Article:
Tang, T, Fonzone, A, Liu, R orcid.org/0000-0003-0627-3184 et al. (1 more author) (2021) Multi-stage deep learning approaches to predict boarding behaviour of bus passengers. Sustainable Cities and Society, 73. 103111. ISSN 2210-6707

https://doi.org/10.1016/j.scs.2021.103111

Please cite the paper as:
Tang, T., Fonzone, A., Liu, R. and Choudhury, C. (2021) Multi-stage deep learning approaches to predicting travel patterns of bus passengers. **Sustainable Cities and Society.** https://doi.org/10.1016/j.scs.2021.103111

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

Tianli Tang [a], Achille Fonzone[b], Ronghui Liu [a*], Charisma Choudhury [a],

[a] *Institute for Transport Studies, University of Leeds, Leeds LS2 9JT, United Kingdom.*

[b] *School of Engineering and The Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, United Kingdom.*

*Corresponding author at: Institute for Transport Studies, University of Leeds, Leeds LS2 9JT, United Kingdom. Tel: +44 0113 343 5338. E-mail address: R.Liu@its.leeds.ac.uk (Ronghui Liu)

Abstract: Smart card data has emerged in recent years and provide a comprehensive, and cheap source of information for planning and managing public transport systems. This paper presents a multi-stage machine learning framework to predict passengers' boarding stops using smart card data. The framework addresses the challenges arising from the imbalanced nature of the data (e.g. many non-travelling data) and the 'many-class' issues (e.g. many possible boarding stops) by decomposing the prediction of hourly ridership into three stages: whether to travel or not in that one-hour time slot, which bus line to use, and at which stop to board. 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) are implemented. The proposed approach is applied to a real-life bus network. We show that the data imbalance has a profound impact on the accuracy of prediction at individual level. At aggregated level, FCN is able to accurately predict the rideship at individual stops, it

is poor at capturing the temporal distribution of ridership. RNN and LSTM are able to measure the temporal distribution but lack the ability to capture the spatial distribution through bus lines.

Keywords: Deep learning; Smart public transport; Travel pattern; Smart card data; Neural network.

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, 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 (DfT, 2019) and 45% in Beijing (BIT, 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 (DfT, 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 a 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, 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) and 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.

Viewing over a long period of time, it is possible to observe certain regularity in travel behaviour. For example, commuters also travel from home to the office in the morning and from office to home in the evening. However, there is inter and intra passenger variability in behaviour given the level-of-service is not static. 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 the reliability and level-of-service. So, when the travelling scenario changes, passengers are likely to change their travel choice. For example, adverse weather conditions may lead to irregular headway and longer travel time, or may even disrupt the public transport services (Ma et al., 2015, Koetse and Rietveld, 2009), which in turn indirectly affects passengers' travel choice. An increasing number of studies have highlighted the varying impacts of weather conditions on public transport (Böcker et al., 2013), alongside level-of-service attributes such as travel time, cost, waiting time, the number of transfers and other network characteristics. Wei et al. (2019) report that rain and snow both have a clear negative impact on bus ridership in Brisbane. There is, however, different findings on the impact of temperature on bus ridership. Stover and McCormack (2012) show that temperature

has a positive impact on bus ridership in Washington, USA, while Zhou et al. (2017) find a negative impact of high temperature on bus ridership 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 in this paper refers to whether to take a bus trip and which bus line and bus stop to use. As stated above, regular boarding behaviour can be tracked by their travel history, while changes 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 obtain the hour ridership on the stop-, line- and network-level. Unlike deriving existing bus ridership directly from the smart card data, the machine learning models developed can be used to predict future ridership. Public transport planning made based on the future situation and changes of ridership will be more reliable and tractable. Not only can appropriate planning meet the passengers' travel demand, but it also provides comfortable and efficient bus services and improves the level-of-service. It is the most important way to attract more travellers using buses and to work towards more sustainable cities.

2 Related works

The development of an automatic data collection system offers an opportunity to understand travel demand and to better plan the public transport system (Zhang et al., 2018). For example, smart card data (Bordagaray et al., 2016) and GPS data (Yang et al., 2019a,2020) have been used to capture bike-sharing travel behaviour and to replicate the public transport system (Liu et al., 2019a). There is 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 the metro in Shenzhen. Oransirikul et al. (2014) 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 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) propose 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 regression equations to forecast the number of boarding and on-board passengers, 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 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. A key common feature of these existing studies is that they rely on historical passenger flow on bus stops only and 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), train arrival times and delays (Yu et al., 2011; Corman and Kecman, 2018), and bus 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-crowed stops under special events scenarios. Liu et al. (2019b) achieve the same objective through a long 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. (2020) incorporate weather conditions and travel history of travellers in a gradient boosting decision tree model to estimate the alighting stop for every smart card trip, and rank the relative importance of the features.

The existing literature tends to cluster bus passengers to decrease the number of objects in the analysis and analyse the group travel behaviour. 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: the S-T clusters the spatial matrix firstly and then temporal matrix for each spatial group; the T-S clusters the temporal matrix firstly and then spatial matrix for each temporal group; and the ST combines both spatial and temporal similarity matrices into one matrix. The study concludes that the S-T method is better used in the cases where spatial similarity is more important, and the 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, 2019) propose to combine cross correlation distance and hierarchical clustering to segment the time series of passengers' travel pattern, and incorporate a sampling method to classify the temporal pattern of bus passengers. Later, He et al. (2020) use time series distance metrics and a hierarchical clustering method to classify the public transport users. Such grouping methods generalise the common characteristics of passenger behaviour and ignore the differences between individuals.

Summing up, the existing research focuses mainly on the average passenger demand and ignores the individual differences in passenger travel behaviour. The existing studies tend to estimate passenger demand over certain time periods of a day (e.g. morning and evening peak, and inter-peak periods), thus do not provide a continuous time-dependent load-profile of

the bus demand, which alongside max load are important factors to consider in planning a holistic bus systems Ceder (2007). This study proposes a bottom-up approach, by directly predicting individual passengers' boarding behaviour and takes account of the individual's travel history and the weather factors in the prediction. From the prediction of individual boarding behaviour, this study contributes to predicting the boarding demand as a continuous variable throughout the day and at different levels (at bus-stop, by bus line, and at network level), which can be used to better plan and manage public transport services and operations.

3 A machine learning framework for boarding stop prediction

3.1 Problem statement

In this study, we take the boarding behaviour of a passenger during a one-hour time slot as the prediction 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 most possible class.

There are three data challenges in our study 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 at 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 did not travel in this period and/or did not board at that particular stop. Boarding observations at 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 lines and each bus line has multiple bus stops, so there are many classes on the label. Such a many-class problem makes the classification more difficult, 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.

3.2 A three-stage framework for predicting boarding stops

Figure 1 illustrates our proposed 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).

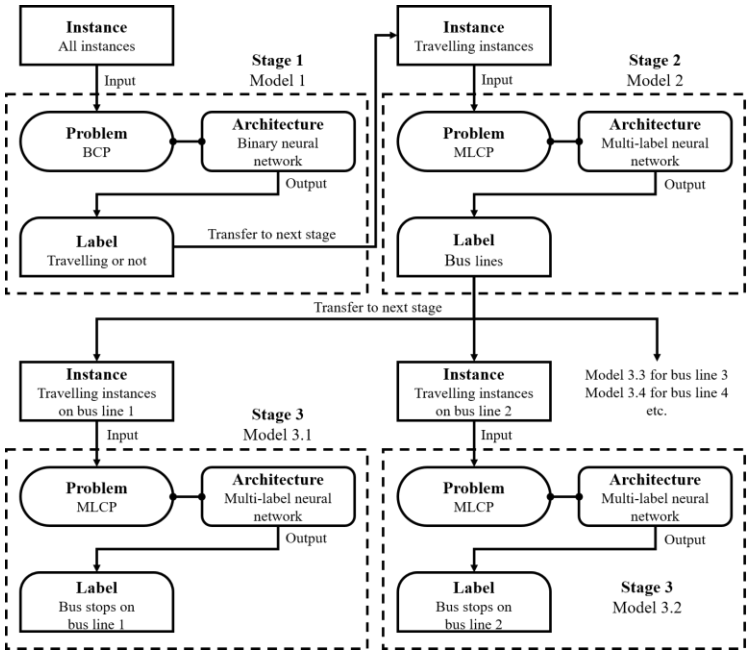

Figure 1 The three-stage prediction framework, and the processes and models involved in each stage of the prediction.

Stage 1 predicts whether a passenger makes a trip at a given one-hour slot. 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 and predicts the bus line taken for each travelling instance. There are multiple bus lines for each instance, and

a passenger may make transfers in the same one-hour time slot. Therefore Stage 2 is a multi-class multi-label classification problem (MLCP) (Tsoumakas and Katakis, 2007). The labels are bus lines passenger used and the classes are all the bus lines in the network.

Stage 3 works on our ultimate target, predicting the boarding stop for each bus trip. Here, we build up a prediction model for each bus line (noted as Model 3.1 and Model 3.2 for bus line 1 and 2 in Figure 1). As the number of classes in each model varies according to the number of stops along the line, Stage 3 also belongs to MLCP. The labels are bus stops passenger used on a bus line and the classes are all bus stops along the line.

3.3 Architectures of neural network

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

3.3.1 Fully connected neural networks (FCN)

Figure 2 illustrates the classic FCN architecture (Svozil et al., 1997). It consists of three layers: input, output, and some hidden layers. Each layer has a 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) to the output nodes. In the context of this study, 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.

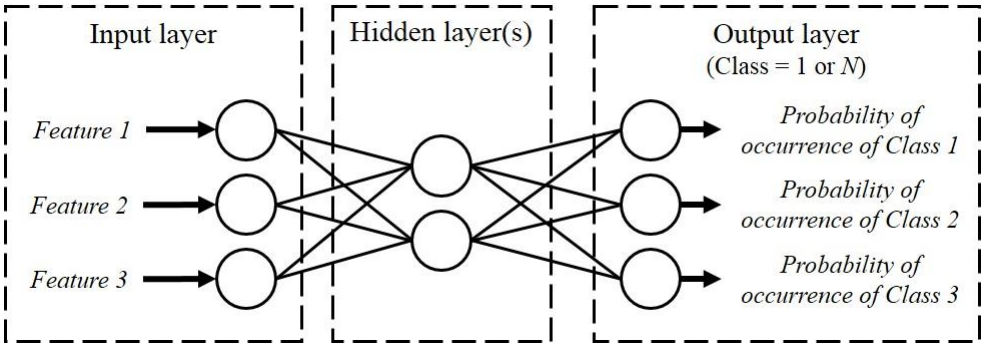

Figure 2 An example architecture of FCN.

3.3.2 Recurrent neural networks (RNN)

RNN architecture is good with sequential (e.g. temporal) data (Connor et al., 1994). RNN also consists of an input, hidden and output layer, and each layer contains one or more nodes (Figure 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. The input of RNN requires a sequence of instance:

$$\{x_1, x_2, \cdots, x_t, \cdots, x_T\} \tag{1}$$

where x_t denotes the input instance at 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 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:

$$h_t = \phi\left(Ux_t + Wh_{t-1}\right) \tag{2}$$

where h_t and $h_{t\text{-}1}$ denote the current and previous hidden state at the position t and t-1in 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:

$$z_t = o\left(Vh_t\right) \tag{3}$$

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.

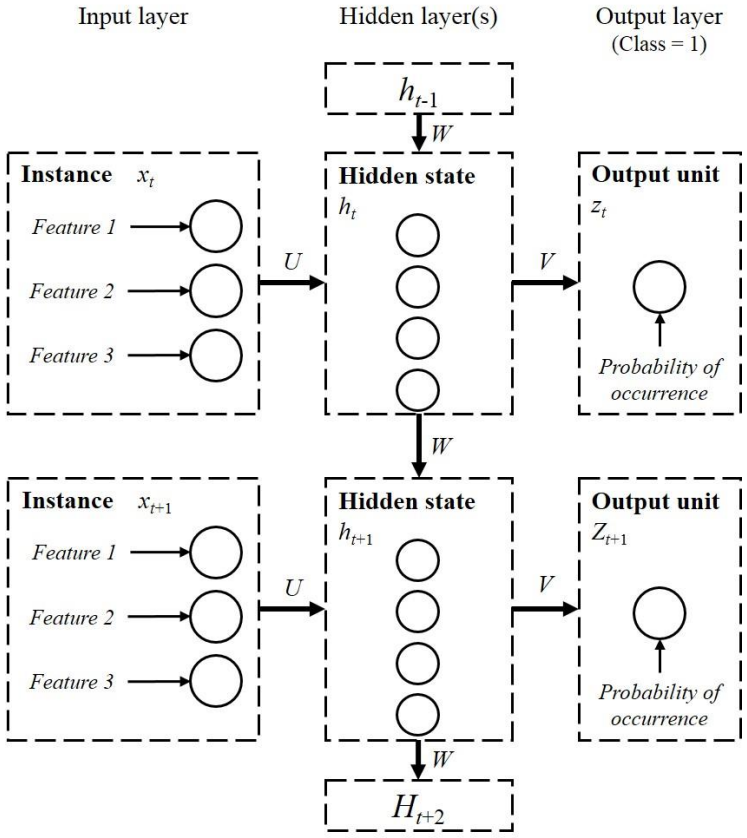

Figure 3 The example architectures of RNN.

3.3.3 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 (shown in Figure 4) is similar to that of RNN but utilises a new concept, called 'cell state'. The cell state, illustrated as the horizontal line going through the top of Figure 4, is 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.

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right]\right) \tag{4}$$

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; $\sigma(\cdot)$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.

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right]\right) \tag{5}$$

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.

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right]\right) \tag{6}$$

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:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{7}$$

$$\tilde{C}_t = \mu\left(W_C \cdot \left[h_{t-1}, x_t\right]\right) \tag{8}$$

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.

$$h_t = o_t \cdot \varphi\left(C_t\right) \tag{9}$$

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

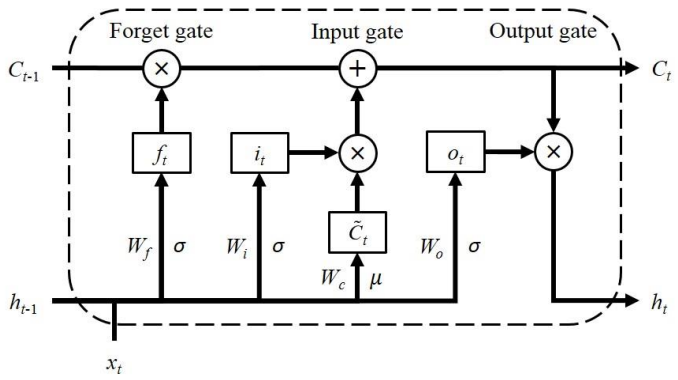

Figure 4 The example architectures of LSTM.

3.4 Feature selection

We define model features into three domains: boarding time, weather condition and travel history; they are listed in Table 1. Features concerning boarding time are clearly relevant to the boarding stop prediction problem. The weather features imply the impacts of different weather conditions on passenger behaviour. The travel history 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 numerical features are normalised.

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

Feature domains	Features	Dimensions		Feature types	Explanation
		Stage 1 & 2	Stage 3		
Boarding time	Season	4		Categorical	Spring; summer; autumn; winter.
	Days in week	7		Categorical	Mon., Tues., Wed., Thurs., Fri., Sat., Sun.
	Holiday	2		Categorical	Holidays and working days.
	Time slot	1		Numerical	One-hour time slot from 6 am on a given to 1 am on the next day
Weather condition	Temperature	1		Numerical	The average temperature during the time slot
	Precipitation	1		Numerical	Total precipitation during the time slot

	Humidity	1		Numerical	Average relative humidity during the time slot
	Visibility	1		Numerical	Minimum visibility during the time slot
	Wind speed	1		Numerical	Maximum instantaneous wind speed during the time slot
	Weather events	6		Categorical	Clear, Cloudy, Fog, Overcast, Rain, Unknown
	AQI	1		Numerical	Air quality index
Travel history	Card ID	17		Nominal	Unique ID to identify the card users
	Bus lines/stops used on day-1	7	10	Categorical	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	Categorical	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	Categorical	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	Categorical	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	Categorical	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	Categorical	Labels of bus lines/stops used by the passengers in the same hour on all previous seven days
	Most used bus line/stop on day-1	7	10	Categorical	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	Categorical	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	Categorical	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	Categorical	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	Categorical	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	Categorical	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		Numerical	Number of trips made by the passengers on the previous day
	Total number of trips on day-7	1		Numerical	Number of trips made by the passengers on the same day last week
	Total number of trips from day-7 to day-1	1		Numerical	Number of trips made by the passengers on all previous seven days
	Total number of trips in the same hour on day-1	1		Numerical	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		Numerical	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		Numerical	Number of trips made by the passengers in the same hour on all previous seven days

4 Case study

4.1 Data 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. From the middle 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, there are 8.4 million people living in Changsha and the urban area covers 1938 square kilometres. Three bus companies operate over 200 bus lines serving more than two million trips per day. The whole bus network covers all the roads in six administrative districts. The accessibility of the 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 at 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 in our case study (in Figure 5). However, these bus lines provide a representative of the bus services connecting key central business districts, high-tech zones and residential zones around the city. 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 parts of the 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.

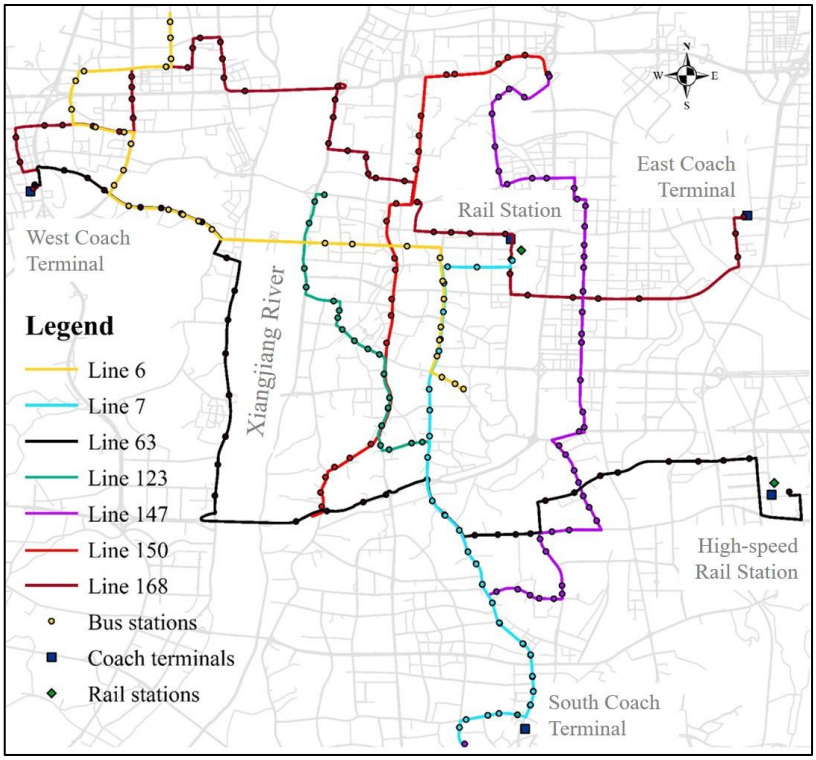

Figure 5 The map of the study case network in Changsha, China

The smart card system in Changsha records the boarding of each trip, i.e. bus lines, vehicle ID, card ID, card type and boarding time. However, the system does not contain a 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 on all buses in the city, and 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.

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 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.

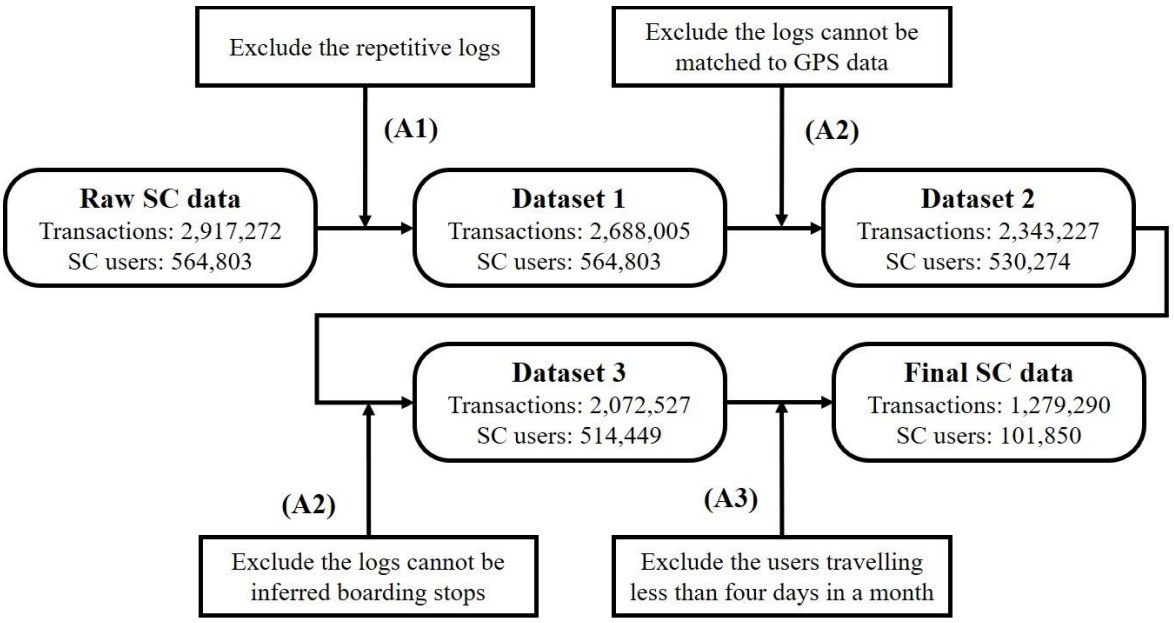

Figure 6 Processes to prepare smart card data.

(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 and take only one of the readings. This 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 because they cannot be inferred from 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 who made less than four trips during the study period. Such infrequent travellers will generator too many non-travelling instances for the model, and they 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 19 time slots (corresponding to the 19 hours of service operation from 6 am to 1 am the following day) in a day so that every user ID has 19 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 makes this an MLCP. If there is no smart card record in a time slot, we label such instances as 'NONE', i.e. not travelling.

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.

Table 2 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	

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 from 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 2 presents the number of instances in the different datasets of models.

Table 3 The structure of the specific machine learning models.

Models		Architectures	The number of nodes					Activation function	
			Input	Hidden 1	Hidden 2	Hidden 3	Output		
Stage 1	Model 1	FCN	133	95	66	-	1	ReLU	sigmoid
		RNN		133	128	64		tanh	
		LSTM		64	32	32		tanh sigmoid	
Stage 2	Model 2	FCN	133	100	75	-	7	ReLU	
		RNN		133	128	128		tanh	
		LSTM		64	32	32		tanh sigmoid	
Stage 3	Model 3.1	FCN	169	128	128	64	33	ReLU	
		RNN		128	64	64		tanh	
		LSTM		128	128	64		tanh sigmoid	
	Model 3.2	FCN		128	128	64	33	ReLU	
		RNN		128	64	64		tanh	
		LSTM		128	128	64		tanh sigmoid	
	Model 3.3	FCN		128	128	64	44	ReLU	
		RNN		128	64	64		tanh	
		LSTM		128	64	64		tanh sigmoid	
	Model 3.4	FCN		128	128	64	46	ReLU	
		RNN		128	64	64		tanh	
		LSTM		128	128	64		tanh sigmoid	
	Model 3.5	FCN		128	128	64	63	ReLU	
		RNN		128	64	64		tanh	
		LSTM		128	128	64		tanh sigmoid	
	Model 3.6	FCN		128	128	64	39	ReLU	
		RNN		128	64	64		tanh	
		LSTM		128	128	64		tanh sigmoid	
	Model 3.7	FCN		128	128	64	48	ReLU	
		RNN		128	64	64		tanh	
		LSTM		128	128	64		tanh sigmoid	

In Section 3.3, we introduced three popular architectures of neural network (FCN, RNN and LSTM) to solve the binary and multi-label classification problems. The specific architectures implemented in this study are presented in Table 3. The number of nodes in the

input and output layer is the same for each of the three architectures. The number of hidden layers and nodes in each hidden layer varies from architecture to architecture. 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 three architectures.

As mentioned in Section 3.3, the input of FCN is the individual instance, but the input of RNN and LSTM is a sequence of instances that 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 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 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.

$$X^p = \left\{ x^p_{t-7m-s+1}, x^p_{t-7m-s+2}, \cdots, x^p_{t-6m-s}, x^p_{t-dm}, x^p_{t-m-s+1}, x^p_{t-m-s+2}, \cdots, x^p_{t-s}, x^p_t \middle| d = 2,3,\cdots,6 \right\} \quad (10)$$

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 19 in this study; d indicates the day.

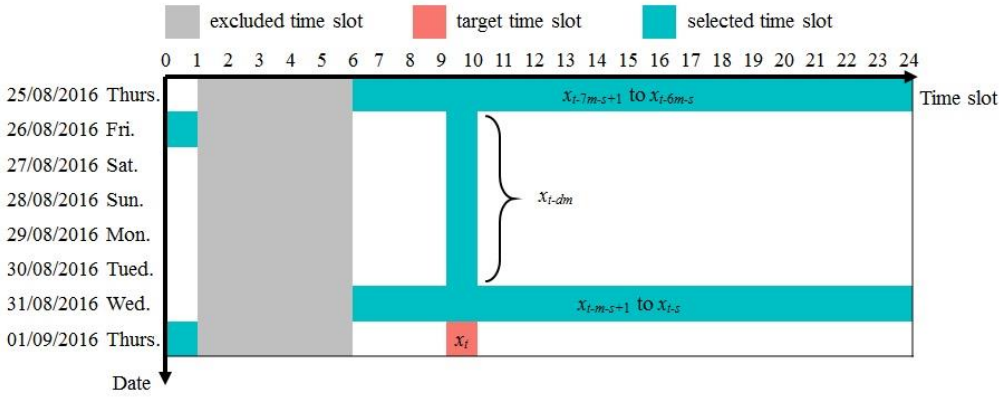

Figure 7 Example to illustrate the time slot selected for the sequence.

5 Model results and discussion

5.1 Performance measurements

The direct result from machine learning models is the boarding stop for every travelling instance. We adopt the confusion matrix (presented in Table 4) and measures on prediction 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. The F1 score is the harmonic mean of the value of precision and recall, which balances the precision and recall of the model. The ground truth for Stage 1 and Stage 2 are the directly recorded trip made (or not made), and for Stage 3 the ground truth is inferred by the geographic information from GPS data.

Table 4 The confusion matrix for a single-label binary classification problem

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

For a single-label binary problem (Stage 1), the performance is measured as:

$$precision = \frac{TP}{TP + FP} \tag{11}$$

$$recall = \frac{TP}{TP + FN} \tag{12}$$

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

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.

For the multi-label problem (Stage 2 and 3), the machine learning models treat them as several single-label binary problems. Each class (bus line or stop) is a single binary classification problem. So, there will a group of TP, FP, FN and TN for each class. We measure the precision and recall for multi-label problem as follows.

$$precision = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} \left(TP_k + FP_k \right)} \tag{14}$$

$$recall = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} \left(TP_k + FN_k \right)} \tag{15}$$

where TP_k, FP_k and FN_k represent the number of *TP*, *FP* and *FN* instances when predicting class *k*.

For the multi-label classification problem (MLCP) of Stage 2 and Stage 3, a more appropriate performance measure is the Hamming Loss (HL) introduced by (Schapire and Singer, 2000). HL measures the fraction of the wrong labels to the total number of labels. A lower HL score indicates the higher performance of models.

$$HL\left(y_m, \hat{y}_m \right) = \frac{1}{M} \sum_{m=1}^{M} \frac{xor\left(y_m, \hat{y}_m \right)}{K} = \frac{\sum_{k=1}^{K} \left(FP_k + FN_k \right)}{MK} \tag{16}$$

where M is the total number of instances to be predicted where m is the index of instances; K denotes the total number of labels; 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.

5.2 Predictions of individual boarding trips

In this section, we examine the performances on the prediction of individual boarding stops, from the three machine learning architectures. Table 5 lists the running time (in seconds) of the three models. It can be seen that RNN is the fastest, in all stages. Since LSTM optimises RNN by adding the gate structure, it is expected to take longer than RNN to run. However, while FCN has the simplest architecture, its running time is consistently longer than RNN, and even longer than LSTM in some cases, suggesting the low computing efficiency of FCN architecture compared to RNN and LSTM architectures.

Table 5 Running time (in seconds) of the machine learning models.

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

Table 6 lists the number of instances for the four measures of the confusion matrix, for the machine learning architectures and one using the random classification method. Only measures for Stage 1 and Stage 2 are presented, as Stage 3 is a multi-class problem, and there would be a confusion matrix for each class. We note firstly from Table 6 that there are very high proportions (up to 97% for Stage 1 models and 80% for Stage 2) of the instances that are true-negative (TN). This is a direct result of the imbalanced issue, as discussed in Section 3, where many instances are simply not travelling (Stage 1) or not travelling on that bus line (Stage 2).

Table 6 The number of TP, FP, FN and TN instances of confusion matrix in Stage 1 and 2.

Stages (Number of instances M)	Architectures	Classes	TP	FP	FN	TN
Stage 1 (1,935,150)	FCN	Travelling or not	28,482	20,039	16,674	1,869,955
	RNN	Travelling or not	27,948	20,573	17,208	1,869,421
	LSTM	Travelling or not	33,625	14,896	11,531	1,875,098
Stage 2	FCN (48,521)	Line 006	7,199	2,358	2,113	36,851
		Line 007	7,391	2,315	2,069	36,746
		Line 063	1,624	1,333	1,230	44,334
		Line 123	3,055	2,162	1,991	41,313
		Line 147	2,225	1,839	1,697	42,760
		Line 150	7,628	2,877	2,598	35,418
		Line 168	2,580	1,909	1,759	42,273
		Average	31,702	14,793	13,457	279,695
	RNN (49,609)	Line 006	6,609	3,221	2,703	37,076
		Line 007	6,792	3,217	2,668	36,932
		Line 063	1,443	1,340	1,411	45,415
		Line 123	2,735	2,275	2,311	42,288
		Line 147	1,977	1,845	1,945	43,842
		Line 150	6,980	3,727	3,246	35,656
		Line 168	2,305	1,981	2,034	43,289
		Average	28,841	17,606	16,318	284,498
	LSTM (49,093)	Line 006	6,844	3,091	2,468	33,264
		Line 007	7,025	3,074	2,435	33,133
		Line 063	1,562	1,424	1,292	41,389
		Line 123	2,930	2,373	2,116	38,248
		Line 147	2,140	1,963	1,782	39,782
		Line 150	7,261	3,628	2,965	31,813
		Line 168	2,476	2,078	1,863	39,250
		Average	30,238	17,631	14,921	256,879

From the confusion matrix values, the performance measures of the machine learning models at each stage of the prediction are derived and presented in Figure 8. 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 8 with Table 2, 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 for 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 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.

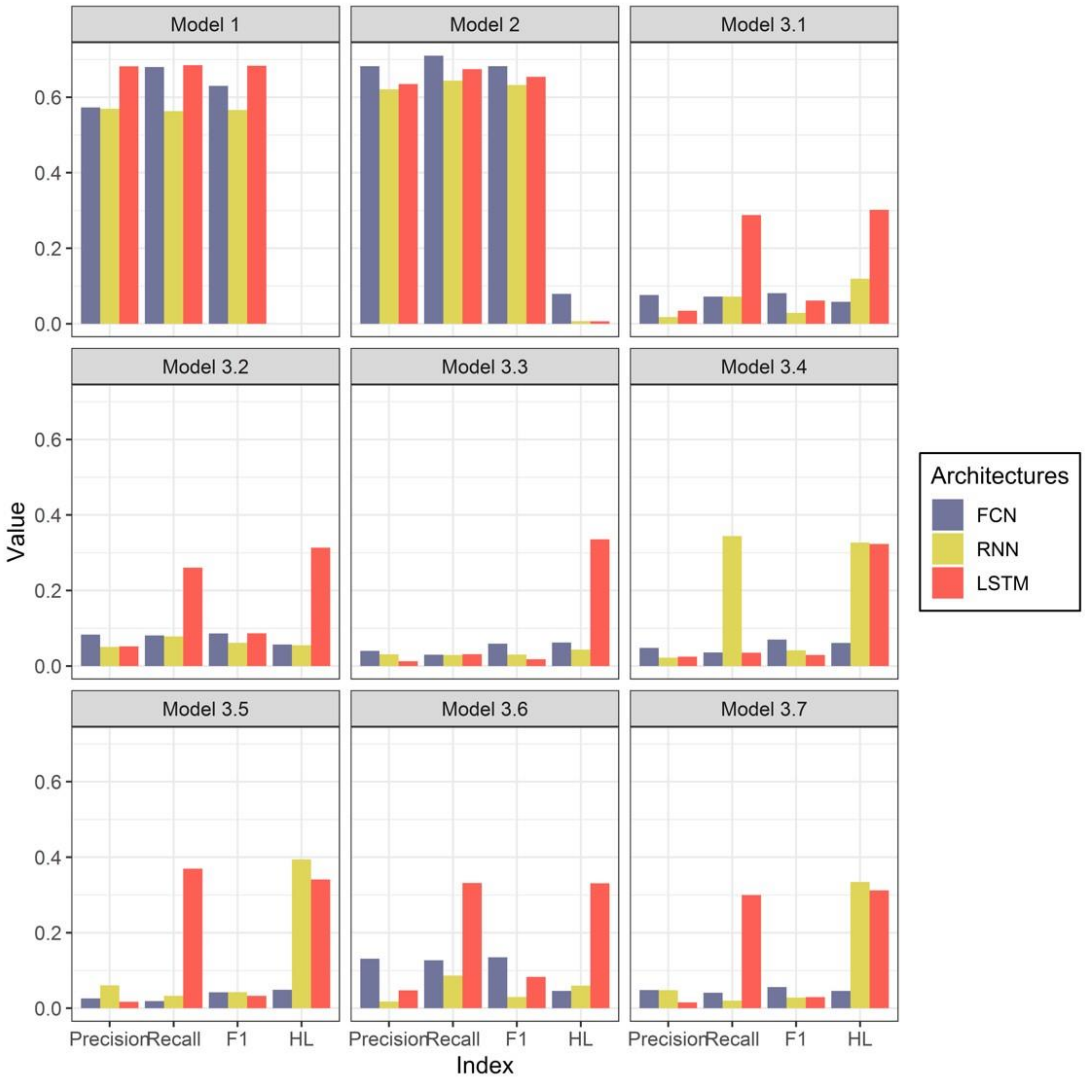

Figure 8 The performance measurements, in terms of Precision, Recall, F1 score and Hamming Loss, of the three different machine learning architectures and for the different prediction stages (models).

We examine the influence of poor prediction of Stage 1 on Stage 2. 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 decrease the performance. Here, we only use TP instances of Stage 1 as the testing dataset of Stage 2 and measure the precision of the new Stage 2. So do models in Stage 3.Figure 9 presents precisions on 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 the stages (Stage 2 and 3) are able to find out which bus lines and stops passengers use.

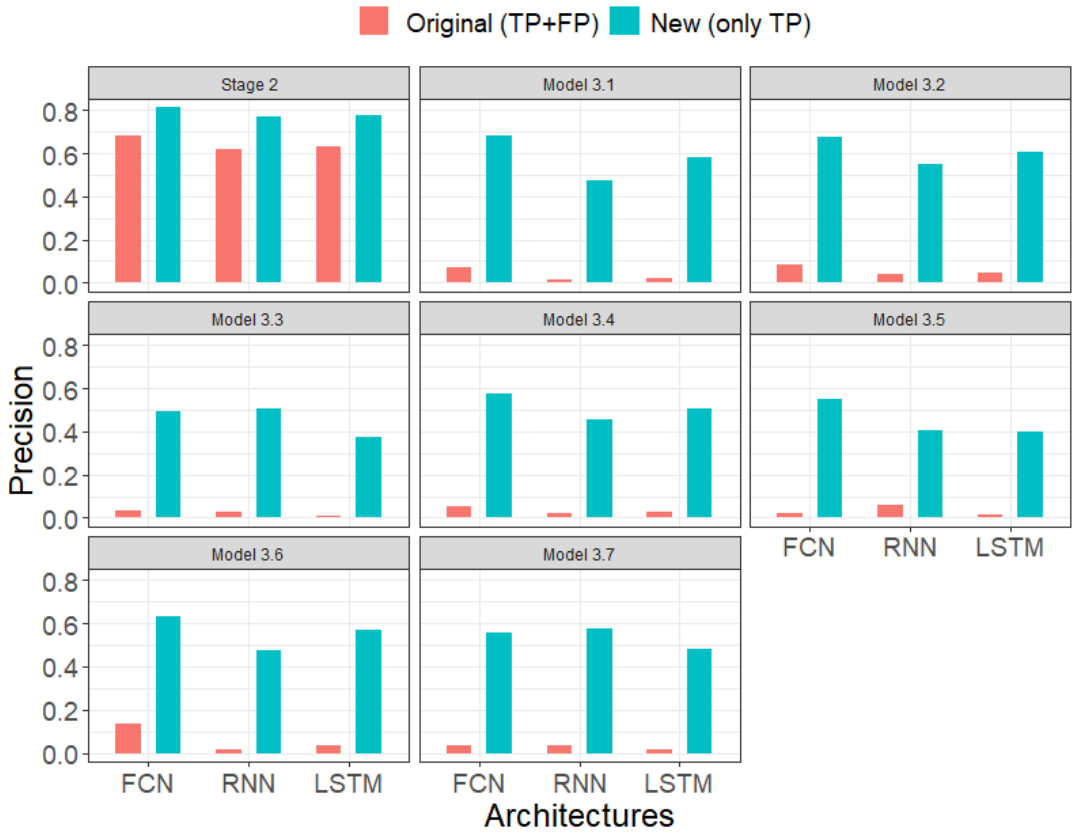

Figure 9 The precision of the model with original testing dataset (with TP+FP) and new testing dataset (TP only).

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). Since we cannot measure the number

of passengers paying by cash, the ground truth of ridership is the statistical results based on the smart card data.

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. The ridership distribution over the day is presented in Figure 10. 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 the evening peak at 18:00. However, it predicted the 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.

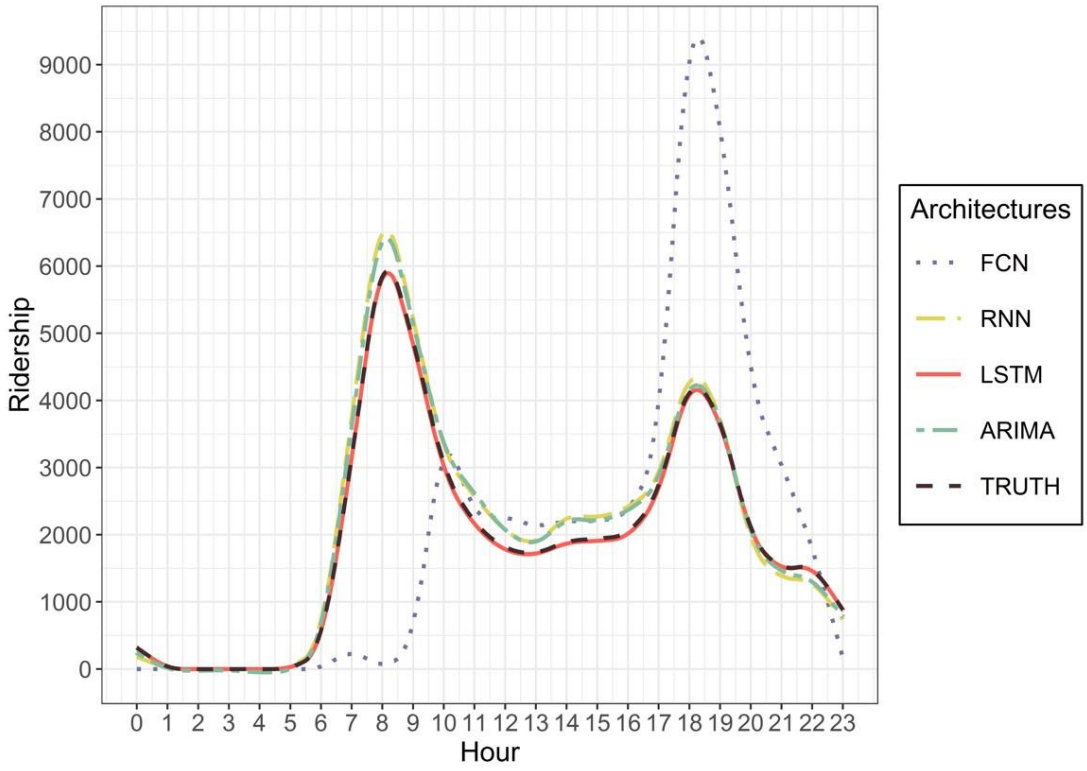

Figure 10 Ridership at the network level: the ground truth and those predicted by different architectures in Stage 1.

Figure 11 presents the true and predicted ridership for bus lines from the results of Stage 2. The overall picture is similar to those in Figure 10 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.

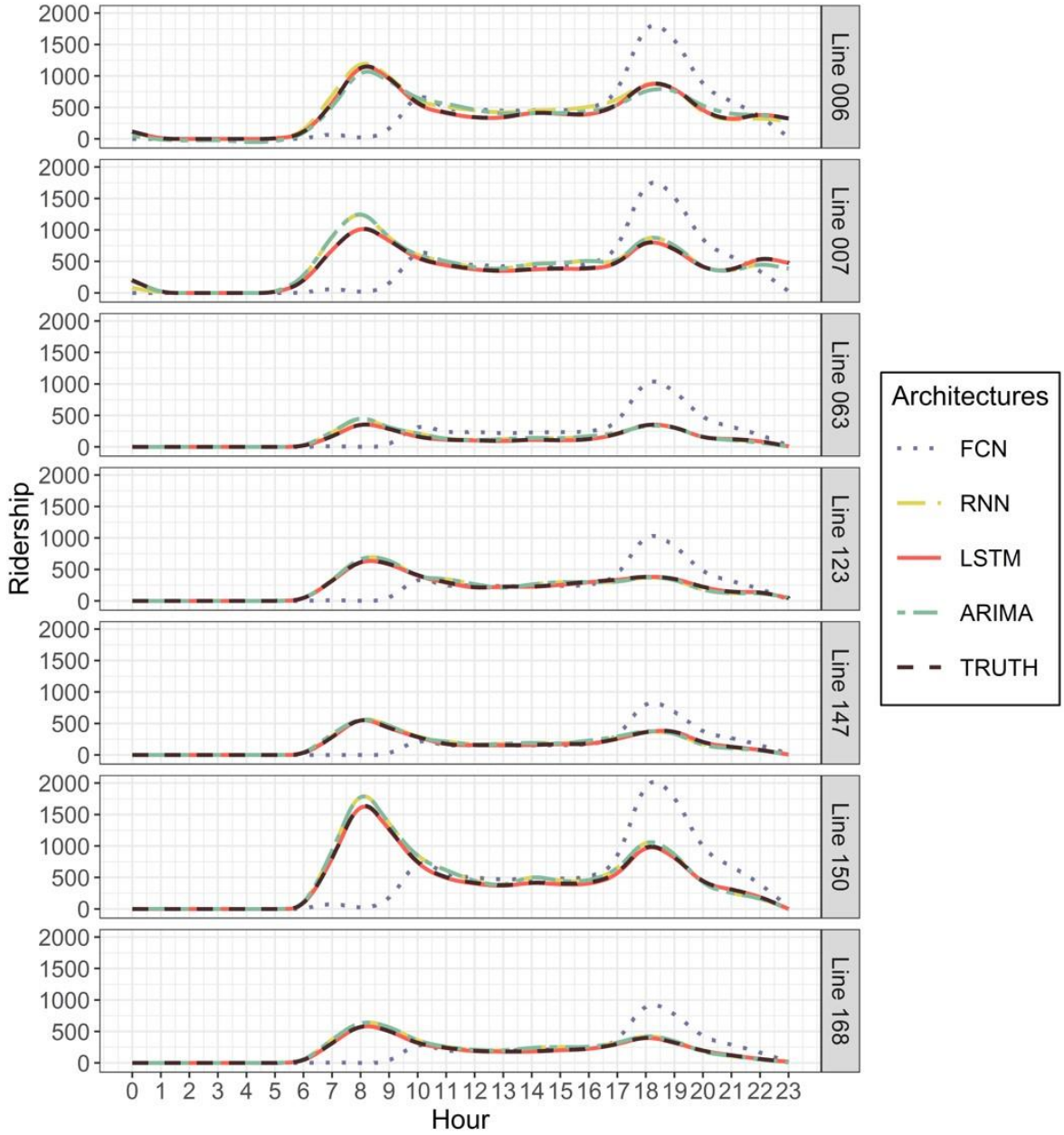

Figure 11 Ridership delivered to bus lines in truth and prediction by Stage 2.

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 12 shows the true and predicted ridership at stop-level from different machine learning architectures. The interpretation of the results in Figure 12 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.

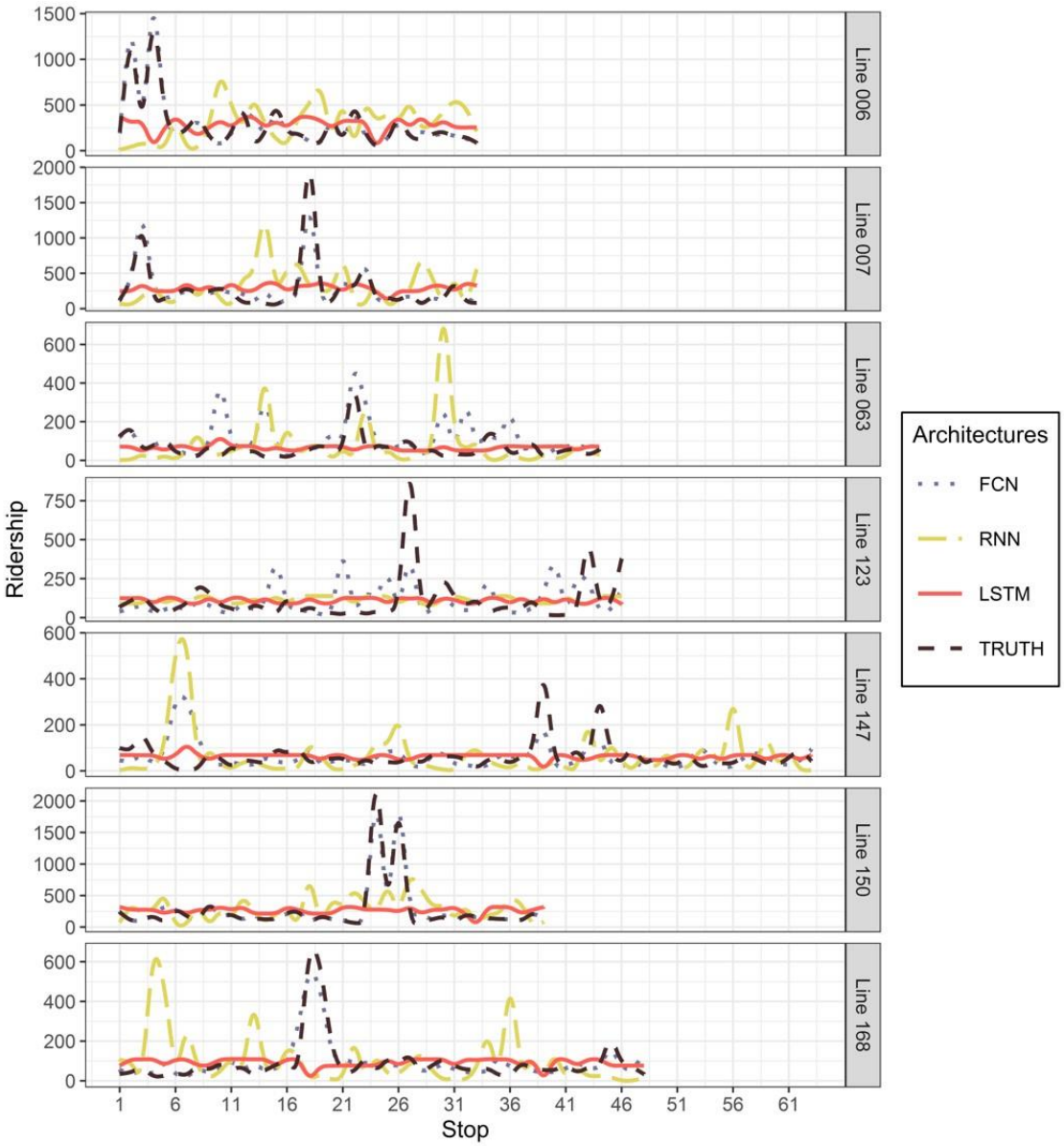

Figure 12 True and predicted ridership at stop-level from different architectures.

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 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 for bus lines is higher than the reality. For RNN and LSTM, their prediction differs from real observation. The indication from Figure 12 is opposite

to that from Figure 10 and 11. On most 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.

It may be noted that Figure 12 presents a whole-day ridership in different bus stops and Figure 11 present the hourly ridership on every bus line. As presented in Figure 12, FCN is the best model to predict the busy stops. However, in Figure 11, FCN has the worst performance in predicting the hourly ridership. Since Figure 11 and 12 present the ridership in terms of different (temporal versus spatial) ways, the opposite conclusions do not conflict. There are many spatial features describing the most used bus stops but limited temporal features related to time. The input of FCN is separate instances. Therefore, the FCN model may ignore the temporal relationship among instances and takes a poor performance in the time-dimension. And many features in travel history are related to the most used bus lines/stops. According to these input information, FCN is easy to find the peak bus stops (corresponding to Figure 12). This difference proves again that FCN is able to capture the daily spatial distribution (along with bus stops) of ridership but lacks the ability on temporal characteristics (over time).

We show the temporal distribution of the demand at the stops with the largest ridership in Figure 13. 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.

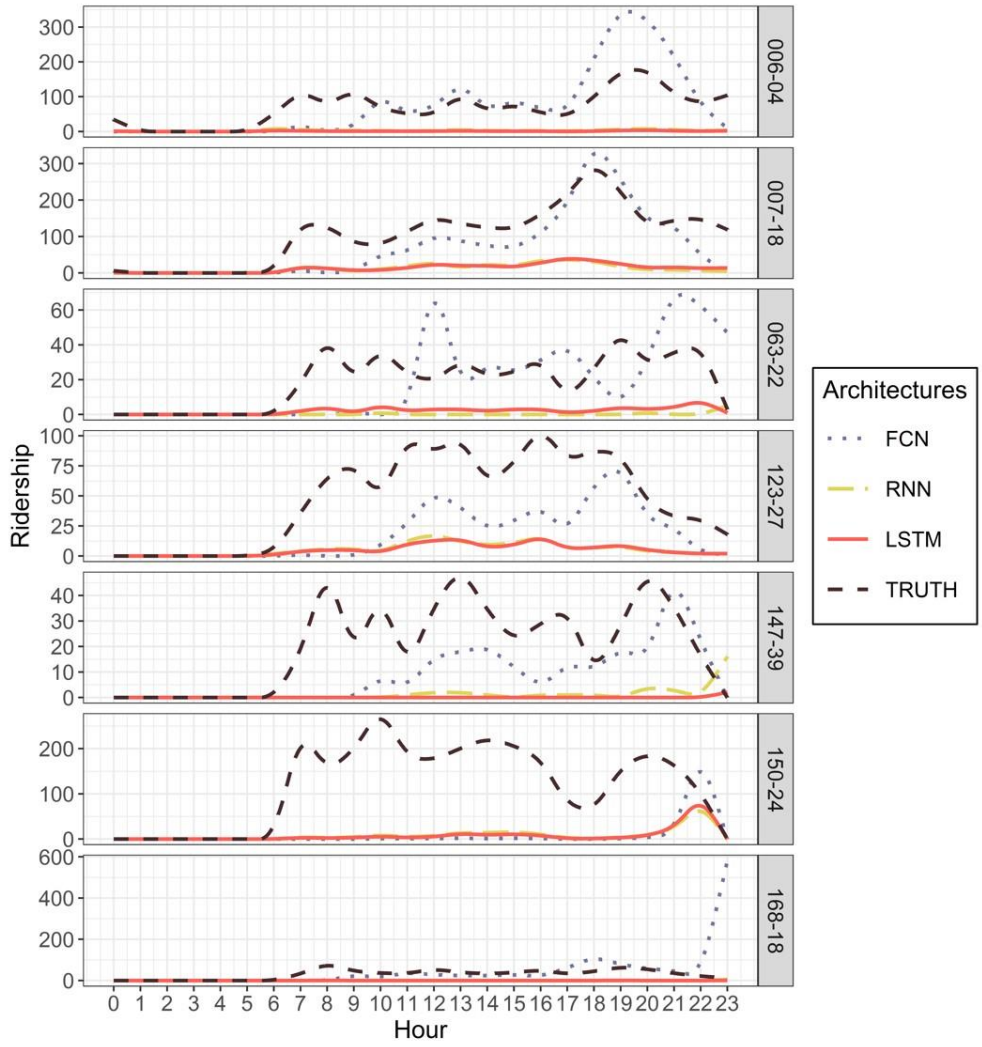

Figure 13 True and predicted ridership at the largest-ridership stop from different architectures.

Overall, 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 fact that LSTM and RNN are able to accurately capture the temporal distribution of ridership is because the methods explicitly consider the temporal relationships inherent in 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.

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 the planner the situation and changes of ridership in the public transport network, and is the basis for long-term planning and short-term operation. Thus, working on predicting the boarding behaviour and making the prediction more accurate will help improve the attraction and patronage of public transport systems, which in turn contributes to the sustainability of cities.

This paper presents a multi-stage framework to predict the boarding stops for each smart card user over an one-hour time slot. First, we predict the states, travelling or not, for each instance. 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. 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. Using results concerning individual users, it is easy to obtain the aggregated ridership at stop-, line- and

network-level. Thus, it reduces the number of models required. When predicting the ridership at stop-level, the classic model, such as ARIMA, needs to build up a single model for each bus stop. LSTM and RNN are shown to produce accurately the time-dependent distribution of the line-level ridership.

We discuss the reasons why the prediction accuracy 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 the 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. All 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, but FCN has a poor performance to predict the 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 prediction framework can be easily transferred to other bus systems trained with local smart card datasets, which are generally available to operators. 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 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 limited study time period 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. Since the data size far exceeded our computation capacity, we have to delete the infrequent passenger in the case study since these passengers can only provide limited information on 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 continuous stages to avoid the many-class issues, too many stops in a bus line still challenges machine learning models. 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 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 the boarding behaviour. However, to guide the public transport planning, it is important to have the full picture of passenger flow origin-destination matrix that contains both the boarding and alighting information. Therefore, combining the alighting stop prediction (Tang et al., 2020) with this study will be a way to clearly understand the travel pattern.

Acknowledgments

We would like to acknowledge the support from the UK Department for Transport (Project "Future Streets") and the National Natural Science Foundation of China (71890972/71890970). We would also like to thank Hunan Longxiang Bus Co., Ltd. and Changsha Meteorological Bureau for making the smart card and weather data available for this study.

References

ALRUKAIBI, F. & ALKHEDER, S. 2019. Optimisation of bus stop stations in Kuwait. *Sustainable Cities and Society,* 44**,** 726-738.

BARRY, J., FREIMER, R. & 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**,** 53-61.

BARRY, J., NEWHOUSER, R., RAHBEE, A. & 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**,** 183-187.

BEIJING TRANSPORT INSTITUTE 2019. Beijing Transport Development Annual Report. *In:* INSTITUTE, B. T. (ed.). Beijing, China.

BERREBI, S. J., WATKINS, K. E. & LAVAL, J. A. 2015. A real-time bus dispatching policy to minimise passenger wait on a high frequency route. *Transportation Research Part B: Methodological,* 81**,** 377-389.

BöCKER, L., DIJST, M. & PRILLWITZ, J. 2013. Impact of everyday weather on individual daily travel behaviours in perspective: A literature review. *Transport Reviews,* 33**,** 71-91.

BORDAGARAY, M., DELL'OLIO, L., IBEAS, A. & CECíN, P. 2013. Modelling user perception of bus transit quality considering user and service heterogeneity. *Transportmetrica A: Transport Science,* 10**,** 705-721.

BORDAGARAY, M., DELL'OLIO, L., FONZONE, A. & 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**,** 231-248.

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

CHOLLET, F. & OTHERS. 2015. *Keras* [Online]. [Accessed].

CONNOR, J. T., MARTIN, R. D. & ATLAS, L. E. 1994. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks,* 5**,** 240-254.

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

DEPARTMENT FOR TRANSPORT 2019. National Travel Survey: 2018. *In:* DEPARTMENT FOR TRANSPORT (ed.). UK.

FAROQI, H., MESBAH, M. & KIM, J. 2017. Spatial-temporal similarity correlation between public transit passengers using smart card data. *Journal of Advanced Transportation,* 2017**,** 1-14.

FAROQI, H., MESBAH, M. & 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**,** 1-16.

FONZONE, A., SCHMöCKER, J.-D. & LIU, R. 2015. A model of bus bunching under reliability-based passenger arrival patterns. *Transportation Research Part C: Emerging Technologies,* 59**,** 164-182.

GODBOLE, S. & SARAWAGI, S. Discriminative methods for multi-labeled classification. *In:* DAI, H., SRIKANT, R. & ZHANG, C., eds. Advances in Knowledge Discovery and Data Mining, 2004// 2004 Berlin, Heidelberg. Springer Berlin Heidelberg, 22-30.

GOLDENBELD, C., LEVELT, P. B. M. & HEIDSTRA, J. 2000. Psychological perspectives on changing driver attitude and behaviour. *Recherche - Transports - Sécurité,* 67**,** 65-81.

GONG, M., FEI, X., WANG, Z. & QIU, Y. 2014. Sequential framework for short-term passenger flow prediction at bus stop. *Transportation Research Record: Journal of the Transportation Research Board,* 2417**,** 58-66.

GONZáLEZ, M. C., HIDALGO, C. A. & BARABáSI, A.-L. 2008. Understanding individual human mobility patterns. *Nature,* 453**,** 779-782.

GORDON, J., KOUTSOPOULOS, H., WILSON, N. & 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**,** 17-24.

HAN, Y., WANG, S., REN, Y., WANG, C., GAO, P. & 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**,** 243.

HE, L., AGARD, B. & 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**,** 56-75.

HE, L., TRéPANIER, M. & 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. & 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. & TIAO, G. C. 1982. An ARIMA-model-based approach to seasonal adjustment. *Journal of the American Statistical Association,* 77**,** 63-70.

HOCHREITER, S., BENGIO, Y., FRASCONI, P. & SCHMIDHUBER, J. 2001. *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*, IEEE Press.

HOCHREITER, S. & SCHMIDHUBER, J. 1997. Long short-term memory. *Neural Computation,* 9**,** 1735-1780.

HOLLANDER, Y. & LIU, R. 2008. Estimation of the distribution of travel times by repeated simulation. *Transportation Research Part C: Emerging Technologies,* 16**,** 212-231.

JIANG, X., ZHANG, L. & 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**,** 110-127.

JIN, W., LI, P., WU, W. & WEI, L. Short-term public transportation passenger flow forecasting method based on multi-source data and shepard interpolating prediction method. *In:* LONG, S. & DHILLON, B. S., eds. Man-Machine-Environment System Engineering, 2019// 2019 Singapore. Springer Singapore, 281-294.

KARNBERGER, S. & ANTONIOU, C. 2020. Network–wide prediction of public transportation ridership using spatio–temporal link–level information. *Journal of Transport Geography,* 82**,** 102549.

KOETSE, M. J. & 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**,** 205-221.

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

LI, Y., WANG, X., SUN, S., MA, X. & 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**,** 306-328.

LIU, R. & SINHA, S. 2007. Modelling urban bus service and passenger reliability. *International Symposium on Transportation Network Reliability.* Hague.

LIU, X., ZHOU, Y. & RAU, A. 2019a. Smart card data-centric replication of the multi-modal public transport system in Singapore. *Journal of Transport Geography,* 76**,** 254-264.

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

MA, F., REN, F., YUEN, K. F., GUO, Y., ZHAO, C. & GUO, D. 2019. The spatial coupling effect between urban public transport and commercial complexes: A network centrality perspective. *Sustainable Cities and Society,* 50**,** 101645.

MA, Z.-L., FERREIRA, L., MESBAH, M. & HOJATI, A. T. 2015. Modeling bus travel time reliability with supply and demand data from automatic vehicle location and smart card systems *Transportation Research Record,* 2533**,** 17-27.

MA, Z., XING, J., MESBAH, M. & FERREIRA, L. 2014. Predicting short-term bus passenger demand using a pattern hybrid approach. *Transportation Research Part C: Emerging Technologies,* 39**,** 148-163.

ORANSIRIKUL, T., NISHIDE, R., PIUMARTA, I. & TAKADA, H. 2014. Measuring bus passenger load by monitoring Wi-Fi transmissions from mobile devices *Procedia Technology,* 18**,** 120-125.

SAKAI, K., LIU, R., KUSAKABE, T. & 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**,** 737-748.

SCHAPIRE, R. E. & SINGER, Y. 2000. BoosTexter: A boosting-based system for text categorisation. *Machine Learning,* 39**,** 135-168.

SCHNEIDER, C. M., BELIK, V., COURONNé, T., SMOREDA, Z. & GONZáLEZ, M. C. 2013. Unravelling daily human mobility motifs. *Journal of The Royal Society Interface,* 10**,** 20130246.

SIERPIŃSKI, G. 2016. *Intelligent Transport Systems and Travel Behaviour,* Katowice, Poland, Springer.

SORRATINI, J., LIU, R. & SINHA, S. 2008. Assessing bus transport reliability using micro-simulation. *Transportation Planning and Technology,* 31**,** 303-324.

STOVER, V. W. & MCCORMACK, E. D. 2012. The impact of weather on bus ridership in Pierce County, Washington. *Journal of Public Transportation,* 15**,** 6.

SUN, Y., SHI, J. & SCHONFELD, P. M. 2016. Identifying passenger flow characteristics and evaluating travel time reliability by visualising AFC data: a case study of Shanghai Metro. *Public Transport,* 8**,** 341-363.

SVOZIL, D., KVASNICKA, V. & POSPICHAL, J. í. 1997. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems,* 39**,** 43-62.

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.

TONG, H. Y. 2019. Development of a driving cycle for a supercapacitor electric bus route in Hong Kong. *Sustainable Cities and Society,* 48**,** 101588.

TOQUé, F., CôME, E., EL MAHRSI, M. K. & OUKHELLOU, L. 2016. Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC).* Rio de Janeiro, Brazil: IEEE.

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

WASHINGTON, S. P., KARLAFTIS, M. G. & MANNERING, F. 2010. *Statistical and Econometric Methods for Transportation Data Analysis*, Chapman and Hall/CRC.

WEI, M., LIU, Y., SIGLER, T., LIU, X. & CORCORAN, J. 2019. The influence of weather conditions on adult transit ridership in the sub-tropics. *Transportation Research Part A: Policy and Practice,* 125**,** 106-118.

WEI, Y. & 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**,** 148-162.

WU, W., LIU, R. & JIN, W. 2017. Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour. *Transportation Research Part B: Methodological,* 104**,** 175-197.

WU, W., LIU, R., JIN, W. & MA, C. 2019. Stochastic bus schedule coordination considering demand assignment and rerouting of passengers. *Transportation Research Part B: Methodological,* 121**,** 275-303.

XU, S., LIU, R., LIU, T. & HUANG, H. 2018. Pareto-improving policies for an idealised two-zone city served by two congestible modes. *Transportation Research Part B: Methodological,* 117**,** 876-891.

XUE, R., SUN, D. & 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**,** 1-11.

YANG, X. & 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**,** 8.

YANG, Y., HEPPENSTALL, A., TURNER, A. & 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**,** 101361.

YANG, Y., HEPPENSTALL, A., TURNER, A. & 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**,** 271.

YANG, Y., HEPPENSTALL, A., TURNER, A. & 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**,** 101521.

YANG, Z.-W., ZHAO, Q., ZHAO, S.-C., JIN, L. & MAO, Y. 2009. Passenger flow volume forecasting method based on public transitintelligent card (IC) survey data. *Transport Standardization,* 9**,** 115-119.

YAO, E., LIU, T., LU, T. & YANG, Y. 2020. Optimisation of electric vehicle scheduling with multiple vehicle types in public transport. *Sustainable Cities and Society,* 52**,** 101862.

YU, B., LAM, W. H. & TAM, M. L. 2011. Bus arrival time prediction at bus stop with multiple routes. *Transportation Research Part C: Emerging Technologies,* 19**,** 1157-1170.

ZHANG, X., ZHANG, Q., SUN, T., ZOU, Y. & 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**,** 357-365.

ZHAO, J., RAHBEE, A. & WILSON, N. H. 2007. Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. *Computer-Aided Civil and Infrastructure Engineering,* 22**,** 376-387.

ZHOU, C., DAI, P. & LI, R. The passenger demand prediction model on bus networks. *In:* DALLAS, T., USA, ed. 2013 IEEE 13th International Conference on Data Mining Workshops, 7-10 Dec. 2013 2013 IEEE. 1069-1076.

ZHOU, M., WANG, D., LI, Q., YUE, Y., TU, W. & CAO, R. 2017. Impacts of weather on public transport ridership: Results from mining data from different sources. *Transportation Research Part C: Emerging Technologies,* 75**,** 17-29.

ZHOU, X., WANG, M. & LI, D. 2019. Bike-sharing or taxi? Modeling the choices of travel mode in Chicago using machine learning. *Journal of Transport Geography,* 79**,** 102479.