Predicting hourly boarding demand of bus passengers using imbalanced records from smart-cards: A deep learning approach

Tianli Tang, Ronghui Liu*, Charisma Choudhury, Achille Fonzone, and Yuanyuan Wang

***Abstract*—The tap-on smart-card data provides a valuable source to learn passengers' boarding behaviour and predict future travel demand. However, when examining the smart-card records (or instances) by the time of day and by boarding stops, the positive instances (i.e. boarding at a specific bus stop at a specific time) are rare compared to negative instances (not boarding at that bus stop at that time). Imbalanced data has been demonstrated to significantly reduce the accuracy of machine-learning models deployed for predicting hourly boarding numbers from a particular location. This paper addresses this data imbalance issue in the smart-card data before applying it to predict bus boarding demand. We propose the deep generative adversarial nets (Deep-GAN) to generate dummy travelling instances to add to a synthetic training dataset with more balanced travelling and non-travelling instances. The synthetic dataset is then used to train a deep neural network (DNN) for predicting 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 predictive model's performance and better fit ridership's actual profile. Comparing the performance of the Deep-GAN with other traditional resampling methods shows that the proposed method can produce a synthetic training dataset with a higher similarity and diversity and, thus, a stronger prediction power. The paper highlights the significance and provides practical guidance in improving the data quality and model performance on travel behaviour prediction and individual travel behaviour analysis.**

***Index Terms*—Boarding behaviour prediction, Smart-card, Bus, Data imbalance issue, Deep generative adversarial network, Deep neural network.**

I. Introduction

THE rapid progress of urbanisation leads to expansion of population in the urban area, increased demand for travel and associated adverse effects in traffic congestion and air pollution [1]–[3]. Public transport has been widely recognised as a green and sustainable mode of transportation to relieve such transport problems. As a conventional public transport mode, buses have always played a dominant role in passenger transportation [4], [5]. However, unreliable travel time, bus-bunching and crowding have led to low level-of-services for buses [6]–[8]. This has decreased the bus ridership in many cities, particularly with the advent of ride-hailing services in recent years [9]–[11]. To sustain and increase bus patronage, bus operators must find a way to improve its performance and enhance its image and attraction. Advanced operation and management for bus systems can significantly improve the level-of-service and service reliability, which in turn helps increase the bus ridership [12]–[14]. This requires understanding the spatial and temporal variations in passenger demand and making necessary changes on the supply side [15]–[18].

The smart-card system is initially designed for automatic fare collection. As the system also records the boarding information, for example, who gets on buses, where and when, smart-card data has become a ready-made and valuable data source for spatio-temporal demand analysis [19], public transport planning [20]–[23], and further analysis of emission reduction for the sustainable transport [24], [25]. From the smart-card data, we can easily observe the passenger flow at bus stops and on bus lines, and from which to derive the spatial and temporal characteristics of bus trips [26], [27]. However, extracting useful information from big data automatically still poses a significant challenge. In recent years, machine learning techniques have emerged as an efficient and effective approach to analysing large smart-card datasets. For instance, Liu *et al.* [28] captured key features in public transport passenger flow prediction via a decision tree model. Zuo *et al.* [29] built a three-stage framework with a neural network model to forecast the individual accessibility in bus systems.

In our own recent research [30], we demonstrate that smart-card data combined with machine learning techniques can be a powerful approach for predicting the spatial and temporal patterns of bus boarding. The predictions were found to be highly accurate at an aggregated level, averaged over all travellers. However, our research has also thrown light on the data imbalance issues, when trying to predict travel behaviour at the level of individual travellers and fine spatial-temporal

We acknowledge the financial support from the National Natural Science Foundation of China (Project No. 71890972/71890970). Tianli Tang is supported by the Key Project of National Natural Science Foundation of China (No. 52131203) and the project of Jiangsu Funding Program for Excellent Postdoctoral Talent. Charisma Choudhury acknowledges the financial support of her UKRI Future Leader Fellowship, UK [MR/T020423/1-NEXUS]. Yuanyuan Wang is supported by the Natural Science Foundation of Zhejiang Province (LQ18G030012), the Humanities and Social Sciences Fund of the Ministry of Education (18YJC630190). The authors would also like to thank Hunan Longxiang Bus Co., Ltd. for providing the smart-card data for this study and .

T. Tang is with the School of Transportation, Southeast University, Nanjing, 211189, China (email: T-Tang@seu.edu.cn).

R. Liu and C. Choudhury are with the Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, United Kingdom (emails: R.Liu@its.leeds.ac.uk, C.F.Choudhury@leeds.ac.uk)

A. Fonzone is with the Transport Research Institute, Edinburgh Napier University, Edinburgh, EH10 5DT, United Kingdom (emails: a.fonzone@napier.ac.uk)

Y. Wang is with the School of Business Administration, Zhejiang University of Finance and Economics, Hangzhou, 310018, China (emails: wangyuan@zufe.edu.cn)

* Coresponding author. Email: R.Liu@its.leeds.ac.uk.

details. For instance, the boarding of an individual smart-card holder at a specific stop during a particular time window (e.g. an hour) is a rare event: most of the records would denote negative (non-travelling, or not boarding at this bus stop during this time window) instances, and only a few are positive (travelling, boarding at this stop at this time) instances. Such data imbalance issues can significantly reduce the efficiency and accuracy of machine learning models deployed for predicting travel behaviour at the level of individual travellers and fine spatial-temporal details. This motivates this current study where we propose an over-sampling method, deep generative adversarial nets (Deep-GAN) model (initially developed in the context of image generation) to address the data imbalance issue in predicting disaggregate boarding demand (i.e. individual passengers boarding behaviour during each hour of the day). We show that, with the synthesised and more balanced database, the prediction accuracy improves significantly. The performance of the proposed approach, based on the Deep-GAN method, is further benchmarked against other resampling methods (including Synthetic Minority Oversampling Technique and Random Under-Sampling) and is shown to have superior performance.

The rest of the paper is organised as follows. Section II reviews the key resampling methods and their applications in transport studies. Section III describes the specific data imbalance issue in predicting the hourly boarding demand. Section IV uses a Deep-GAN to provide a synthesised, more balanced training data sample and a deep neural network (DNN) to predict the individual smart-card holders' boarding actions (boarding or not boarding) in any hour of a day. Section V applies the proposed method to a real-world case study, and the results are discussed in Section VI. Finally, Section VII summarises the main findings and contributions of this paper and suggests future investigations.

II. Data imputation methods

Data imbalance is a common issue in many real-world contexts. Examples include fault diagnosis, anomaly detection, malware detection [31]. In this section, we review the general resampling methods developed to re-balance the datasets and their applications in transport systems.

A. 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-writing recognition [32], Iris Plants Database for pattern recognition [33] and ImageNet data of image recognition [34]. In many real-world problems, however, the data is not all as good as those standard datasets. A particular issue is data imbalance, where the positive instances are the minority, and the negative instances are the majority. For example, when detecting dangerous behaviour [35], [36], the event (dangerous) instances are much less than the normal (non-dangerous) instances. The skewed distribution of classes, or imbalanced data, challenges the traditional machine learning models in a number of ways:

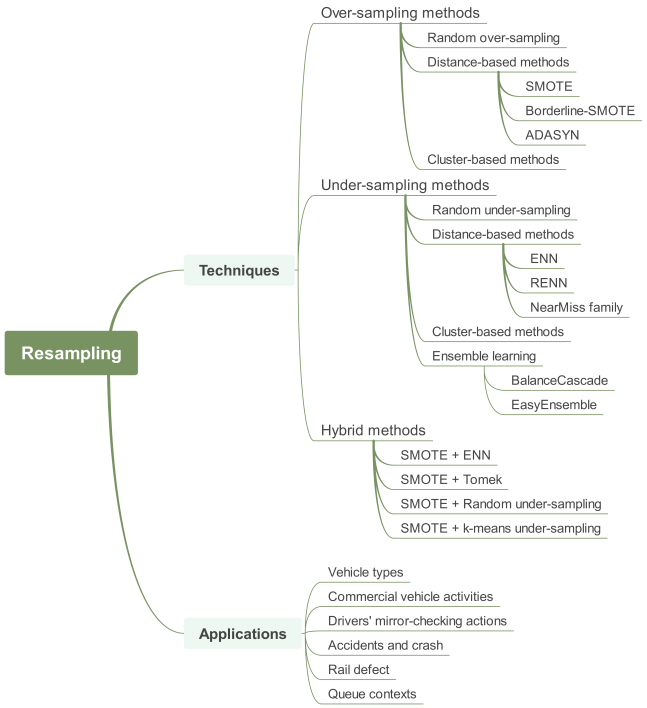

Fig. 1. Reviewed key resampling techniques.

i) rare minor instances may be mistaken as noise and vice versa [37]; ii) minor instances overlap with other regions where the prior probabilities of classes are almost equal [38]; iii) it is challenging to detect underlying patterns for the minor instances with high-dimension features [39].

Data resampling methods have been developed to address such data imbalance issues. The basic idea of resampling is to produce a more balanced dataset. There are three main resampling methods: over-sampling, under-sampling and hybrid method [40]. 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 combine over-sampling or the minority and the under-sampling on the majority instances. Next, we briefly introduce the key models developed under each of the three main resampling methods and summarise them in Figure 1.

The simplest over-sampling method is random over-sampling [41], which randomly duplicates those minor instances. The new dataset created by the random over-sampling method can superficially enhance the existing minor instances which then leads to overfitting. Many repetitive instances may lead to the specification of the classifier on the dominant instances. In addition, if some of the instances are mislabelled or noisy, the error can easily be multiplied. To overcome such deficiencies, a distance-based over-sampling method, synthetic minority over-sampling technique or in short SMOTE [42], has been developed. SMOTE uses the k-nearest neighbours (kNN) algorithm to calculate the closest instances of each instance in the minority, and randomly selects several neighbours according to the imbalance ratio and randomly generates a new

instance between the central instance and neighbour instances. Hence SMOTE can reduce the risk of overfitting. However, it can increase the possibility of overlapping as well, and it does not provide new instances with useful information. Han *et al.* [43] proposed 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 the minority category. He *et al.* [44] 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 compared to the SMOTE. However, they inherit some common failings of the SMOTE, namely that: i) the scattered data distribution, feature redundancy and feature irrelevance in high-dimension data challenge the algorithm to identify minor instances, and ii) the SMOTE and associated methods are not suitable for the categorised features (e.g. weather types, seasons) because their distances cannot be calculated. Besides distance-based methods, clustering is another over-sampling method. Jo and Japkowicz [45] made use of the K-means clustering to categorise the imbalanced dataset and does the over-sampling process for each category of data.

The development of machine learning techniques has promoted advanced data-driven models based on the generative adversarial nets (GAN) for the over-sampling. Shamsolmoali *et al.* [46] proposed the capsule adversarial network, an improved GAN by combining both majority and minority classes, to recognise highly overlapping classes. Jiang and Ge [47] enhanced the quality of generated data from the GAN by using the Mahalanobis-distance-based data filtering method and the Euclidean-distance-based data purification method, where the randomness of the generation process of the GAN reduces the quality of the generated data. Goodfellow *et al.* [48] stated that GAN-based models can generate clearer and more realistic samples compared to other methods. GAN-based models have been widely applied in image generation, image super resolution and image inpainting. As far as we are aware, this is the first time a GAN-based model has been applied in addressing a public transport data imbalance issue for predicting the individual travel choices.

As opposed to over-sampling, the 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 approaches can also be used in under-sampling. In distance-based methods, Wilson [49] 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 [50]. Besides, Mani and Zhang [51] proposed 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; Most-Distance selects the major instances with the most massive average distances to three closest minor instances. For the clustering-based method, Yen and Lee [52] partitioned all the data into several clusters and randomly selected the instances from each cluster. Zhang *et al.* [53] applied k-means clustering in the partition process and used the number of major instances in every cluster as a weight to decide the number of selected major instances. As stated before, the under-sampling method may miss some information. To overcome the weakness of missing information, two methods, EasyEnsemble [54] and BalanceCascade [55], were 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 [56] showed 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 [57] and González *et al.* [58] suggested 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.

For very large training datasets, a hybrid approach that combines the over-sampling and under-sampling methods has been applied. For example, the integration of SMOTE with ENN and SMOTE with Tomek links were developed by Batista *et al.* [41]. Jian *et al.* [59] used SMOTE over-sampling and multiple random under-sampling with ensemble learning in support vector machine classification. Song *et al.* [60] combined 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 [61]. Although the hybrid method shows a greater ability to rebalance datasets than using a single method, the performance of the different combinations of methods can vary significantly (see the examples of [62]–[64]). It requires significantly computing effort to test which combination is the best selection for our case.

Given that the under-sampling methods could lose the information in the majority (non-travelling) data, the over-sampling methods are widely used in addressing data imbalanced issues. In Section II-B, all applications of resampling methods in the transport domain use the over-sampling methods. Therefore, we will model the data imbalance issue in predicting the hourly

boarding behaviour of bus passengers via a GAN-based model.

B. Imbalanced data issue in the transport domain

When data imbalance occurs in real-life applications, it is often the case that the minority class is usually the more important one [65]. In the field of transport, the problem of data imbalance is especially acute in accident detection, where the data representing accidents are the minority while the no-accident scenario is in the majority. Appropriate data imputation has been shown to help improve accident data analysis and prediction. For example, Park and Ha [66] proved that over-sampling by the data mining tools, Hive and MapReduce, can improve the precision in predicting traffic accidents. Parsa *et al.* [67] compared the performance of the SMOTE, Borderline-SMOTE and SVM-SMOTE used in over-sampling the minor accident instances. They found that all three methods have similar accuracy, but SMOTE tends to have a higher detection rate and lower false alarm rate. Sharifirad *et al.* [68] enhanced 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.* [69] used the deep convolutional GAN 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 vehicle type recognition [70] and commercial vehicle activity prediction [71]. Hajizadeh *et al.* [72] used semi-supervised techniques of self-training and co-training to identify and add minor instances for detecting the rail defect from rail image data, and they showed that semi-supervised techniques perform better than other classic over-sampling methods such as SMOTE and random over-sampling. Similarly, Mohammadi *et al.* [73] applied the ADASYN method to overcome the imbalanced data issue when predicting rail defects by track geometry measurement dataset. Rahaman *et al.*. [74] predicted the queue context (various states of queues related to taxis and passengers) 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 better than the original dataset in every evaluation index and that the random over-sampling performs the best.

C. Research gaps and scopes

To sum up, researchers have contributed to developing resampling methods for the imbalanced datasets, and some works have been aware of the detriments of data imbalance issue in transport systems. However, there are still some research limitations in the existing work.

- The data generated by SMOTE and ADASYN are susceptible to outliers. They may generate some data in the majority data space due to minority outlier instances (usually noisy data), causing blurred classification borderlines and making the learning difficulties of the classification model.
- The under-sampling methods usually have to pay the price of losing parts of the information of the majority of data because they have to remove a part of the data. Although the EasyEnsemble and BalanceCascade tried to solve the problem of lost information, they increased the number of models tens of times, significantly increasing the computational burden.
- Little study has noticed the loss caused by the data imbalance issue in the public transport system. There is also no research to validate the efficiency of the existing resampling methods on imbalanced data in the boarding prediction task.

In general, the data imbalance issue is more acute in predicting individual behaviour or a particular type of event. In our previous study of predicting public transport board demand [30], we showed that the prediction is good at the aggregated level, but is poor when we tried to predict the hourly boarding behaviour of individual bus users, due to the data imbalance issue. In Section III-A, we introduce this particular data imbalance issue relating to public transport demand prediction. Therefore, this study proposed the Deep-GAN as the over-sampling method to address the data imbalance issue in the boarding behaviour prediction task for creating a more balanced data sample. The Deep-GAN is based on the deep learning approaches, which has the advantages of dealing with the massive-amount and high-dimension data compared with the existing works. Additionally, the Deep-GAN learn the feature pattern of minority data, while the other over-sampling methods generate the data in the space of minority data. Compared to other over-sampling methods, Deep-GAN is more able to ensure the similarity of the synthetic data to the real data and, at the same time, makes the generated data diverse. This capability allows the synthetic training dataset not to over-enhance some features and avoids the learning bias in the prediction model.

III. Formulating boarding behaviour instances from the smart-card data

A. Description of the data imbalance issue

The target of this study is to predict the hourly boarding demand for bus systems. We select one hour as a prediction time slot in this study. We then model individual passengers' boarding status, travelling or not, for any hour during the operation period as the measure of demand. Thus, the instance in this study is the trip made by a passenger during a specific time slot. 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:

$$r_t^p = (x_t^p, y_t^p) \tag{1}$$

where r_t^p denotes the trip r of passenger p during the time slot t; x_t^p is a feature vector describing the trip r_t^p; and y_t^p is the label to estimation on the travel behaviour of trip r_t^p. Feature vector, x, contains several features that characterise the trip (e.g. the hour in which the boarding was made), the environment (e.g. temperature of the day), and the past travel history of the trip-maker (e.g. the number of trips made on

the previous day). The individual features are denoted as v_1, v_2, etc. Features selected in this study will be introduced in Section V-B. The label, y, represents the state of the trip: 1 denotes a travelling instance, and 0 denotes the non-travelling instance.

$$x = (v_1, v_2, v_3, \cdots) \tag{2}$$

$$y = \begin{cases} 1, & \text{travelling} \\ 0, & \text{non-travelling.} \end{cases} \tag{3}$$

Since the non-travelling instances are much more frequent than travelling instances, the dataset is therefore 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, compared to over two million non-travelling instances in a day. The ratio of minority (travelling) class to majority (non-travelling) class is 1:44, 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 for the sake of achieving a good fit and significantly decrease the accuracy of the machine learning model. In Section IV, we introduce a method based on deep generative adversarial nets to produce a more balanced dataset for the prediction of hourly bus boarding demand.

B. Data pre-processing to prepare an original dataset

The smart-card data used in this study includes time-stamped boarding records along with the user ID. 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 for a single boarding. In real-life situations, some passengers may (accidentally) swipe their cards more than once when boarding, which causes two or more transactions during a short period. We consider these data as repetitive data and retain only the first-appearance record.
- The number of bus trips made by each smart-card user during a day is limited to a maximum value. 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. For IDs that appeared more than 19 times a day (i.e. more than one trip an hour), we consider them as testing smart-cards from the bus company and remove them from the database.
- This study only focuses on the regular smart-card users who travel at least once a week. We exclude users who travelled less than five times during the one-month study period. This is to avoid the excess non-travelling instances associated with such infrequent users.

After the initial data cleaning, we transform the remaining smart-card data into instances described in Eq. (1) and label them according to Eq. (3). The cleaned and transformed smart-card dataset will then be combined (fused) with other supplementary data, to create the baseline dataset used in this study.

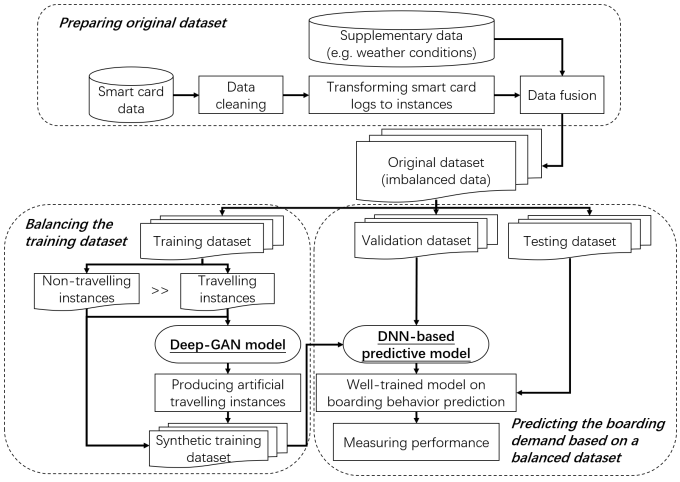

Fig. 2. The flow chart on predicting the boarding behaviour from an originally imbalanced dataset.

IV. SMART-CARD DATA IMPUTATION AND PASSENGER BOARDING DEMAND PREDICTION

This section introduces the methods developed to produce a more balanced dataset for the prediction of hourly passenger boarding demand. Figure 2 illustrates the processes involved. The pre-processed smarted card data are first transformed into travelling instances as described by Equations (1)-(3).

Additional supplementary data are added to the instances, and these include weather data that provides the impacts of weather conditions on the boarding behaviour. The combined data forms the baseline dataset for the study and is divided into three sub-datasets: training, validation and testing dataset.

As explained in Section III-A above, the original training dataset is severely imbalanced at a ratio of 1:44 between travelling and non-travelling instances. We will apply the deep generative adversarial nets (Deep-GAN) to generate artificial travelling instances and add them to the original training dataset to create a synthetic training dataset that is more balanced. The real travelling instances are data inputted to Deep-GAN, which then generates many artificial travelling instances that do not exist in the real world but have the same characteristics as real travelling instances.

The more-balanced synthetic training dataset will then be used to train a DNN-based predictive model to predict the state, travelling or not, and the validation dataset will be used to evaluate the performance of the training process. The architecture of the predictive model will be described in Section IV-B. Finally, we apply the testing dataset to the well-trained model to measure the performance of the model in Section IV-C.

A. Deep generative adversarial nets to balance the dataset

The generative adversarial nets (GAN), firstly introduced by Goodfellow *et al.* [48], is a deep learning architecture that consists of two multi-layer perceptrons (a generator and a discriminator). Figure 3 illustrates the basic architecture of the GAN model. The generator utilises a noise vector, which is made of random numbers, to produce synthetic data. The

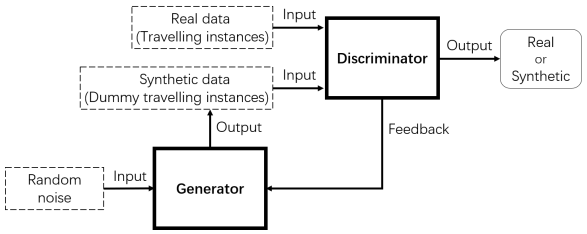

Fig. 3. The basic structure of the GAN [48].

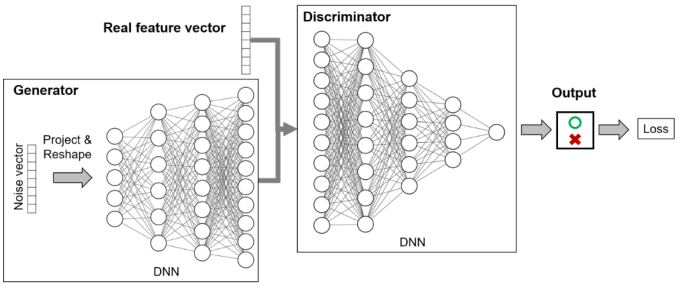

Fig. 4. The architecture of Deep-GAN. The number of layers and nodes are only examples. The basic framework is from [48].

synthetic data from the generator needs to imitate the real data in the same format and similar characteristics. In our study, the real data is the travelling data formulated in Section III-A, while the synthetic travelling data will be a feature vector with the same dimension and similar correlations among features. The discriminator tries to distinguish between real data and synthetic data by giving the full knowledge of real data. The generator learns from the feedback of the discriminator in order to generate more similar data.

The naïve GAN, as originally proposed by [48], applies two artificial neural networks (ANNs) as the generator and discriminator. Later studies have applied different neural networks of the generator and discriminator in GAN according to the characteristics of input data and the task of the algorithm. An example is the deep convolutional GAN which replaces ANN with two deep neural networks (DNN) to generate the image data [75]. Figure 4 illustrates the general framework of Deep-GAN. DNN will be introduced in the next subsection.

We define a discriminator D to identify if 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:

$$F_D = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \tag{4}$$

where $\mathbb{E}[\cdot]$ denotes the expectation. Maximising Eq. 4 means that D can correctly predict $D(x) = 1$ when x follows the probability density of real travelling data. That is to say, that the discriminator correctly labels the real travelling data, which is expressed as:

$$D(x) = 1, \qquad x \sim p_{data}(x) \tag{5}$$

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

$$F_G = \mathbb{E}_{z\sim p_z(x)}[\log(1 - D(G(x)))] \tag{6}$$

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:

$$\max_G \min_D V(G, D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(x)}\left[\log\left(1 - D\left(G(x)\right)\right)\right] \tag{7}$$

The process of training Deep-GAN is that D and G play a two-player minimax game with value function $V(D, G)$. D tries to maximise the value of V, which represents the best ability to discriminate between real and synthetic travelling instances. At the same time, G tries to minimise the value of V when the distribution of generated synthetic travelling instances is much closer to real travelling instances.

After the training process, the generator G can capture the distribution of real travelling instances and produce the synthetic travelling instances. An input random noise vector can be transferred and reshaped through the DNN in generator G and the output is a synthetic travelling instance with a similar data pattern of real travelling instances. The number of synthetic travelling instances depends on a given imbalance ratio (i.e. the ratio between travelling to non-travelling instances). Krawczyk [65] suggests that a good imbalance ratio for the training dataset is around 1:4, and in Section V-C we examine the relative performances of different imbalance ratios. Grouping with the real and artificial travelling instances and non-travelling instances, we can obtain a more balanced training dataset compared to the original dataset.

B. A deep neural network for predicting the boarding demand

As shown in Figure 2, Deep-GAN, described in the previous section IV-A, outputs a set of artificial travelling instances that do not really happen. A combination of artificial travelling instances from Deep-GAN and real travelling and non-travelling instances from the original dataset produces a more balanced dataset. Trained by this synthetic balanced dataset, we predict the boarding behaviour using a DNN-based predictive model. Compared to a simple ANN model, DNN has more hidden layers, as illustrated in Figure 5. As the number of features increases, a simple ANN model cannot capture the entire non-linear relationship among features. The DNN model, due to the larger network with more hidden layers and nodes, is able to describe the implicit and non-linear relationship and build up a complex model for the high-dimension input data [76].

A DNN model includes one input layer, one output layer and several hidden layers. In the input layer, there are several nodes. The feature vector x comes into the DNN model via the nodes in the input layer, where each node represents a feature v in x in Eq. (2). Several hidden layers follow the input layer, which are consists of some fully-connected nodes and one bias node. The values of the fully-connected nodes are calculated by:

$$a_j^l = \sigma\left(z_j^l\right) = \sigma\left(\sum_{i=1}^{I} \omega_{ij}^l a_j^{l-1} + b_j^{l-1}\right) \tag{8}$$

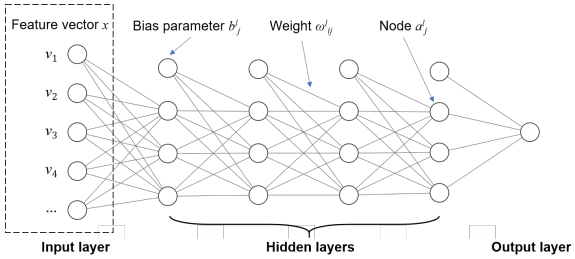

Fig. 5. An example of DNN's architecture [76].

Confusion matrix (CM)		Real boarding behaviour: Travelling instances (positive)	Real boarding behaviour: Non-travelling instances (negative)
Predicted boarding behaviour	Travelling instances (positive)	Ture-positive (TP)	False-positive (FP)
	Non-travelling instances (negative)	False-negative (FN)	Ture-negative (TN)

Fig. 6. Confusion matrix for binary classification of predicting travelling or non-travelling instances [79].

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 accumulating results from the nodes in the layer $l-1$ and the bias parameter b^{l-1}_j, and $\sigma(\cdot)$ 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. The output of this DNN is a binary (0-1) classification result of travelling or non-travelling instances.

C. Evaluation measurements

1) Evaluation the synthetic travelling instances: This section introduces a measurement, Fréchet distance (FD) [77], to examine the similarity of the synthetic travelling instances by over-sampling methods and the real travelling instances. Fréchet Inception Distance [78] is an important index to evaluate the generation performance of GAN, which uses an Inception V3 network to extract the features of images and calculation the FD value between the features of the real and produced images. However, the data in our case is structure (tabular) data, whose feature vectors are gained directly. Therefore, we adopt the FD value for evaluating the similarity of synthetic and real data. The FD value calculates the distance between two groups of instances in the feature space, which is formulated as:

$$FD(x,z) = \|\mu_x - \mu_z\|_2^2 + Tr\left[C_x + C_x - 2\left(C_x C_x\right)^{\frac{1}{2}}\right] \quad (9)$$

Where μ_x and μ_z are respective means of real travelling instances x and synthetic travelling instances z. C_x and C_z are the co-variance matrix of x and z, respectively. $Tr[\cdot]$ is the trace of matrix. A smaller value of FD represents that two groups of data are more similar.

2) Evaluating the prediction results: In previous sections, we apply our proposed Deep-GAN method, together with the DNN-based predictive model to predict the boarding actions of individual smart-card users at any hour of a day. To illustrate the performance of the prediction, this section introduces indices for measuring the direct predictive performance of DNN and the predicted hourly ridership of bus lines. Confusion matrix (CM) is one of the most used measurements for the classification problem [79]. The CM for the binary classification problem is shown in Figure 6. 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.

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 sides of the model, which are mutually constrained. An increase in the value of one index usually results in a decrease in the value of the others. Thus, the F-measure, the weighted harmonic mean of Recall and Precision, in Eq. (12) has been proposed in order to have a comprehensive consideration of precision and recall [80]. The parameter β (in Eq. (12a)) adjusts the weight of the focus of the model on precision and recall: with $\beta < 1$, , the F-measure gives more weight to the precision, while $\beta > 1$, more weight is given to the recall. The most common use is $\beta = 1$, which means the precision and recall are equally considered in this evaluation. We can obtain the most common and classic performance metrics, F1-measure in Eq. (12b), to evaluate the overall performance of machine learning models.

$$precision = \frac{TP}{TP + FP} \quad (10)$$

$$recall = \frac{TP}{TP + FN} \quad (11)$$

$$F_\beta = \frac{(\beta^2 + 1) \times precision \times recall}{\beta^2 (precision + recall)} \quad (12a)$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \quad (12b)$$

TABLE I
AN EXCERPT FROM SMART-CARD DATA.

COMPANY	LINE	VEHICLE	ENGINE	ID	TYPE	DATE	TIME
200	6	202111	159804	18725002	1	2016/8/1	11:43:15
200	6	202311	61502	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
…	…	…	…	…	…	…	…

Next, 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.

$$RMSPE = \sqrt{\frac{1}{H}\sum_{h=1}^{H}\left|\frac{\hat{Rider}_h - Rider_h}{Rider_h}\right|^2} \quad (13)$$

$$RMSE = \sqrt{\frac{1}{H}\sum_{h=1}^{H}\left|\hat{Rider}_h - Rider_h\right|^2} \quad (14)$$

where $\hat{Rider}_h$ 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.

V. CASE STUDY

A. The smart-card data source

The smart-card data used in this study records the trips made on seven bus lines from the bus network in the city of Changsha, China. The dataset covers the period from 1^{st} August to 1^{st} 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 III-B, 1,279,290 transactions from 101,850 smart-card IDs are retained. As shown in Table I, 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 smart-card data. In this study, we are concerned only with whether to travel or not; we do not consider (or estimate) the bus line and stops they used.

B. Feature selection

We choose the features from three domains: boarding time, weather conditions and travel history, because these three domains all have impacts on passengers' decision-making during bus trips [22], [81]. 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 II. 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 II shows the full list of features considered in this study.

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:

$$\grave{v} = \frac{v - v_{min}}{v_{max} - v_{min}} \quad (15)$$

where v is the value of a numerical feather in feature vector x and v_{min} and v_{max} respectively represent the minimum and maximum value and $\grave{v}$ is the value after min-max normalisation. 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 that can be made between categories. Here, we use One-hot Encoding to present a categorical feature as 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. A special categorical feature is the nominal feature of Card ID. The process of One-hot Encoding can generate sparse vectors with extremely high dimensions for this nominal feature. Thus, we use the feature hashing [82] to represent such categorical/nominal features. In total, there are 18 features and 49 dimensions in the feature vectors.

C. Experimental design

The resampling and prediction processes are conducted via Keras [83] in Python programming language. All the experiments are run on an Aliyun cloud graphics processing unit (GPU) platform with one 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 8^{th} to 31^{st} August as the combined training and validation dataset and 1^{st} September 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 preprocessing, it retains 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 are listed in Table III.

The resampling method is applied to the training dataset only. As the baseline (BL), the original data without any

TABLE II
INVESTIGATED DOMAIN OF FEATURES EMPLOYED IN MACHINE LEARNING MODELS.

Feature domains	Features	Dimensions	Feature types	Explanation
Boarding time	Season	4	Categorical	Spring; summer; autumn; winter.
	Day of the 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
	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 time slot on day-1	1	Numerical	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	Numerical	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	Numerical	Number of trips made by the passengers in the same time slot on all previous seven days

TABLE III
INVESTIGATED DOMAIN OF FEATURES EMPLOYED IN MACHINE LEARNING MODELS.

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
imbalance ratio	1:45	1:45	1:42
Total	37,154,880	9,288,720	1,935,150

TABLE IV
THE SYNTHETIC TRAINING DATASETS WITH DIFFERENT IMBALANCE RATIOS (BY DEEP-GAN).

Types of instances	BL (original)	$E_{1:20}$	$E_{1:10}$	$E_{1:5}$	$E_{1:2}$	$E_{1:1}$
Real travelling instances	819,278					
Synthetic data	0	997,502	2,814,282	6,447,842	17,348,523	35,516,324
Non-travelling	36,335,602					
imbalance ratio	1:45	1:20	1:10	1:05	1:02	1:01
Total	37,154,880	38,152,382	39,969,162	43,602,722	54,503,403	72,671,204

resampling method is directly used to predict the travelling behaviour. To analysis the impact of the balanced rate (minority to majority), we design a set of experiments, denoted as $E_{1:1}$, $E_{1:2}$, $E_{1:5}$, $E_{1:10}$ and $E_{1:20}$, where the subscript $1:m$ indicates the imbalance ratio. Table IV records the number of synthetic data and their imbalance ratio of training datasets where the synthetic data is generated by Deep-GAN.

In order to compare the different performance of Deep-GAN to other existing over- and under-sampling methods, we select two of the most commonly used over- and under-sampling methods, including the methods of random over-sampling (ROS), SMOTE, ADASYN, random under-sampling (RUS), ENN, k-means clustering and NearMiss-1 respectively. The imbalance ratio is decided by the best performance from the five variants of the synthetic data presented in Table IV. Here, we adopt an imbalance ratio of 1:5. The experiments with over-sampling methods (E_{ROS}, E_{SMOTE} and E_{ADASYN}) have the same size of training dataset with $E_{Deep-GAN}$,

TABLE V
THE SYNTHETIC TRAINING DATASETS BY DIFFERENT RESAMPLING METHODS (WITH IMBALANCED RATIO OF 1:5).

Experiments	Resampling methods		The number of instances		
	Groups	Methods	Travelling (minority)	Non-travelling (majority)	Total
BL (original)	None		819,278	36,335,602	37,154,880
$E_{Deep-GAN}$	Over-resampling	Deep-GAN	7,267,120	36,335,602	43,602,722
E_{ROS}		Random Over-Sampling			
E_{SMOTE}		SMOTE			
E_{ADASYN}		ADASYN			
E_{RUS}	Under-resampling	Random Under-Sampling	819,278	4,098,390	4,915,668
E_{ENN}		ENN			
$E_{k-means}$		k means clustering			
$E_{NearMiss}$		NearMiss-1			

and the experiments with under-sampling methods (E_{RUS}, E_{ENN}, $E_{k-means}$ and $E_{NearMiss}$ use all of the true travelling instances and part of non-travelling instances. The number of instances in the training dataset produced by under-sampling methods is much less than over-sampling methods. The detailed components of training datasets in these experiments are presented in Table V.

D. Model configuration

There are two DNNs in the Deep-GAN for generation and discrimination. Table VI displays the configurations of the generator and discriminator in Deep-GAN 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

TABLE VI
THE CONFIGURATIONS OF THE GENERATOR AND DISCRIMINATOR IN DEEP-GAN FOR $E_{1:20}$ TO $E_{1:1}$.

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 = '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'
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 VII
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'

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; the loss function is the binary_crossentropy function.

Table VII 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; the loss function is the binary_crossentropy function.

VI. 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 imbalance ratio on the accuracy of the predictive model the most. We then compare the performances of three different resampling methods using the best-balanced rate. After that, we discuss the prediction results on hourly demand.

A. Sensitivity analysis on imbalance ratio $1:m$

We use the same setting in our Deep-GAN to generate different training datasets with different imbalance ratios and apply these training datasets to the same predictive model. Figure 7 shows the performance metrics, on the precision, recall and F1 of the predictive models trained by the different training datasets with different imbalance ratios. The BL uses the original imbalanced training dataset of which the imbalance ratio is up to 1:44. In the cases of $E_{1:20}$, $E_{1:10}$, $E_{1:5}$, $E_{1:2}$ and $E_{1:1}$, we gradually increase the number of synthetic travelling instances in the training dataset, and hence reducing the imbalance ratio of the training dataset.

We can see from Figure 7 that 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 II-A earlier, the reasons for 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 ratios, the performance of the predictive models improves. Figure 7 shows that as the imbalance ratio in $E_{1:20}$, $E_{1:10}$, $E_{1:5}$, $E_{1:2}$ and $E_{1:1}$ reduces, the performance metrics increase. With an absolutely balanced training dataset ($E_{1:1}$), the values of all three metrics are over 0.88, suggesting that 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 a significantly longer time to train. We note in Figure 7 that,

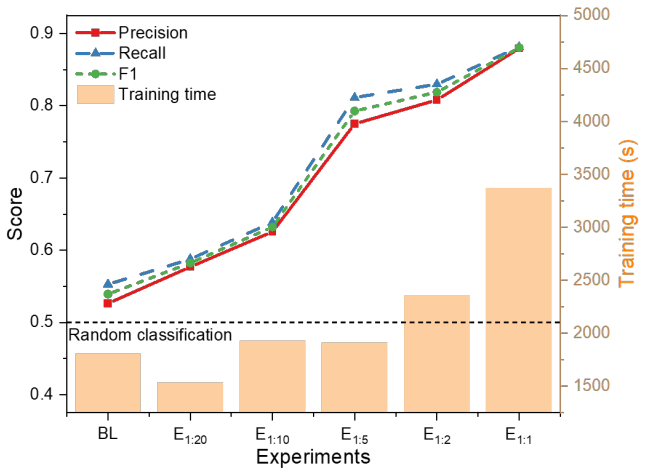

Fig. 7. The performance metrics (Precision, Recall and F1) and computation time for the training datasets with different rates of imbalance.

TABLE VIII
THE RMSPE AND RMSE OF HOURLY RIDERSHIP BY DIFFERENT IMBALANCE RATIO.

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

there is a significant improvement in the performance of the predictive models between $E_{1:10}$ and $E_{1:5}$, and the improvements in $E_{1:5}$, $E_{1:2}$ and $E_{1:1}$ are relatively small. On a balance of computational burden and model prediction accuracy, we consider $E_{1:5}$ with an imbalance ratio of 1:5 as an acceptable choice.

We can also see from Figure 7 that the 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 precision score.

For a further analysis of the sensitivity on imbalance ratio, we then analyse how the predicted results perform at the aggregated level of hourly demand with imbalance ratio $1:m$. Table VIII presents the RMSPE and RMSE of hourly ridership of the first group of experiments ($E_{1:20}$, $E_{1:10}$, $E_{1:5}$, $E_{1:2}$ and $E_{1:1}$) in Table IV. The result shows that from the view of hourly ridership $E_{1:5}$ has the lowest RMSPE and RMSE, although the measurements of precision, recall and F1 in $E_{1:5}$ is slightly lower (worse) than $E_{1:2}$ and $E_{1:1}$. We will analyse this situation with the help of the profile of hourly ridership in Section VI-C. Cooperating the results in Figure 7 and VIII, rebalancing the dataset with an imbalance ratio of 1:5 leads to a great performance of predictive model and the best accuracy in hourly ridership with a low computation time; and this finding agrees with that of Krawczyk [65].

B. Best-balanced method among resampling methods

Table IX shows the FD values between the real and synthetic travelling instances generated by different over-sampling methods. We also calculate the FD values between travelling and non-travelling instances, to check whether the synthetic data overlaps with the non-travelling data in the feature space. As shown in Table IX, the big FD value between real travelling and non-travelling instances shows a significant difference between these two kinds of instances. The synthetic travelling instances by different over-sampling methods are close to the real ones and far from the non-travelling instances, indicating that the selected over-sampling methods can produce synthetic travelling instances that have similar characteristics to the real ones. Since the instances by ROS are the repetition of the real data, the FD values are almost the same as the real travelling instances. The instances produced by SMOTE and ADASYN are more similar to the real data than by Deep-GAN. SMOTE and ADASYN over-sample the instances in the same feature space, which can lead to the risk of overemphasising a certain condition in feature space. Accordingly, it may result in a learning bias when training the following prediction model on those data. Overall, the Deep-GAN is able to produce synthetic travelling instances that are similar to the real instances and are significantly different from the non-travelling instances. In addition, the Deep-GAN ensures that there is a diversity in the synthetic data so that the subsequent prediction model does not over-fit some data characteristics.

TABLE IX
THE FD VALUES BETWEEN DIFFERENT DATA GROUPS BY DIFFERENT OVER-SAMPLING METHODS.

Synthetic travelling instances by method	The values of FD	
	Real travelling instances	Non-travelling instances
Real travelling instances	-	455.24
$E_{Deep-GAN}$	87.17	315.32
E_{ROS}	$<10^{-4}$	455.23
E_{SMOTE}	0.53	452.15
E_{ADASYN}	16.96	341.00

Next, we compare the performances of different resampling methods designed in Table V: Deep-GAN, ROS, SMOTE, ADASYN, RUS, ENN, k-means and NearMiss-1, in these experiments denoted $E_{Deep-GAN}$, E_{ROS}, E_{SMOTE}, E_{ADASYN}, E_{RUS}, E_{ENN}, $E_{k-means}$ and $E_{NearMiss}$. The same imbalance ratio (1:5) is applied. Figure 8 displays the performance metrics of the predictive models for these experiments. Overall, the prediction results with improved training data are much better than in BL, suggesting that the accuracy of the predictive model will be improved as long as the imbalance ratio can be reduced by any resampling method. Comparing over-sampling methods (Deep-GAN, ROS, SMOTE and ADASYN), $E_{Deep-GAN}$ produces more accurate predictions than other methods in all three performance metrics, suggesting that the synthetic training dataset produced by Deep-GAN more benefits the following prediction model than that by the other three over-sampling methods. E_{ROS} has the second top Recall value but the lowest Precision value. As the FD values (in Table IX) increase from E_{ROS} to E_{SMOTE} to E_{ADASYN} to $E_{Deep-GAN}$, their Precision and F1 values rise from 0.65 to 0.78 and from 0.72 to 0.79, respectively. It proves that enriching the diversity of synthetic training data would

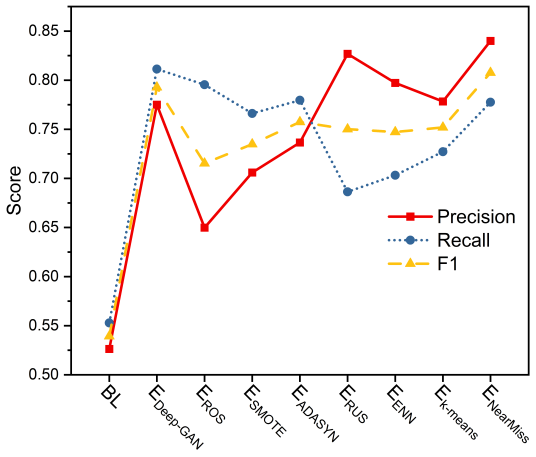

Fig. 8. The performance metrics (Precision, Recall and F1) for the training datasets generated by different resampling methods.

help the precision of the prediction model. Looking at the under-sampling methods (RUS, ENN, k-means clustering and NearMiss-1), the result shows an opposite performance to the over-sampling methods in that the precision values are greater than the recall values. According to the F1 values, E_{RUS}, E_{ENN} and $E_{k-means}$ have a better prediction ability than E_{ROS} and E_{SMOTE}, and $E_{NearMiss}$ has similar performance to $E_{Deep-GAN}$. E_{RUS} shows a strong power in the precision with a high value of 0.83. However, the recall of E_{RUS} only scores 0.69, which indicates that the learning bias is towards the non-travelling instances. It could be argued that the under-sampling methods produce a more reliable training dataset than most over-sampling methods. However, the proposed Deep-GAN significantly improve the quality of the synthetic training dataset and contributes to greater overall performance, especially the recall, of the prediction model. Therefore, Deep-GAN provides a sound method for over-sampling data in situations where, for example, the overall sample size and the sample size of the majority of data are small.

We note in Figure 8 that the precision scores of over-sampling methods are greater than their respective recall scores, while the opposite is true for under-sampling methods. The number of false-negative instances is less than false-positive ones in the datasets by over-sampling methods. E_{ROS}, E_{SMOTE}, E_{ADASYN} and $E_{Deep-GAN}$ tend more toward predicting to be positive (travelling instances). This is because the over-sampling methods artificially enhance the weight (number) of the travelling instances, so the models are more likely to predict actual non-travelling instances as positive. Accordingly, their precision value will be lower, such as E_{ROS}. However, improving the diversity of data, as the increasing FD values in Table IX, benefits for reducing the learning bias of the prediction model and improving the precision, where the Deep-GAN makes a great achievement. On the contrary, the under-sampling method deletes some non-travelling instances, which also reduces the information redundancy. Thus, the number of true-negative instances increases and the number of false-positive instances decreases, which contributes to the improvement of the precision score.

C. Results of hourly demand

In this section, we analyse the profiles of hourly ridership for answering the question at the end of Section VI-A, why there is an opposite conclusion of $E_{1:5}$, $E_{1:2}$ and $E_{1:1}$ in model performance and hourly demand. Figure 9 shows the profiles of hourly ridership observed from smart-card data (ground truth) and predicted by BL, $E_{1:20}$, $E_{1:10}$, $E_{1:5}$, $E_{1:2}$ and $E_{1:1}$. The observed ridership has two peaks: the morning peak at 7 to 9 am and the evening peak at 6 to 7 pm. The prediction based on BL (original imbalanced data) produced a delayed morning peak to 10 am, and a 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. In 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 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 Figure 6.

VII. Conclusion

The motivation of this study was because we have faced the challenge of imbalanced data when we used the real-world bus smart-card data to prediction the boarding behaviour of passengers at a time window. In this research, we proposed a Deep-GAN 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-based prediction model of individual boarding behaviour. The performance of Deep-GAN was 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 ratios in the training dataset, we found out that in general, the performance of the model improves with more imbalanced data and the most significant improvement comes at a 1:5 ratio between positive and negative instances. From the perspective of prediction accuracy of the hourly distribution of

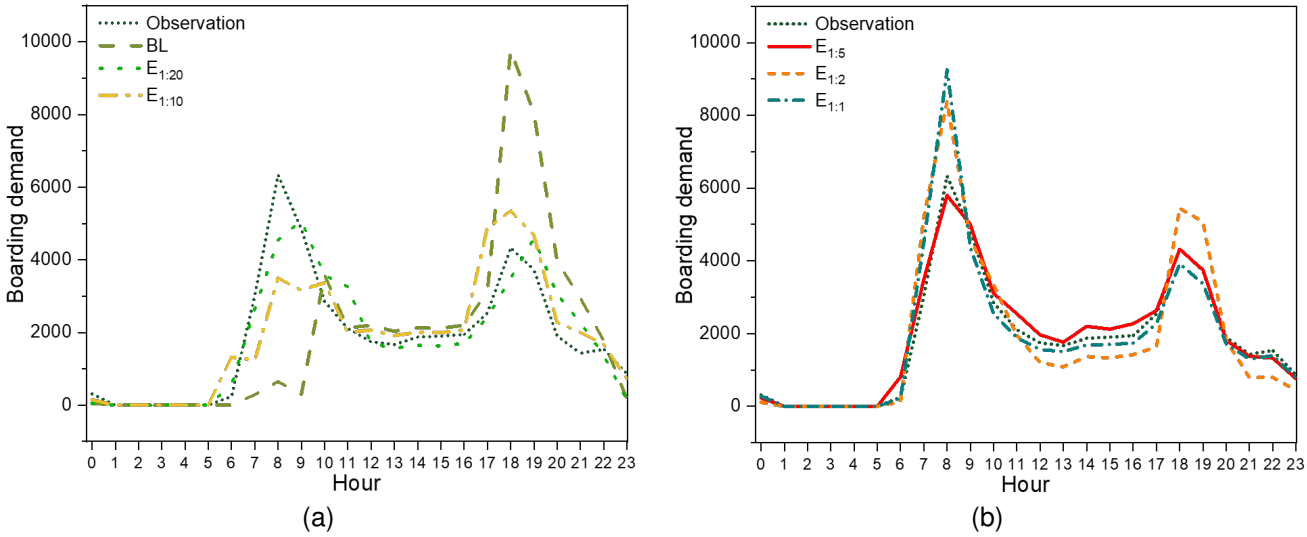

Fig. 9. The profile of hourly ridership observed from smart-card data and predicted by different synthesised datasets. (a) including BL, $E_{1:20}$ and $E_{1:20}$. (b) including $E_{1:5}$, $E_{1:2}$ and $E_{1:1}$.

bus ridership, the high rate of imbalance will cause misleading load profiles and the absolutely balanced data may over-predict the ridership during peak hours. Comparison of different resampling methods reveals that both over-sampling and under-sampling benefits the performance of the model. Deep-GAN has the best recall score and its precision scores best among the over-sampling methods. Although the performance of the predictive model trained by the Deep-GAN-data is not significantly beyond other resampling methods, the Deep-GAN also presented a powerful ability to improve the quality of training dataset and the performance of predictive models, especially when the under-sampling is not suitable for the data.

The contributions of this study are:

- The data imbalance issue in the public transport system has received little attention, and this study is the first to focus on this issue and propose a deep learning approach, Deep-GAN, to solve it.
- This study compared the differences in similarity and diversity between the real and synthetic travelling instanced generated from Deep-GAN and other over-sampling methods. It also compared different resampling methods for the improvement of data quality by evaluating the performance of the next travel behaviour prediction model. This is the first validation and evaluation of the performance of different data resampling methods based on real data in the public transport system.
- This paper innovatively modelled individual boarding behaviour, which is uncommon in other travel demand prediction tasks. Compared to the popular aggregated prediction, this individual-based model is able to provide more details on the passengers' behaviour, and the results will benefit the analysis of the similarities and heterogeneities.

As technology and computing power develop, predicting models will become more and more refined. In the field of demand prediction of the public transport systems, the target will gradually evolve from the bus network and bus lines to individual travel behaviour. This advancement can greatly benefit public transport planning and management, such as the digital twin of the public transport system. It is foreseeable that future prediction work in public transport systems will also encounter the challenge of imbalanced data. Our research proposes a Deep-GAN model to address the data imbalance issue in travel behaviour prediction. The validation via real-world data illustrated that the Deep-GAN showed a better ability to deal with the data imbalance issue and benefits the predictive models compared to other resampling methods. This research provides valuable experience for more researchers and managers in dealing with similar data imbalance issues, especially in public transport.

It may be noted that despite the great performance of Deep-GAN and DNN models, there are still some limitations. First, in this research, Deep-GAN 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. Second, this study makes the prediction at the individual level, which creates an explosion of information and makes the computation more difficult. Classifying the passengers (using clustering methods for instance) may be useful in terms of reducing the size of the dataset. Third, the current Deep-GAN does not consider the spatio-temporal characteristics of boarding behaviour. Customising the networks of generator and discriminator in GAN based on the characteristics of the boarding behaviour will further improve the quality of generated dummy travelling instances and the performance of the following predictive models. Finally, the proposed Deep-GAN selected the features and variants of the data augmentation independently. So, the improvements are likely to be sub-optimal. Jointly selecting the features and the optimum

imbalance ratio is likely to result in further improvements but at the cost of computational complexity. This can be tested in future. Similarly, the optimum rate of imbalance for Deep-GAN has been assumed to be the optimum rate for other resampling methods. This assumption needs to be tested in future research.

Even in its current form, this 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 or transfer behaviour, for instance.

REFERENCES

[1] X. Guo, J. Wu, H. Sun, R. Liu, and Z. Gao, "Timetable coordination of first trains in urban railway network: A case study of beijing," *Applied Mathematical Modelling*, vol. 40, no. 17, pp. 8048–8066, 2016.

[2] W. Wu, P. Li, R. Liu, W. Jin, B. Yao, Y. Xie, and C. Ma, "Predicting peak load of bus routes with supply optimization and scaled shepard interpolation: A newsvendor model," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, p. 102041, 2020.

[3] N. Bešinović, L. De Donato, F. Flammini, R. M. Goverde, Z. Lin, R. Liu, S. Marrone, R. Nardone, T. Tang, and V. Vittorini, "Artificial intelligence in railway transport: Taxonomy, regulations and applications," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

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

[5] Y. Wang, W. Zhang, T. Tang, D. Wang, and Z. Liu, "Bus od matrix reconstruction based on clustering wi-fi probe data," *Transportmetrica B: Transport Dynamics*, pp. 1–16, 2021, doi: 10.1080/21680566.2021.1956388.

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

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

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

[9] D. Chen, Q. Shao, Z. Liu, W. Yu, and C. L. P. Chen, "Ridesourcing behavior analysis and prediction: A network perspective," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[10] E. Nelson and N. Sadowsky, "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*, vol. 19, no. 1, p. 20180151, 2019.

[11] Z. Chen, K. Liu, J. Wang, and T. Yamamoto, "H-convlstm-based bagging learning approach for ride-hailing demand prediction considering imbalance problems and sparse uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 140, p. 103709, 2022.

[12] R. Liu and S. Sinha, "Modelling urban bus service and passenger reliability," 2007.

[13] J. A. Sorratini, R. Liu, and S. Sinha, "Assessing bus transport teliability using micro-simulation," *Transportation Planning and Technology*, vol. 31, no. 3, pp. 303–324, 2008.

[14] Y. Wang, W. Zhang, T. Tang, D. Wang, and Z. Liu, "Bus od matrix reconstruction based on clustering wi-fi probe data," *Transportmetrica B: Transport Dynamics*, pp. 1–16, 2021.

[15] Y. Hollander and R. Liu, "Estimation of the distribution of travel times by repeated simulation," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 2, pp. 212–231, 2008.

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

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

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

[19] S. Zhong and D. J. Sun, *A Spatio-temporal Distribution Model for Determining Origin–Destination Demand from Multisource Data*. Springer, Singapore, 2022, pp. 33–52.

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

[21] B. Chidlovskii, "Mining smart card data for travellers' mini activities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3676–3685, 2018.

[22] T. Tang, R. Liu, and C. Choudhury, "Incorporating weather conditions and travel history in estimating the alighting bus stops from smart card data," *Sustainable Cities and Society*, vol. 53, p. 101927, 2020.

[23] X. Zhang, Q. Zhang, T. Sun, Y. Zou, and H. Chen, "Evaluation of urban public transport priority performance based on the improved topsis method: A case study of wuhan," *Sustainable Cities and Society*, vol. 43, pp. 357–365, 2018.

[24] F. Chen, Z. Yin, Y. Ye, and D. Sun, "Taxi hailing choice behavior and economic benefit analysis of emission reduction based on multi-mode travel big data," *Transport Policy*, vol. 97, pp. 73–84, 2020.

[25] D. J. Sun, Y. Zheng, and R. Duan, "Energy consumption simulation and economic benefit analysis for urban electric commercial-vehicles," *Transportation Research Part D: Transport and Environment*, vol. 101, p. 103083, 2021.

[26] Y. Sun, J. Shi, and P. M. Schonfeld, "Identifying passenger flow characteristics and evaluating travel time reliability by visualizing afc data: a case study of shanghai metro," *Public Transport*, vol. 8, no. 3, pp. 341–363, 2016.

[27] Y. Yang, A. Heppenstall, A. Turner, and A. Comber, "Who, where, why and when? using smart card and social media data to understand urban mobility," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, p. 271, 2019.

[28] Y. Liu, C. Lyu, X. Liu, and Z. Liu, "Automatic feature engineering for bus passenger flow prediction based on modular convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2349–2358, 2021.

[29] Y. Zuo, X. Fu, Z. Liu, and D. Huang, "Short-term forecasts on individual accessibility in bus system based on neural network model," *Journal of Transport Geography*, vol. 93, p. 103075, 2021.

[30] T. Tang, A. Fonzone, R. Liu, and C. Choudhury, "Multi-stage deep learning approaches to predict boarding behaviour of bus passengers," *Sustainable Cities and Society*, vol. 73, p. 103111, 2021.

[31] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem," *International Journal of Advanced Computer Application*, vol. 5, no. 3, pp. 1–30, 2013.

[32] Y. LeCun, C. Cortes, and C. J. C. Burges, "Mnist handwritten digit database," 2010.

[33] D. Dua and C. Graff, "Uci machine learning repository," 2017.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[35] A. Azaria, A. Richardson, S. Kraus, and V. S. Subrahmanian, "Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 2, pp. 135–155, 2014.

[36] X. Gao, Z. Chen, S. Tang, Y. Zhang, and J. Li, "Adaptive weighted imbalance learning with application to abnormal activity recognition," *Neurocomputing*, vol. 173, 2016.

[37] C. Beyan and R. Fisher, "Classifying imbalanced data sets using similarity based hierarchical decomposition," *Pattern Recognition*, vol. 48, no. 5, pp. 1653–1672, 2015.

[38] M. Denil and T. Trappenberg, "Overlap versus imbalance," pp. 220–231, 2010.

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

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

[41] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations Newsletter*, vol. 6, no. 1, p. 20–29, 2004.

[42] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[43] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-smote: A new over-sampling method in imbalanced sata sets learning," in *ICIC 2005: Advances in Intelligent Computing*, ser. Advances in Intelligent Computing. Springer Berlin Heidelberg, 2005, Conference Proceedings, pp. 878–887.

[44] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, Conference Proceedings, pp. 1322–1328.

[45] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, p. 40–49, 2004.

[46] P. Shamsolmoali, M. Zareapoor, L. Shen, A. H. Sadka, and J. Yang, "Imbalanced data learning by minority class augmentation using capsule adversarial networks," *Neurocomputing*, vol. 459, pp. 481–493, 2021.

[47] X. Jiang and Z. Ge, "Data augmentation classifier for imbalanced fault classification," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1206–1217, 2021.

[48] I. Goodfellow, J. Pouget Abadie, M. Mirza, B. Xu, D. Warde Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS 2014)*, vol. 27, 2014, Conference Proceedings, pp. 1–9.

[49] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, 1972.

[50] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 6, pp. 448–452, 1976.

[51] I. Mani and I. Zhang, "Knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126. ICML United States, 2003, Conference Proceedings.

[52] S. J. Yen and Y. S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications*, vol. 36, no. 3, Part 1, pp. 5718–5727, 2009.

[53] Y. Zhang, L. Zhang, and Y. Wang, "Cluster-based majority under-sampling approaches for class imbalance learning," in *2010 2nd IEEE International Conference on Information and Financial Engineering*, 2010, Conference Proceedings, pp. 400–404.

[54] T. Liu, "Easyensemble and feature selection for imbalance data sets," in *2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing*, 2009, Conference Proceedings, pp. 517–520.

[55] X. Liu, J. Wu, and Z. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2009.

[56] J. Ha and J. S. Lee, "A new under-sampling method using genetic algorithm for imbalanced data classification," p. Article 95, 2016.

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

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

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

[60] J. Song, X. Huang, S. Qin, and Q. Song, "A bi-directional sampling based on k-means method for imbalance text classification," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 2016, Conference Proceedings, pp. 1–5.

[61] A. More, "Survey of resampling techniques for improving classification performance in unbalanced datasets," *arXiv e-prints*, p. arXiv:1608.06048, 2016.

[62] S. Gazzah, A. Hechkel, and N. E. B. Amara, "A hybrid sampling method for imbalanced data," in *2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15)*, 2015, Conference Proceedings, pp. 1–6.

[63] Z. Liu, D. Tang, Y. Cai, R. Wang, and F. Chen, "A hybrid method based on ensemble welm for handling multi class imbalance in cancer microarray data," *Neurocomputing*, vol. 266, pp. 641–650, 2017.

[64] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "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*, vol. 33, no. 2, pp. 245–265, 2012.

[65] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[66] S. H. Park and Y. G. Ha, "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*. IEEE, 2014, Conference Proceedings, pp. 45–49.

[67] A. B. Parsa, H. Taghipour, S. Derrible, and A. Mohammadian, "Real-time accident detection: Coping with imbalanced data," *Accident Analysis & Prevention*, vol. 129, pp. 202–210, 2019, (Kouros).

[68] S. Sharifirad, A. Nazari, and M. Ghatee, "An enhanced smote algorithm using entropy and clustering for imbalanced accident data," 2014.

[69] Q. Cai, M. Abdel Aty, J. Yuan, J. Lee, and Y. Wu, "Real-time crash prediction on expressways using deep generative models," *Transportation Research Part C: Emerging Technologies*, vol. 117, p. 102697, 2020.

[70] S. Dabiri, N. Marković, K. Heaslip, and C. K. Reddy, "A deep convolutional neural network based approach for vehicle classification using large-scale gps trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 116, p. 102644, 2020.

[71] R. Low, L. Cheah, and L. You, "Commercial vehicle activity prediction with imbalanced class distribution using a hybrid sampling and gradient boosting approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1401–1410, 2021.

[72] S. Hajizadeh, A. Núñez, and D. M. J. Tax, "Semi-supervised rail defect detection from imbalanced image data," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 78–83, 2016.

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

[74] M. S. Rahaman, M. Hamilton, and F. D. Salim, "Predicting imbalanced taxi and passenger queue contexts in airport," in *PACIS*, 2017, Conference Proceedings, p. 172.

[75] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[76] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–7, 2006, 1095-9203 Hinton, G E Salakhutdinov, R R Journal Article United States 2006/07/29 Science. 2006 Jul 28;313(5786):504-7. doi: 10.1126/science.1127647.

[77] D. Dowson and B. Landau, "The fréchet distance between multivariate normal distributions," *Journal of multivariate analysis*, vol. 12, no. 3, pp. 450–455, 1982.

[78] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[79] A. Luque, A. Carrasco, A. Martín, and A. de Las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition*, vol. 91, pp. 216–231, 2019.

[80] D. M. Powers, "What the f-measure doesn't measure: Features, flaws, fallacies and fixes," *arXiv preprint arXiv:1503.06410*, 2015.

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

[82] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, "Feature hashing for large scale multitask learning," p. 1113–1120, 2009.

[83] F. Chollet and Others, "Keras," 2015.

Tianli Tang is a post-doctoral research fellow at the School of Transportation, Southeast University. He received his Ph.D. in transport studies from University of Leeds in 2021. His current research focuses on data mining and deep learning on transport data, real-time monitoring on public transport system, and dynamic scheduling and management of public transport.

Ronghui Liu received her BSc degree from Peking University, China and PhD from Cambridge University, UK. She is the Professor in Networks and Transport Operations at the Institute for Transport Studies, University of Leeds. Her main research interests lie in developing mathematical, simulation and optimisation models to analyse the dynamic and complex interplays among policy instruments, operational controls and travellers' behavioural responses in transportation networks.

Charisma Choudhury is a Professor at the Institute for Transport Studies and School of Civil Engineering at the University of Leeds (UoL) where she leads the Choice Modelling Research Group. Charisma holds a PhD and MSc from Massachusetts Institute of Technology (MIT). She is an Honorary Guest Professor of Beijing Jiaotong University, China and a Turing Fellow of The Alan Turing Institute, London, UK. Her research interests include behaviour modelling and discrete choice analysis, transport modelling using big data sources, transportation in developing countries, traffic microsimulation.

Achille Fonzone is Professor of Transport Analysis and Planning at Edinburgh Napier University. He has a background in Civil Engineering, a PhD in Transport and Planning, a Post Graduate Certificate in Teaching and Learning in Higher Education. Achille's current research is in the area of travel attitudes and behaviour in relation to smart and sustainable mobility. His research aims to promote a more equitable, cleaner, and safer mobility, making the most of the opportunities generated by automation, data and connectivity, and new means of transport and business models.

Yuanyuan Wang was born in Cangzhou, Hebei, China in 1985. She received the B.S. and ph.D. degrees in transportation planning and management from Southwest Jiaotong University, Chengdu, China, in 2008, in 2013, respectively. She ever studied in Institute for Railway Engineering and Traffic Safety, Technical University of Braunschweig, Germany, in 2011-2013. Currently,She is working in School of Business Administration, Zhejiang University of Finance and Economics. Her research interests include public transport operation, transport network resilience and complex network.