systems.* **1**(2), pp.135-155.

Bagchi, M. and White, P.R. 2005. The potential of public transport smart card data. *Transport Policy.* **12**(5), pp.464-474.

Banerjee, N., Morton, A. and Akartunalı, K. 2020. Passenger demand forecasting in scheduled transportation. *European Journal of Operational Research.* **286**(3), pp.797-810.

Barabino, B., Francesco, M.D. and Mozzoni, S. 2014. An offline framework for handling automatic passenger counting raw data. *IEEE Transactions on Intelligent Transportation Systems.* **15**(6), pp.2443-2456.

Barry, J., Freimer, R. and Slavin, H. 2009. Use of entry-only automatic fare collection data to estimate linked transit trips in New York City. *Transportation Research Record: Journal of the Transportation Research Board.* **2112**, pp.53-61.

Barry, J., Newhouser, R., Rahbee, A. and Sayeda, S. 2002. Origin and destination estimation in New York City with automated fare system data. *Transportation Research Record: Journal of the Transportation Research Board.* **1817**(1), pp.183-187.

Batista, G.E., Prati, R.C. and Monard, M.C. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter.* **6**(1), pp.20-29.

Beijing Transport Institute. 2019. *Beijing Transport Development Annual Report.* Beijing, China.

Berrebi, S.J., Watkins, K.E. and Laval, J.A. 2015. A real-time bus dispatching policy to minimize passenger wait on a high frequency route. *Transportation Research Part B: Methodological.* **81**, pp.377-389.

Beyan, C. and Fisher, R. 2015. Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition.* **48**(5), pp.1653-1672.

Bin, Y., Zhong-zhen, Y. and Bao-zhen, Y. 2006. Bus arrival time prediction using support vector machines. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations.* **10**(4), pp.151 - 158.

Böcker, L., Dijst, M. and Prillwitz, J. 2013. Impact of everyday weather on individual daily travel behaviours in perspective: A literature review. *Transport Reviews.* **33**(1), pp.71-91.

Bordagaray, M., dell'Olio, L., Ibeas, A. and Cecín, P. 2013. Modelling user perception of bus transit quality considering user and service heterogeneity. *Transportmetrica A: Transport Science.* **10**(8), pp.705-721.

Bordagaray, M., dell'Olio, L., Fonzone, A. and Ibeas, Á. 2016. Capturing the conditions that introduce systematic variation in bike-sharing travel behavior using data mining techniques. *Transportation Research Part C: Emerging Technologies.* **71**, pp.231-248.

Boschetti, F., Maurizi, I. and Cré, I. 2014. Innovative urban transport solutions: CIVITAS makes the difference.

Brakewood, C. and Watkins, K. 2019. A literature review of the passenger benefits of real-time transit information. *Transport Reviews.* **39**(3), pp.327-356.

Bruin, T.d., Verbert, K. and Babuška, R. 2017. Railway track circuit fault diagnosis using recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems.* **28**(3), pp.523-533.

Cai, Q., Abdel-Aty, M., Yuan, J., Lee, J. and Wu, Y. 2020. Real-time crash prediction on expressways using deep generative models. *Transportation Research Part C: Emerging Technologies.* **117**, p102697.

Cantarella, G.E. and De Luca, S. 2003. Modeling transportation mode choice through artificial neural networks. In: *4th International Symposium on Uncertainty Modeling and Analysis, ISUMA 2003,* College Park, MD, USA. IEEE, pp.84-90.

Carpio-Pinedo, J. 2014. Urban bus demand forecast at stop Level: space syntax and other built environment factors: Evidence from Madrid. *Procedia - Social and Behavioral Sciences.* **160**, pp.205-214.

Ceder, A. 2007. Public Transit Planning and Operation: Theory, Modeling and Practice. Oxford, UK: Butterworth-Heinemann.

Celikoglu, H.B. 2006. Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling. *Mathematical and Computer Modelling.* **44**(7-8), pp.640-658.

Chaudhary, S., Kaur, T., Aggarwal, N., Raman, B., Bansal, D. and Ramakrishnan, K.K. 2016. Bus boarding event detection using smartphone sensors. In: *2016 8th International Conference on Communication Systems and Networks (COMSNETS),* Bangalore, India. IEEE, pp.1-6.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research.* **16**, pp.321-357.

Chen, Q., Wen, D., Li, X., Chen, D., Lv, H., Zhang, J. and Gao, P. 2019. Empirical mode decomposition based long short-term memory neural network forecasting model for the short-term metro passenger flow. *PLoS ONE.* **14**(9), pe0222365.

Chollet, F. and Others. 2015. Keras. [Online].

Connor, J.T., Martin, R.D. and Atlas, L.E. 1994. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks.* **5**(2), pp.240-254.

Corman, F. and Kecman, P. 2018. Stochastic prediction of train delays in real-time using Bayesian networks. *Transportation Research Part C: Emerging Technologies.* **95**, pp.599-615.

Cryer, J.D. and Chan, K.-S. 2008. Time Series Analysis: With Applications in R. 2nd ed. Springer Science & Business Media.

Dabiri, S., Marković, N., Heaslip, K. and Reddy, C.K. 2020. A deep convolutional neural network based approach for vehicle classification using large-scale GPS trajectory data. *Transportation Research Part C: Emerging Technologies.* **116**, p102644.

Deloitte. 2018. Super smart city: Happier society with higher quality. Shanghai, China: Deloitte Public Sector, TMT Industry & Deloitte Research.

Deloitte. 2020. Super smart city 2.0: Artificial intelligence is leading the way. Shanghai, Chian: Deloitte Public Sector, TMT Industry & Deloitte Research.

den Boer, L.C. and Schroten, A. 2007. Traffic noise reduction in Europe. Delft, the Netherlands: CE Delft.

Denil, M. and Trappenberg, T. 2010. Overlap versus imbalance. In: *23rd Canadian Conference on Artificial Intelligence,* Ottawa, Canada. Springer, pp.220-231.

Department for Transport. 1996. Traffic Appraisal in Urban Areas. *Design Manual for Roads and Bridges (DMRB).*

Department for Transport. 2019. *National Travel Survey: 2018.* UK.

Ding, C., Wang, D., Ma, X. and Li, H. 2016. Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees. *Sustainability.* **8**(11), p1100.

Dou, H., Liu, H. and Yang, X. 2007. OD matrix estimation method of public transportation flow based on passenger boarding and alighting. *Computer and Communications.* **2**(25), pp.79-82.

Dua, D. and Graff, C. 2019. Iris Plants Database. UCI Machine Learning Repository. [Online]. Available from: http://archive.ics.uci.edu/ml

Elkosantini, S. and Darmoul, S. 2013. Intelligent public transportation systems: A review of architectures and enabling technologies. In: *2013 International Conference on Advanced Logistics and Transport,* Sousse, Tunisia. IEEE, pp.233-238.

Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R. and Schutter, B.D. 2016. Deep convolutional neural networks for detection of rail surface defects. In: *2016 International Joint Conference on Neural Networks (IJCNN),* Vancouver, BC, Canada. IEEE, pp.2584-2589.

Faroqi, H., Mesbah, M. and Kim, J. 2017. Spatial-temporal similarity correlation between public transit passengers using smart card data. *Journal of Advanced Transportation.* **2017**(4), pp.1-14.

Faroqi, H., Mesbah, M. and Kim, J. 2018. Applications of transit smart cards beyond a fare collection tool: a literature review. *Advances in Transportation Studies.* **45**, pp.107-122.

Faroqi, H., Mesbah, M. and Kim, J. 2019. Comparing Sequential with Combined Spatiotemporal Clustering of Passenger Trips in the Public Transit Network Using Smart Card Data. *Mathematical Problems in Engineering.* **2019**(2105), pp.1-16.

Firlik, B. and Tabaszewski, M. 2020. Monitoring of the technical condition of tracks based on machine learning. **234**(7), pp.702-708.

Fonzone, A., Schmöcker, J.-D. and Liu, R. 2015. A model of bus bunching under reliability-based passenger arrival patterns. *Transportation Research Part C: Emerging Technologies.* **59**, pp.164-182.

Friedman, J., Hastie, T. and Tibshirani, R. 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics.* **28**(2), pp.337-407.

Friedman, J.H. 2001. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics.* **5**(29), pp.1189-1232.

Friedman, J.H. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis.* **38**(4), pp.367-378.

Gao, S., Szugs, T. and Ahlbrink, R. 2018. Use of Combined Railway Inspection Data Sources for Characterization of Rolling Contact Fatigue. In: *12th ECNDT,* Gothenburg, Sweden.

Gao, X., Chen, Z., Tang, S., Zhang, Y. and Li, J. 2016. Adaptive weighted imbalance learning with application to abnormal activity recognition. *Neurocomputing.* **173**, pp.1927-1935.

Gazzah, S., Hechkel, A. and Amara, N.E.B. 2015. A hybrid sampling method for imbalanced data. In: *2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15),* 16-19 March 2015, Mahdia, Tunisia. IEEE, pp.1-6.

Gerland, H.E. and Sutter, K. 1999. Automatic passenger counting (APC): Infra-red motion analyzer for accurate counts in stations and rail, light-rail and bus operations. In: *1999 American Public Transportation Association Bus Conference, Proceedings* Washington, DC.

Giben, X., Patel, V.M. and Chellappa, R. 2015. Material classification and semantic segmentation of railway track images with deep convolutional neural networks. In: *2015 IEEE International Conference on Image Processing (ICIP),* 27-30 Sept. 2015, pp.621-625.

Godbole, S. and Sarawagi, S. 2004. Discriminative methods for multi-labeled classification. In: *Advances in Knowledge Discovery and Data Mining,* 2004//, Berlin, Heidelberg. Springer Berlin Heidelberg, pp.22-30.

Goldenbeld, C., Levelt, P.B.M. and Heidstra, J. 2000. Psychological perspectives on changing driver attitude and behaviour. *Recherche - Transports - Sécurité.* **67**, pp.65-81.

Gong, M., Fei, X., Wang, Z. and Qiu, Y. 2014. Sequential framework for short-term passenger flow prediction at bus stop. *Transportation Research Record: Journal of the Transportation Research Board.* **2417**, pp.58-66.

González, M.C., Hidalgo, C.A. and Barabási, A.-L. 2008. Understanding individual human mobility patterns. *Nature.* **453**(7196), pp.779-782.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. 2014. Generative adversarial nets. In: *Advances in neural information processing systems,* Montreal, QC, Canada. pp.2672-2680.

Gordon, J., Koutsopoulos, H., Wilson, N. and Attanucci, J. 2013. Automated inference of linked transit journeys in London using fare-transaction and vehicle location data. *Transportation Research Record: Journal of the Transportation Research Board.* **2343**, pp.17-24.

Gundlegård, D., Rydergren, C., Breyer, N. and Rajna, B. 2016. Travel demand estimation and network assignment based on cellular network data. *Computer Communications.* **95**, pp.29-42.

Guo, H., Li, Y., Jennifer, S., Gu, M., Huang, Y. and Gong, B. 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications.* **73**, pp.220-239.

Guo, X., Wu, J., Sun, H., Liu, R. and Gao, Z. 2016. Timetable coordination of first trains in urban railway network: A case study of Beijing. *Applied Mathematical Modelling.* **40**(17-18), pp.8048-8066.

Guo, Z., Wilson, N. and Rahbee, A. 2007. Impact of weather on transit ridership in Chicago, Illinois. *Transportation Research Record: Journal of the Transportation Research Board.* **2034**, pp.3-10.

Ha, J. and Lee, J.-S. 2016. A new under-sampling method using genetic algorithm for imbalanced data classification. In: *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication,* Danang, Viet Nam. Association for Computing Machinery, pp.1-6.

Hagenauer, J. and Helbich, M. 2017. A comparative study of machine learning classifiers for modeling travel mode choice. *Expert Systems with Applications.* **78**, pp.273-282.

Haghani, A. and Shafahi, Y. 2002. Bus maintenance systems and maintenance scheduling: model formulations and solutions. *Transportation Research Part A: Policy and Practice.* **36**(5), pp.453-482.

Hajizadeh, S., Núñez, A. and Tax, D.M.J. 2016. Semi-supervised Rail Defect Detection from Imbalanced Image Data. *IFAC-PapersOnLine.* **49**(3), pp.78-83.

Han, H., Wang, W.-Y. and Mao, B.-H. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: *International Conference on Intelligent Computing,* Berlin, Heidelberg. Springer Berlin Heidelberg, pp.878-887.

Han, Y., Liu, Z., Lyu, Y., Liu, K., Li, C. and Zhang, W. 2020. Deep learning-based visual ensemble method for high-speed railway catenary clevis fracture detection. *Neurocomputing.* **396**, pp.556-568.

Han, Y., Wang, S., Ren, Y., Wang, C., Gao, P. and Chen, G. 2019. Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks. *ISPRS International Journal of Geo-Information.* **8**(6), p243.

Hanafi, R. and Kozan, E. 2014. A hybrid constructive heuristic and simulated annealing for railway crew scheduling. *Computers & Industrial Engineering.* **70**, pp.11-19.

He, H., Bai, Y., Garcia, E. and Li, S. 2008. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence),* Hong Kong. IEEE, pp.1322-1328.

He, L., Agard, B. and Trépanier, M. 2020. A classification of public transit users with smart card data based on time series distance metrics and a hierarchical clustering method. *Transportmetrica A: Transport Science.* **16**(1), pp.56-75.

He, L. and Trépanier, M. 2015. Estimating the destination of unlinked trips in transit smart card fare data. *Transportation Research Record.* **2535**(1), pp.97-104.

He, L., Trépanier, M. and Agard, B. 2017. Comparing time series segmentation methods for the analysis of transportation patterns with smart card data. Montreal, Quebec: Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT).

He, L., Trépanier, M. and Agard, B. 2019. Sampling method qpplied to the clustering of temporal patterns of public transit smart card users. Montreal, Quebec: Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT).

Hillmer, S.C. and Tiao, G.C. 1982. An ARIMA-model-based approach to seasonal adjustment. *Journal of the American Statistical Association.* **77**(377), pp.63-70.

Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. IEEE Press.

Hochreiter, S. and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation.* **9**(8), pp.1735-1780.

Hofmann, M. and O'Mahony, M. 2005. The impact of adverse weather conditions on urban bus performance measures. In: *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE,* Vienna, Austria. IEEE, pp.84-89.

Hollander, Y. and Liu, R. 2008. Estimation of the distribution of travel times by repeated simulation. *Transportation Research Part C: Emerging Technologies.* **16**(2), pp.212-231.

Hou, Y., He, M. and Zhang, S. 2012. Origin-destination matrix estimation method based on bus smart card records. *Journal of Transport Information and Safety.* **30**(6), pp.109-114.

Hu, C. and Liu, X. 2016. Modeling track geometry degradation using support vector machine technique. In: *2016 Joint Rail Conference,* Columbia, South Carolina, USA.

Hughes-Cromwick, M. and Dickens, M. 2020. 2020 Public transportation fact book. 7th Edition ed. American Public Transportation Association.

Ibarra-Rojas, O.J., Delgado, F., Giesen, R. and Muñoz, J.C. 2015. Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological.* **77**, pp.38-75.

Jacyna, M., Wasiak, M., Lewczuk, K. and Karoń, G. 2017. Noise and environmental pollution from transport: decisive problems in developing ecologically efficient transport systems. *Journal of Vibroengineering.* **19**(7), pp.5639-5655.

Jian, C., Gao, J. and Ao, Y. 2016. A new sampling method for classifying imbalanced data based on support vector machine ensemble. *Neurocomputing.* **193**, pp.115-122.

Jiang, X., Zhang, L. and Chen, X. 2014. Short-term forecasting of high-speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China. *Transportation Research Part C: Emerging Technologies.* **44**, pp.110-127.

Jin, W., Li, P., Wu, W. and Wei, L. 2019. Short-term public transportation passenger flow forecasting method based on multi-source data and shepard interpolating prediction method. In: *Man-Machine-Environment System Engineering,* 2019//, Singapore. Springer Singapore, pp.281-294.

Jo, T. and Japkowicz, N. 2004. Class imbalances versus small disjuncts. *Association for Computing Machinery.* **6**(1), pp.40–49.

Johnson, A. 2003. Bus transit and land use: illuminating the interaction. *Journal of Public Transportation.* **6**(4), pp.21-39.

Jung, J. and Sohn, K. 2017. Deep-learning architecture to forecast destinations of bus passengers from entry-only smart-card data. *IET Intelligent Transport Systems.* **11**(6), pp.334-339.

Kang, G., Gao, S., Yu, L. and Zhang, D. 2019. Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning. *IEEE Transactions on Instrumentation and Measurement.* **68**(8), pp.2679-2690.

Karlaftis, M.G. and Vlahogianni, E.I. 2011. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies.* **19**(3), pp.387-399.

Karnberger, S. and Antoniou, C. 2020. Network–wide prediction of public transportation ridership using spatio–temporal link–level information. *Journal of Transport Geography.* **82**, p102549.

Kashfi, S.A., Bunker, J.M. and Yigitcanlar, T. 2015. Understanding the effects of complex seasonality on suburban daily transit ridership. *Journal of Transport Geography.* **46**, pp.67-80.

Khadilkar, H. 2019. A scalable reinforcement learning algorithm for scheduling railway lines. *IEEE Transactions on Intelligent Transportation Systems.* **20**(2), pp.727-736.

Khan, M.F., Asghar, S., Tamimi, M.I. and Noor, M.A. 2019. Multi-objective transport system based on regression analysis and genetic algorithm using transport data. *IEEE Access.* **7**, pp.81121-81131.

Koetse, M.J. and Rietveld, P. 2009. The impact of climate change and weather on transport: An overview of empirical findings. *Transportation Research Part D: Transport and Environment.* **14**(3), pp.205-221.

Krawczyk, B. 2016. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence.* **5**(4), pp.221-232.

Krummenacher, G., Ong, C.S., Koller, S., Kobayashi, S. and Buhmann, J.M. 2018. Wheel defect detection with machine learning. *IEEE Transactions on Intelligent Transportation Systems.* **19**(4), pp.1176-1187.

Kumar, V., Kumar, B.A. and Vanajakshi, L. 2014. Comparison of model based and machine learning approaches for bus arrival time prediction. In: *the 93rd Transportation Research Board Annual Meeting,* Washington, US. pp.14-2518.

Kurzweil, R. 2005. The Singularity Is Near: When Humans Transcend Biology. Penguin.

Kwan, S.C. and Hashim, J.H. 2016. A review on co-benefits of mass public transportation in climate change mitigation. *Sustainable Cities and Society.* **22**, pp.11-18.

Lasisi, A. and Attoh-Okine, N. 2018. Principal components analysis and track quality index: A machine learning approach. *Transportation Research Part C: Emerging Technologies.* **91**, pp.230-248.

LeCun, Y., Cortes, C. and Burges, C.J.C. 2010. MNIST Handwritten Digit Database. [Online]. Available from: http://yann.lecun.com/exdb/mnist

Legrain, A., Eluru, N. and El-Geneidy, A.M. 2015. Am stressed, must travel: The relationship between mode choice and commuting stress. *Transportation Research Part F: Traffic Psychology and Behaviour.* **34**, pp.141-151.

Li, Q. and Ren, S. 2012. A visual detection system for rail surface defects. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews).* **42**(6), pp.1531-1542.

Li, T., Sun, D., Jing, P. and Yang, K. 2018. Smart card data mining of public transport destination: a literature review. *Information.* **9**(1), p18.

Li, Y., Wang, X., Sun, S., Ma, x. and Lu, G. 2017. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transportation Research Part C: Emerging Technologies.* **77**, pp.306-328.

Lim, S.S., Vos, T., Flaxman, A.D., Danaei, G., Shibuya, K., Adair-Rohani, H., Amann, M., Anderson, H.R., Andrews, K.G., Aryee, M., Atkinson, C., Bacchus, L.J., Bahalim, A.N., Balakrishnan, K., Balmes, J., Barker-Collo, S., Baxter, A., Bell, M.L., Blore, J.D., Blyth, F., Bonner, C., Borges, G., Bourne, R., Boussinesq, M., Brauer, M., Brooks, P., Bruce, N.G., Brunekreef, B., Bryan-Hancock, C., Bucello, C., Buchbinder, R., Bull, F., Burnett, R.T., Byers, T.E., Calabria, B., Carapetis, J., Carnahan, E., Chafe, Z., Charlson, F., Chen, H., Chen, J.S., Cheng, A.T.A., Child, J.C., Cohen, A., Colson, K.E., Cowie, B.C., Darby, S., Darling, S., Davis, A., Degenhardt, L., Dentener, F., Des Jarlais, D.C., Devries, K., Dherani, M., Ding, E.L., Dorsey, E.R., Driscoll, T., Edmond, K., Ali, S.E., Engell, R.E., Erwin, P.J., Fahimi, S., Falder, G., Farzadfar, F., Ferrari, A., Finucane, M.M., Flaxman, S., Fowkes, F.G.R., Freedman, G., Freeman, M.K., Gakidou, E., Ghosh, S., Giovannucci, E., Gmel, G., Graham, K., Grainger, R.,

Grant, B., Gunnell, D., Gutierrez, H.R., Hall, W., Hoek, H.W., Hogan, A., Hosgood Iii, H.D., Hoy, D., Hu, H., Hubbell, B.J., Hutchings, S.J., Ibeanusi, S.E., Jacklyn, G.L., Jasrasaria, R., Jonas, J.B., Kan, H., Kanis, J.A., Kassebaum, N., Kawakami, N., Khang, Y.H., Khatibzadeh, S., Khoo, J.P., Kok, C., Laden, F., Lalloo, R., Lan, Q., Lathlean, T., Leasher, J.L., Leigh, J., Li, Y., Lin, J.K., Lipshultz, S.E., London, S., Lozano, R., Lu, Y., Mak, J., Malekzadeh, R., Mallinger, L., Marcenes, W., March, L., Marks, R., Martin, R., McGale, P., McGrath, J., Mehta, S., Mensah, G.A., Merriman, T.R., Micha, R., Michaud, C., Mishra, V., Hanafiah, K.M., Mokdad, A.A., Morawska, L., Mozaffarian, D., Murphy, T., Naghavi, M., Neal, B., Nelson, P.K., Nolla, J.M., Norman, R., Olives, C., Omer, S.B., Orchard, J., Osborne, R., Ostro, B., Page, A., Pandey, K.D., Parry, C.D.H., Passmore, E., Patra, J., Pearce, N., Pelizzari, P.M., Petzold, M., Phillips, M.R., Pope, D., Pope Iii, C.A., Powles, J., Rao, M., Razavi, H., Rehfuess, E.A., Rehm, J.T., Ritz, B., Rivara, F.P., Roberts, T., Robinson, C., Rodriguez-Portales, J.A., Romieu, I., Room, R., Rosenfeld, L.C., Roy, A., Rushton, L., Salomon, J.A., Sampson, U., Sanchez-Riera, L., Sanman, E., Sapkota, A., Seedat, S., Shi, P., Shield, K., Shivakoti, R., Singh, G.M., Sleet, D.A., Smith, E., Smith, K.R., Stapelberg, N.J.C., Steenland, K., Stöckl, H., Stovner, L.J., Straif, K., Straney, L., Thurston, G.D., Tran, J.H., Van Dingenen, R., Van Donkelaar, A., Veerman, J.L., Vijayakumar, L., Weintraub, R., Weissman, M.M., White, R.A., Whiteford, H., Wiersma, S.T., Wilkinson, J.D., Williams, H.C., Williams, W., Wilson, N., Woolf, A.D., Yip, P., Zielinski, J.M., Lopez, A.D., Murray, C.J.L. and Ezzati, M. 2012. A comparative risk assessment of burden of disease and injury attributable to 67 risk factors and risk factor clusters in 21 regions, 1990-2010: A systematic analysis for the Global Burden of Disease Study 2010. *The Lancet.* **380**(9859), pp.2224-2260.

Liu, C., Susilo, Y.O. and Karlström, A. 2015. Investigating the impacts of weather variability on individual's daily activity–travel patterns: A comparison between commuters and non-commuters in Sweden. *Transportation Research Part A: Policy and Practice.* **82**, pp.47-64.

Liu, R. and Sinha, S. 2007. Modelling urban bus service and passenger reliability. In: *International Symposium on Transportation Network Reliability,* July 2017, Hague.

Liu, T. 2009. EasyEnsemble and feature selection for imbalance data sets. In: *2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing,* 3-5 Aug. 2009, pp.517-520.

Liu, X., Wu, J. and Zhou, Z. 2009. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man and Cybernetics.* **39**(2), pp.539-550.

Liu, X., Zhou, Y. and Rau, A. 2019a. Smart card data-centric replication of the multi-modal public transport system in Singapore. *Journal of Transport Geography.* **76**, pp.254-264.

Liu, Y., Liu, Z. and Jia, R. 2019b. DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transportation Research Part C: Emerging Technologies.* **101**, pp.18-34.

Liu, Y., Qin, Y., Guo, J., Cai, C., Wang, Y. and Jia, L. 2018. Short-term forecasting of rail transit passenger flow based on long short-term memory neural network. In: *2018 International Conference on Intelligent Rail Transportation (ICIRT)*: IEEE, pp.1-5.

Liu, Z., Tang, D., Cai, Y., Wang, R. and Chen, F. 2017. A hybrid method based on ensemble WELM for handling multi class imbalance in cancer microarray data. *Neurocomputing.* **266**, pp.641-650.

Long, Y., Zhang, Y. and Cu, C. 2012. Identifying commuting pattern of Beijing using bus smart card data. *Acta Geographica Sinica.* **67**(10), pp.1339-1352.

López, V., Fernández, A., García, S., Palade, V. and Herrera, F. 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences.* **250**, pp.113-141.

Low, R., Cheah, L. and You, L. 2020. Commercial vehicle activity prediction with imbalanced class distribution using a hybrid sampling and gradient boosting approach. *IEEE Transactions on Intelligent Transportation Systems.* pp.1-10.

Loyola-González, O., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A. and García-Borroto, M. 2016. Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases. *Neurocomputing.* **175**, pp.935-947.

Luo, X. 2020. Smart public transport system: pattern in Hangzhou. In: *9th Intelligent Transport System Market Seminar,* Hangzhou.

Ma, F., Ren, F., Yuen, K.F., Guo, Y., Zhao, C. and Guo, D. 2019. The spatial coupling effect between urban public transport and commercial complexes: A network centrality perspective. *Sustainable Cities and Society.* **50**, p101645.

Ma, X., Liu, C., Liu, J., Chen, F. and Yu, H. 2015a. Boarding stop inference based on transitic card data. *Journal of Transportation Systems Engineering and Information Technology.* **15**(4), pp.78-84.

Ma, Z.-L., Ferreira, L., Mesbah, M. and Hojati, A.T. 2015b. Modeling bus travel time reliability with supply and demand data from automatic vehicle location and smart card systems *Transportation Research Record.* **2533**(1), pp.17-27.

Ma, Z., Xing, J., Mesbah, M. and Ferreira, L. 2014. Predicting short-term bus passenger demand using a pattern hybrid approach. *Transportation Research Part C: Emerging Technologies.* **39**, pp.148-163.

Mani, I. and Zhang, I. 2003. kNN approach to unbalanced data distributions: a case study involving information extraction. In: *Proceedings of workshop on learning from imbalanced datasets,* Washington DC. pp.1-7.

McLeod, S., Scheurer, J. and Curtis, C. 2017. Urban public transport: planning principles and emerging practice. *Journal of Planning Literature.* **32**(3), pp.223-239.

Mobileye. 2020. AI-based collision avoidance technology: Providing road-tested, advanced driver safety. [Online].

Mohammadi, R., He, Q., Ghofrani, F., Pathak, A. and Aref, A. 2019. Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects. *Transportation Research Part C: Emerging Technologies.* **102**, pp.153-172.

More, A. 2016. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint.* **arXiv**:.**06048**.

Munizaga, M., Devillaine, F., Navarrete, C. and Silva, D. 2014. Validating travel behavior estimated from smartcard data. *Transportation Research Part C: Emerging Technologies.* **44**, pp.70-79.

Munizaga, M.A. and Palma, C. 2012. Estimation of a disaggregate multimodal public transport Origin–Destination matrix from passive smartcard data from Santiago, Chile. *Transportation Research Part C: Emerging Technologies.* **24**, pp.9-18.

Nassir, N., Khani, A., Lee, S.G., Noh, H. and Hickman, M. 2011. Transit stop-level origin–destination estimation through use of transit schedule and automated data collection system. *Transportation Research Record.* **2263**(1), pp.140-150.

National Academies of Sciences Engineering and Medicine. 2009. Public transportation's role in addressing global climate change Washington, DC.

Nelson, E. and Sadowsky, N. 2019. Estimating the impact of ride-hailing App company entry on public transportation use in major US urban areas. *The B.E. Journal of Economic Analysis & Policy.* **19**(1), pp.1-35.

Nunes, A.A.N., Dias, T.G.D. and Falcão e Cunha, J. 2016. Passenger journey destination estimation from automated fare collection system data using spatial validation. *IEEE transactions on intelligent transportation systems.* **17**(1), pp.133-142.

Nurul Habib, K., El-Assi, W., Hasnine, M.S. and Lamers, J. 2017. Daily activity-travel scheduling behaviour of non-workers in the National Capital Region (NCR) of Canada. *Transportation Research Part A: Policy and Practice.* **97**, pp.1-16.

Obara, M., Kashiyama, T. and Sekimoto, Y. 2018. Deep reinforcement learning approach for train rescheduling utilizing graph theory. In: *2018 IEEE International Conference on Big Data (Big Data),* 10-13 Dec. 2018, Seattle, WA, USA. IEEE, pp.4525-4533.

Omrani, H. 2015. Predicting travel mode of individuals by machine learning. In: *Transportation Research Procedia*, pp.840-849.

Oransirikul, T., Nishide, R., Piumarta, I. and Takada, H. 2014. Measuring bus passenger load by monitoring Wi-Fi transmissions from mobile devices *Procedia Technology.* **18**, pp.120-125.

Park, S.H. and Ha, Y.G. 2014. Large imbalance data classification based on MapReduce for traffic accident prediction. In: *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing,* 2-4 July 2014, Birmingham, UK. IEEE, pp.45-49.

Parsa, A.B., Taghipour, H., Derrible, S. and Mohammadian, A. 2019. Real-time accident detection: coping with imbalanced data. *Accident Analysis & Prevention.* **129**, pp.202-210.

Paulley, N., Balcombe, R., Mackett, R., Titheridge, H., Preston, J., Wardman, M., Shires, J. and White, P. 2006. The demand for public transport: The effects of fares, quality of service, income and car ownership. *Transport Policy.* **13**(4), pp.295-306.

Pei, M., Lin, P., Liu, R. and Ma, Y. 2019. Flexible transit routing model considering passengers' willingness to pay. *IET Intelligent Transport Systems.* **13**(5), pp.841-850.

Pelletier, M.-P., Trépanier, M. and Morency, C. 2011. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies.* **19**(4), pp.557-568.

Petersen, N., Rodrigues, F. and Pereira, F. 2019. Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Systems with Applications.* **120**, pp.426-435.

Powers, D.M. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies.* **2**(1), pp.37-63.

Pucher, J. 2005. Public transport in Seoul: Meeting the burgeoning travel demands of a megacity. *Public Transport International.* **54**(3), pp.54-61.

Rahaman, M.S., Hamilton, M. and Salim, F.D. 2017. Predicting imbalanced taxi and passenger queue contexts in airport. In: *Pacific Asia Conference on Information Systems (PACIS),* Malaysia. Association for Information Systems, p.172.

Ramentol, E., Caballero, Y., Bello, R. and Herrera, F. 2012. SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowledge and Information Systems.* **33**(2), pp.245-265.

Ranjitkar, P., Tey, L.-S., Chakravorty, E. and Hurley, K.L. 2019. Bus arrival time modeling based on auckland data. *Transportation Research Record.* **2673**(6), pp.1-9.

Rasouli, S. and Timmermans, H.J.P. 2014. Using ensembles of decision trees to predict transport mode choice decisions: effects on predictive success and uncertainty estimates. *EJTIR.* **14**(4), pp.412-424.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A. and Bernstein, M. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision.* **115**(3), pp.211-252.

Sabir, M. 2011. *Weather and travel behaviour*. PhD thesis, Vrije Universiteit Amsterdam.

Sakai, K., Liu, R., Kusakabe, T. and Asakura, Y. 2017. Pareto-improving social optimal pricing schemes based on bottleneck permits for managing congestion at a merging section. *International Journal of Sustainable Transportation.* **11**(10), pp.737-748.

Salsingikar, S. and Rangaraj, N. 2020. Reinforcement learning for train movement planning at railway stations. In: *Adaptive and Learning Agents Workshop,* Auckland.

Saneinejad, S., Roorda, M.J. and Kennedy, C. 2012. Modelling the impact of weather conditions on active transportation travel behaviour. *Transportation Research Part D: Transport and Environment.* **17**(2), pp.129-137.

Schapire, R.E. and Singer, Y. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning.* **39**(2-3), pp.135-168.

Schmöcker, J.-D., Sun, W., Fonzone, A. and Liu, R. 2016. Bus bunching along a corridor served by two lines. *Transportation Research Part B: Methodological.* **93**, pp.300-317.

Schneider, C.M., Belik, V., Couronné, T., Smoreda, Z. and González, M.C. 2013. Unravelling daily human mobility motifs. *Journal of The Royal Society Interface.* **10**(84), p20130246.

Shalit, N., Fire, M. and Elia, E.B. 2020. Imputing missing boarding stations with machine learning methods. [Online]. 2003.05285. **arXiv preprint**, p2003.05285.

Sharifirad, S., Nazari, A. and Ghatee, M. 2014. An enhanced smote algorithm using entropy and clustering for imbalanced accident data In: *2nd National Conference on Applied Research in Computer Science and Information Technology,* Tehran, Iran. pp.1-6.

Sierpiński, G. 2016. Intelligent Transport Systems and Travel Behaviour. Katowice, Poland: Springer.

Singhal, A., Kamga, C. and Yazici, A. 2014. Impact of weather on urban transit ridership. *Transportation Research Part A: Policy and Practice.* **69**, pp.379-391.

Song, J., Huang, X., Qin, S. and Song, Q. 2016. A bi-directional sampling based on K-means method for imbalance text classification. In: *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS),* 26-29 June 2016, Okayama, Japan. IEEE, pp.1-5.

Sorratini, J., Liu, R. and Sinha, S. 2008. Assessing bus transport reliability using micro-simulation. *Transportation Planning and Technology.* **31**(3), pp.303-324.

Sorrell, S. 2019. What's in it for citizans? Juniper Research.

Stehman, S.V. 1997. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment.* **62**(1), pp.77-89.

Stover, V.W. and McCormack, E.D. 2012. The impact of weather on bus ridership in Pierce County, Washington. *Journal of Public Transportation.* **15**(1), p6.

Sun, Y., Jiang, G., Lam, S.-K., Chen, S. and He, P. 2019. Bus travel speed prediction using attention network of heterogeneous correlation features. In: *the 2019 SIAM International Conference on Data Mining,* Calgary, Alberta, Canada. pp.73-81.

Sun, Y., Shi, J. and Schonfeld, P.M. 2016. Identifying passenger flow characteristics and evaluating travel time reliability by visualizing AFC data: a case study of Shanghai Metro. *Public Transport.* **8**(3), pp.341-363.

Svozil, D., Kvasnicka, V. and Pospichal, J.í. 1997. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems.* **39**(1), pp.43-62.

Tang, T., Fonzone, A., Liu, R. and Choudhury, C.F. 2020a. Predicting passengers' boarding stops under different weather conditions using machine learning technique. In: *99th Transportation Research Board Annual Meeting,* Washinton D.C.

Tang, T., Liu, R. and Choudhury, C. 2020b. Incorporating weather conditions and travel history in estimating the alighting bus stops from smart card data. *Sustainable Cities and Society.* **53**, p101927.

Tomek, I. 1976. Experiment with the edited nearest-nieghbor rule. *IEEE Transactions on Systems, Man and Cybernetics.* **SMC-6**(6), pp.448-452.

Tong, H.Y. 2019. Development of a driving cycle for a supercapacitor electric bus route in Hong Kong. *Sustainable Cities and Society.* **48**, p101588.

Toqué, F., Côme, E., El Mahrsi, M.K. and Oukhellou, L. 2016. Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC),* Rio de Janeiro, Brazil. IEEE, pp.1071-1076.

Trépanier, M. and Chapleau, R. 2006. Destination estimation from public transport smartcard data. *IFAC Proceedings Volumes.* **39**(3), pp.393-398.

Trépanier, M., Tranchant, N. and Chapleau, R. 2007. Individual trip destination estimation in a transit smart card automated fare collection system. *Journal of Intelligent Transportation Systems.* **11**(1), pp.1-14.

Tsoumakas, G. and Katakis, I. 2007. Multi-label classification: an overview. *International Journal of Data Warehousing and Mining (IJDWM).* **3**(3), pp.1-13.

Turok, I. 2014. Cities as frivers of development. In: Kayizzi-Mugerwa, S., et al. eds. *Urbanization and Socio-Economic Development in Africa: Challenges and Opportunities.* Routledge, pp.14-41.

United Nations. 2015. Transforming our world: The 2030 agenda for sustainable development. *General Assembley 70 session.*

United Nations. 2016. *The World Cities Report 2016.* A/CONF.226/PC.1/5. New York.

Varagouli, E.G., Simos, T.E. and Xeidakis, G.S. 2005. Fitting a multiple regression line to travel demand forecasting: The case of the prefecture of Xanthi, Northern Greece. *Mathematical and Computer Modelling.* **42**(7), pp.817-836.

Wang, W., Attanucci, J.P. and Wilson, N.H. 2011. Bus passenger origin-destination estimation and related analyses using automated data collection systems. *Journal of Public Transportation.* **14**(4), p7.

Washington, S.P., Karlaftis, M.G. and Mannering, F. 2010. Statistical and Econometric Methods for Transportation Data Analysis. Chapman and Hall/CRC.

Wei, M., Liu, Y., Sigler, T., Liu, X. and Corcoran, J. 2019. The influence of weather conditions on adult transit ridership in the sub-tropics. *Transportation Research Part A: Policy and Practice.* **125**, pp.106-118.

Wei, M., Sun, B. and Jin, W. 2012. Model and algorithm of regional bus scheduling with grey travel time. *Journal of Transportation Systems Engineering and Information Technology.* **12**(6), pp.106-112.

Wei, Y. and Chen, M.-C. 2012. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. *Transportation Research Part C: Emerging Technologies.* **21**(1), pp.148-162.

Weinberger, K., Dasgupta, A., Langford, J., Smola, A. and Attenberg, J. 2009. Feature hashing for large scale multitask learning. In: *Proceedings of the 26th annual international conference on machine learning,* Montreal Canada. pp.1113-1120.

Wilson, D.L. 1972. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics.* (3), pp.408-421.

Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J. 2016. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.

Wu, W., Jiang, S., Liu, R., Jin, W. and Ma, C. 2020a. Economic development, demographic characteristics, road network and traffic accidents in Zhongshan, China: Gradient boosting decision tree model. *Transportmetica A: Transport Science.* **16**(3), pp.359-387.

Wu, W., Li, P., Liu, R., Jin, W., Yao, B., Xie, Y. and Ma, C. 2020b. Predicting peak load of bus routes with supply optimization and scaled Shepard interpolation: A newsvendor model. *Transportation Research Part E: Logistics and Transportation Review.* **142**, p102041.

Wu, W., Liu, R. and Jin, W. 2016. Designing robust schedule coordination scheme for transit networks with safety control margins. *Transportation Research Part B: Methodological.* **93**, pp.495-519.

Wu, W., Liu, R. and Jin, W. 2017. Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour. *Transportation Research Part B: Methodological.* **104**, pp.175-197.

Wu, W., Liu, R., Jin, W. and Ma, C. 2019. Stochastic bus schedule coordination considering demand assignment and rerouting of passengers. *Transportation Research Part B: Methodological.* **121**, pp.275-303.

Xie, B., An, Z., Zheng, Y. and Li, Z. 2019a. Incorporating transportation safety into land use planning: Pre-assessment of land use conversion effects on severe crashes in urban China. *Applied geography.* **103**, pp.1-11.

Xie, B., Jiao, J., An, Z., Zheng, Y. and Li, Z. 2019b. Deciphering the stroke–built environment nexus in transitional cities: Conceptual framework, empirical

evidence, and implications for proactive planning intervention. *Cities.* **94**, pp.116-128.

Xing, J., Liu, Z., Wu, C. and Chen, S. 2019. Traffic volume estimation in multimodal urban networks using cell phone location data. *IEEE Intelligent Transportation Systems Magazine.* **11**(3), pp.93-104.

Xu, S., Liu, R., Liu, T. and Huang, H. 2018. Pareto-improving policies for an idealized two-zone city served by two congestible modes. *Transportation Research Part B: Methodological.* **117**, pp.876-891.

Xue, R., Sun, D. and Chen, S. 2015. Short-Term bus passenger demand prediction based on time series model and interactive multiple model approach. *Discrete Dynamics in Nature and Society.* **2015**, pp.1-11.

Yan, F., Yang, C. and Ukkusuri, S.V. 2019. Alighting stop determination using two-step algorithms in bus transit systems. *Transportmetrica A: Transport Science.* **15**(2), pp.1522-1542.

Yang, X. and Liu, L. 2016. Short-term passenger flow forecasting on bus station based on affinity propagation and support vector machine. *Journal of Wuhan University of Technology (Transportation Science & Engineering).* **1**, p8.

Yang, Y., Heppenstall, A., Turner, A. and Comber, A. 2019a. A spatiotemporal and graph-based analysis of dockless bike sharing patterns to understand urban flows over the last mile. *Computers, Environment and Urban Systems.* **77**, p101361.

Yang, Y., Heppenstall, A., Turner, A. and Comber, A. 2019b. Who, where, why and when? Using smart card and social media data to understand urban mobility. *ISPRS International Journal of Geo-Information.* **8**(6), p271.

Yang, Y., Heppenstall, A., Turner, A. and Comber, A. 2020. Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems. *Computers, Environment and Urban Systems.* **83**, p101521.

Yang, Z.-w., Zhao, Q., Zhao, S.-c., Jin, L. and Mao, Y. 2009. Passenger flow volume forecasting method based on public transitintelligent card (IC) survey data. *Transport Standardization.* **9**, pp.115-119.

Yao, E., Liu, T., Lu, T. and Yang, Y. 2020. Optimization of electric vehicle scheduling with multiple vehicle types in public transport. *Sustainable Cities and Society.* **52**, p101862.

Yen, S.J. and Lee, Y.S. 2009. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications.* **36**(3, Part 1), pp.5718-5727.

Yigitcanlar, T., Kamruzzaman, M., Buys, L., Ioppolo, G., Sabatini-Marques, J., da Costa, E.M. and Yun, J.J. 2018. Understanding 'smart cities': Intertwining development drivers with desired outcomes in a multidimensional framework. *Cities.* **81**, pp.145-160.

Yin, H., Wu, J., Sun, H., Kang, L. and Liu, R. 2019. Optimizing last trains timetable in the urban rail network: social welfare and synchronization. *Transportmetrica B: Transport Dynamics.* **7**(1), pp.473-497.

Yin, J., Yu, D., Yin, Z., Liu, M. and He, Q. 2016. Evaluating the impact and risk of pluvial flash flood on intra-urban road network: A case study in the city center of Shanghai, China. *Journal of Hydrology.* **537**, pp.138-145.

Yu, B., Lam, W.H. and Tam, M.L. 2011. Bus arrival time prediction at bus stop with multiple routes. *Transportation Research Part C: Emerging Technologies.* **19**(6), pp.1157-1170.

Zannat, K.E. and Choudhury, C.F. 2019. Emerging big data sources for public transport planning: A systematic eeview on current state of art and future research directions. *Journal of the Indian Institute of Science.* **99**, pp.601–619.

Zeng, W., Miwa, T. and Morikawa, T. 2015. Exploring trip fuel consumption by machine learning from GPS and CAN bus data. *Journal of the Eastern Asia Society for Transportation Studies.* **11**, pp.906-921.

Zhang, G.P. 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing.* **50**, pp.159-175.

Zhang, J., Chen, F. and Shen, Q. 2019. Cluster-based LSTM network for short-term passenger flow forecasting in urban rail transit. *IEEE Access.* **7**, pp.147653-147671.

Zhang, X., Zhang, Q., Sun, T., Zou, Y. and Chen, H. 2018. Evaluation of urban public transport priority performance based on the improved TOPSIS method: A case study of Wuhan. *Sustainable Cities and Society.* **43**, pp.357-365.

Zhang, Y., Zhang, L. and Wang, Y. 2010. Cluster-based majority under-sampling approaches for class imbalance learning. In: *2nd IEEE International Conference on Information and Financial Engineering,* IEEE. pp.400-404.

Zhao, J., Rahbee, A. and Wilson, N.H. 2007. Estimating a rail passenger trip origin - destination matrix using automatic data collection systems. *Computer - Aided Civil and Infrastructure Engineering.* **22**(5), pp.376-387.

Zhou, C., Dai, P. and Li, R. 2013. The passenger demand prediction model on bus networks. In: *2013 IEEE 13th International Conference on Data Mining Workshops,* 7-10 Dec. 2013, IEEE. pp.1069-1076.

Zhou, L. 2013. Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods. *Knowledge-Based Systems.* **41**, pp.16-25.

Zhou, M., Wang, D., Li, Q., Yue, Y., Tu, W. and Cao, R. 2017. Impacts of weather on public transport ridership: Results from mining data from different sources. *Transportation Research Part C: Emerging Technologies.* **75**, pp.17-29.

Zhou, X., Wang, M. and Li, D. 2019. Bike-sharing or taxi? Modeling the choices of travel mode in Chicago using machine learning. *Journal of Transport Geography.* **79**, p102479.

Zhou, Z. 2016. Machine Learning. Tsinghua University Press.

Appendix A

One–way analysis of variance of trip types vs feature ‘weather event’

Table A.1 Results of the one–way analysis of variance of trip types vs feature ‘weather event’.

Total number of trips in categories	Trip types	Homogeneity of test	p–value
Trips in a chain	X1	0.737	0.023
Segments in transfer journey excluding last one	X2	0.338	0.004
Last segment in the transfer journey	X3	0.403	0.112
Other trips	X4	0.483	0.068
Transfer trips	X2+X3	0.559	0.020
Trips chains and transfer trips	X1+X2+X3	0.889	0.018
All the trips	X1+X2+X3+X4	0.588	0.032
Trips used in the paper	X1+X2	0.777	0.017

Appendix B

Features selected for the boarding behaviour prediction in Chapter 3

Table B.1 Investigated domain of features employed in machine learning models.

Feature types: C- Categorical; Nom-Nominal; Num-Numerical.

Feature domains	Features	Dimensions		Feature types	Explanation
		Stage 1 & 2	Stage 3		
Boarding time	Season	4		C	Spring; summer; autumn; winter.
	Days in week	7		C	Mon., Tues., Wed., Thurs., Fri., Sat., Sun.
	Holiday	2		C	Holidays and working days.
	Time slot	1		Num	One-hour time slot from 6 am on a given to 1 am on the next day

Feature domains	**Features**	**Dimensions**		**Feature types**	**Explanation**
		Stage 1 & 2	Stage 3		
Weather condition	Temperature	1		Num	The average temperature during the time slot
	Precipitation	1		Num	Total precipitation during the time slot
	Humidity	1		Num	Average relative humidity during the time slot
	Visibility	1		Num	Minimum visibility during the time slot
	Wind speed	1		Num	Maximum instantaneous wind speed during the time slot
	Weather events	6		C	Clear, Cloudy, Fog, Overcast, Rain, Unknown
	AQI	1		Num	Air quality index
Travel history	Card ID	17		Nom	Unique ID to identify the card users
	Bus lines/stops used on day-1	7	10	C	Labels of bus lines/stops used by the passengers on the previous day, i.e. day-1
	Bus lines used on day-7	7	10	C	Labels of bus lines/stops used by the passengers on the same day last week, i.e. day-7
	Bus lines/stops used from day-7 to day-1	7	10	C	Labels of bus lines/stops used by the passengers on all previous seven days, i.e. from day-7 to day-1
	Bus lines/stops used in the same hour on day-1	7	10	C	Labels of bus lines/stops used by the passengers in the same hour on the previous day
	Bus lines/stops used in the same hour on day-7	7	10	C	Labels of bus lines/stops used by the passengers in the same hour on the same day last week
	Bus lines/stops used in the same hour from day-7 to day-1	7	10	C	Labels of bus lines/stops used by the passengers in the same hour on all previous seven days

Feature domains	Features	Dimensions		Feature types	Explanation
		Stage 1 & 2	Stage 3		
Travel history	Most used bus line/stop on day-1	7	10	C	Label of the most used bus line/stop by the passengers on the previous day
	Most used bus line/stop on day-7	7	10	C	Label of the most used bus line/stop by the passengers on the same day last week
	Most used bus line/stop from day-7 to day-1	7	10	C	Label of the most used bus line/stop by the passengers on all previous seven days
	Most used bus line/stop in the same hour on day-1	7	10	C	Label of the most used bus line/stop by the passengers in the same hour on the previous day
	Most used bus line/stop in the same hour on day-7	7	10	C	Label of the most used bus line/stop by the passengers in the same hour on the same day last week
	Most used bus line/stop in the same hour from day-7 to day-1	7	10	C	Label of the most used bus line/stop by the passengers in the same hour on all previous seven days
	Total number of trips on day-1	1		Num	Number of trips made by the passengers on the previous day
	Total number of trips on day-7	1		Num	Number of trips made by the passengers on the same day last week
	Total number of trips from day-7 to day-1	1		Num	Number of trips made by the passengers on all previous seven days
	Total number of trips in the same hour on day-1	1		Num	Number of trips made by the passengers in the same hour on the previous day
	Total number of trips in the same hour on day-7	1		Num	Number of trips made by the passengers in the same hour on the same day last week
	Total number of trips in the same hour from day-7 to day-1	1		Num	Number of trips made by the passengers in the same hour on all previous seven days

Appendix C

Model configurations of Deep-GAN resampling model and DNN-based predictive model in Chapter 4

Table C.1 The configurations of the generator and discriminator in Deep-GAN model for E1:20 to E1:1.

Networks	No.	Name of Layer	Configurations
Generator	1	Input layer	input_shape = (batch_size, 8); output_shape = (batch_size, 8)
	2	Dense layer	neurons = 8; input_shape = (batch_size, 8); output_shape = (batch_size, 8); activation v 'relu'
	3	Dense layer	neurons = 16; input_shape = (batch_size, 8); output_shape = (batch_size, 16); activation = 'relu'
	4	Dense layer	neurons = 32; input_shape = (batch_size, 16); output_shape = (batch_size, 32); activation = 'relu'
	5	Dense layer	neurons = 36; input_shape = (batch_size, 32); output_shape = (batch_size, 36); activation = 'relu'
	6	Dense layer	neurons = 49; input_shape = (batch_size, 32); output_shape = (batch_size, 49); batch_normalization = Yes; activation = 'tanh'

Networks	No.	Name of Layer	Configurations
Discriminator	1	Input layer	input_shape = (batch_size, 49); output_shape = (batch_size, 49)
	2	Dense layer	neurons = 36; input_shape = (batch_size, 49); output_shape = (batch_size, 36); activation = 'leakyrelu'; leaky_relu_alpha = 0.2
	3	Dense layer	neurons = 25; input_shape = (batch_size, 36); output_shape = (batch_size, 25); activation = 'leakyrelu'; leaky_relu_alpha = 0.2
	4	Dense layer	neurons = 16; input_shape = (batch_size, 25); output_shape = (batch_size, 16); activation = 'leakyrelu'; leaky_relu_alpha = 0.2
	5	Dense layer	neurons = 1; input_shape = (batch_size, 16); output_shape = (batch_size, 1); activation = 'sigmoid'

Table C.2 The configurations of the DNN-based predictive model.

No.	Name of Layer	Configurations
1	Input layer	input_shape = (batch_size, 49); output_shape = (batch_size, 49)
2	Dense layer	neurons = 36; input_shape = (batch_size, 49); output_shape = (batch_size, 36); activation = 'relu'
3	Dense layer	neurons = 32; input_shape = (batch_size, 36); output_shape = (batch_size, 32); activation = 'relu'
4	Dense layer	neurons = 25; input_shape = (batch_size, 32); output_shape = (batch_size, 25); activation = 'relu'
5	Dense layer	neurons = 25; input_shape = (batch_size, 25); output_shape = (batch_size, 16); activation = 'relu'
6	Dense layer	neurons = 1; input_shape = (batch_size, 16); output_shape = (batch_size, 1); activation = 'sigmoid'