# What influences passenger's arrival rate at stops in Melbourne: Wasea-Lstm, a novel deep learning model leveraging multi-source data fusion 

Mustafa Rezazada ${ }^{1}$, Neema Nassir ${ }^{2}$, Egemen Tanin ${ }^{3}$<br>${ }^{1,2,3}$ The University of Melbourne, Australia<br>mustafa.rezazada@student.unimelb.edu.au

Neema.nassir@unimelb.edu.au
etanin@unimelb.edu.au


#### Abstract

Public transportation demand plays a crucial role in service planning and operation. Accurate prediction of passenger arrival rates at transit stops allows transportation planners and operators to optimize resources and improve service efficiency. Current methodologies primarily focus on weather's impact in the aviation industry, supply dynamics, and arrival time prediction, while overlooking its influence on public transport demand variation. This study addresses these gaps by designing a deep neural network model that can predict public transit demand, using large-scale datasets from multiple sources in Melbourne, Australia. We propose a novel deep learning architecture called Wasea-Lstm (Weather-Aware Smart Exponential Activation LSTM) that captures spatial, temporal, and external correlations for passenger arrival rate prediction at tram stops. The model is trained and tested on integrated datasets from automatic fare collection (AFC), automatic passenger count (APC), and weather data over a period of three months. Results show that the Wasea-Lstm model significantly outperforms benchmark models, including gradient boosting machine (GBMR) and multi-layer perceptron (MLP) regression by $15 \%$ and $6 \%$ in $R^{2}$ metric, respectively. The feature importance ranking reveals that stop location, time of the day, temperature, and humidity are the key influencers of passenger arrival behaviour in Melbourne. Overall, this study contributes to the development of a model that accounts for multi-dimensional, high-resolution determinants of passenger demand using large-scale datasets from real world. The proposed Wasea-Lstm architecture shows exceptional performance in precisely forecasting stop-level demand for one of Melbourne's largest tram routes. Moreover, its applicability extends seamlessly to all routes within the network.


## 1. Introduction

Demand is a crucial factor in public transportation planning and operation. It can lead to reliability issues such as bunching and exhibits significant variability across space and time (Rezazada, Nassir and Egemen 2022). Passenger demand is a key metric that determines service types, frequency and schedule, the type, size, and number of in-service vehicles, synchronization and connectivity, network design, and service reliability indicators. (Ceder 2007) succinctly explains that demand rises when public transit service is perceived as a delicious food. The passenger arrival rate at public transit stops is an essential measure of passenger demand, enabling transportation planners and operators to optimize resources, improve service efficiency, and enhance the passenger experience. Due to the unpredictable nature of urban environments, demand frequently fluctuates during different times of the day, days of the week, seasons of the year, types of land-use, stop locations, and the proximity of
attractions, parks, and shopping centers (Rezazada, Nassir and Egemen 2022). In short, the more unpredictable transport demand becomes, the more we rely on prediction (Ceder 2007).
Accurate predictions require a deep understanding of the sources of variability, which can be caused by casual uncertainties, such as protests or accidents, or by systematic uncertainties associated with service dynamics (Soza-Parra, Raveau and Muñoz 2021). Several studies have explored different uncertainties that influence demand change and variability, including public transit fares, frequency, and station characteristics (Ingvardson et al. 2018; Toro-González, Cantillo and Cantillo-García 2020), network and time of day (Ahn et al. 2016; Frumin and Zhao 2012), passenger crowding (Hensher 2020), on-time vehicle performance (Mai, List and Hranac 2012), timetable dependency (Zhang, Chen and Han 2014), pandemics like COVID-19 (Downey et al. 2022), and socioeconomic background and fare elasticity (Kholodov et al. 2021). The presence of spatio-temporal and directional demand imbalances in public transport networks is unavoidable, and it can be very difficult to identify and predict casual uncertainties (Hörcher and Graham 2018). However, systematic and seasonal behaviors can be identified using emerging technologies such as machine and deep learning models and real-world data sensing and automation on a large scale.
Numerous analytical, statistical, optimization, simulation, machine learning, and deep neural network methods are utilized in the literature to formulate and model various aspects of public transport networks and operations. The growth of technology, the availability of large datasets from real-world operations using automated sensors and devices, the development of high computational power and GPUs, and rapid advancements in artificial intelligence and deep neural networks, all present new potential. These factors now make it possible to study complex dynamics and a significant number of parameters using multi-source data integration at high scale and resolution with comparatively minimal computation time and cost. However, to the best of the authors' knowledge, a comprehensive model that can capture demand non-linearity and variability across networks and predict in real-time using multi-dimensional longitudinal datasets from automatic data fusion is absent. Specifically, adding weather as an explanatory variable that can influence passengers' arrival at public transport stops or stations is often overlooked in public transportation. For instance, (Smith and Sherry 2008) modeled the influence of weather on aircraft arrival rates using Support Vector Machines, while (Venkatesh et al. 2017) deployed neural networks and deep learning to predict flight arrival delays, considering weather input. (Ke et al. 2017) developed a fusion convolutional long short-term memory network (FCL-Net) that captures three different dependencies in demand estimation for on-demand ride services, including spatial, temporal, and exogenous factors. Such implementation is absent in public transport demand estimation modeling, although it is challenging due to the complex nature of public transport systems, network and stop alignment and distributions, demand-supply sensitivity, and capacity compared to on-demand ride-hailing and taxi services.

Various researchers have proposed different methodologies and models that consider weather and other relevant parameters to predict bus/train arrival times and feeder-bus operations (Arshad and Ahmed 2021; Bao, Zhang and Shi 2020; Liu et al. 2022; Yang et al. 2016). The existing methodologies primarily focus on the impact of weather in the aviation industry, supply dynamics, and arrival time prediction, while overlooking its influence on public transport demand variation. Similar studies, such as (Ke et al. 2017), have yet to address weather and spatio-temporal correlations, as well as the clear distinction between temporal time series and spatio-temporal dependencies in public transport demand estimation. Additionally, the literature currently lacks a proper model that can capture spatial and temporal demand nonlinearity with temperature explanatory variables. The limitations of the existing literature can be categorized as follows:

- Mainly focused on aviation and lacking studies for ground traffic and public transport
- Primarily studied supply dynamics and their associated aspects
- Validated using synthesized data and lacking high-quality, large-scale data from the real world
- Evaluated using a single data source or data available for a short period, which cannot capture seasonal variation

To address these gaps, this research aims to design a deep neural network that can predict public transit demand in real-time and is applicable to both route and network levels. With the potential and availability of automatic data sources, this study contributes to the development of a model that accounts for multi-dimensional, high-resolution determinants of passenger demand using large-scale datasets from the real world. Furthermore, the proposed method is trained and tested on integrated datasets from multiple sources, including automatic fare collection (AFC), automatic passenger count (APC), and weather data in Melbourne, Australia, over a period of three months. Finally, the suggested architecture for the LSTM model, built on a deep neural network, can accurately capture the non-linearity of demand across time and space.
The remaining sections are organized as follows: Section 2 explains the methodology, Section 3 describes the case study in this research, and Sections 4 and 5 present the results and conclusions, respectively.

## 2. Methodology

### 2.1. Overview

In this paper, we present an innovative approach called Wasea-Lstm (Weather-Aware Smart Exponential Activation LSTM Estimator) that captures spatial, temporal, and external dependencies for passenger arrival rate prediction at tram stops. Contrary to the existing literature methodologies, our research predicts arrival rates for potentially large dimensions with high spatio-temporal resolutions. We tested various machine and deep learning models; however, the proposed deep learning architecture outperforms the benchmarks in all metrics. In this section, we first briefly introduce gradient boosting machine, multi-layer perceptron, and then the designed architecture for LSTM model training and evaluation. We also propose a simple algorithm for calculating relative feature importance based on Friedman's work in 2001, which ranks the contribution of each input variable in predicting the target variable.

### 2.2. Model architecture

### 2.2.1. Gradient Boosting Machine Regression (GBMR)

Gradient descent (boosting) is a popular machine learning strategy with numerous applications. Gradient boosting of regression trees (GBRT) generates highly competitive, robust, and easily understandable methods for both regression and classification, making it particularly wellsuited for analyzing imperfect data (Friedman 2001). Gradient boosting regressors are additive models, formulated as the prediction $\hat{Y}_{i}$ for given input $X_{i}$ using a series of weak learners $U_{j}$ (Friedman 2001), see Equation 1. $F_{J}\left(X_{i}\right)$ in this notation denotes that the prediction estimates $\widehat{Y}_{i}$ conditioned both on $J$ number of weak learners and $X_{i}$ set of inputs.
$\hat{Y}_{i}=F_{J}\left(X_{i}\right)=\sum_{j=1}^{J} U_{j}\left(X_{i}\right)$
Equation 2 employs a greedy algorithm utilizing a fixed number of weak learners $U_{j}$, in the gradient boosting tree.

$$
\begin{equation*}
F_{j}(X)=F_{j-1}(X)+U_{j}(X) \tag{2}
\end{equation*}
$$

The new term $U_{j}(X)$, on the right side must be fitted to minimize the total losses represented by the loss function $L_{j}$, given the prior approximation $F_{j-1}(X)$, as shown in relation 3 .

$$
\begin{equation*}
U_{j}(X)=\arg \min L_{j}=\arg \min _{h} \sum_{i=1}^{n} l\left(Y_{i}, F_{j-1}\left(X_{i}\right)+U\left(X_{i}\right)\right) \tag{3}
\end{equation*}
$$

The initial term on the right side represents the loss parameter, which can be measured using various metrics like squared error, absolute error, or Huber. The scikit-learn version 1.2.1's GBMR is utilized to train the model over appropriately preprocessed datasets to use in boosting machine learning algorithms.

### 2.2.2. Feature selection using relative importance

There are always non-homogenous influences from different input variables $X_{i}=\left\{X_{1}, \ldots, X_{n}\right\}$ on output variable(s) $Y_{i}$. To improve the computational efficiency of the model and identify the most relevant and important features in approximation problems, particularly in cases with many explanatory variables, the relative influence $I_{j}$ of individual input variables $X_{i}$ on the variation of $Y_{i}$ is among the most powerful tools for interpreting approximation $\widehat{F}(X)$ (Friedman 2001). Several methodologies calculate the relative importance of individual input variables on the prediction of output variables, such as Gini importance or Mean Decrease Impurity (MDI) and Mean Decrease Accuracy (MDA), which is also known as Permutation importance. This research develops an algorithm based on the relative importance proposed in the work by (Friedman 2001) (See 4).
$I_{j}=\sqrt{E_{x}\left[\frac{\partial \hat{F}(X)}{\partial X_{j}}\right]^{2} \operatorname{var}\left[X_{j}\right]}$
The left term $\left(I_{j}\right)$ represents the relative influence of feature j , while the right term approximates $\hat{F}(X)$ conditioned on $X$. The function is approximated by a surrogate measure because the solution for decision trees does not strictly exist. (Breiman 1984) proposed relation 5, which takes the summation over each node $t$ of the $J$-node tree $T$, the corresponding variable associated with node $t$, and the improvement in squared error at the end of the split at node $t$.
$\hat{I}_{j}^{2}=\sum_{t=1}^{J-1} \hat{\imath}_{j}^{2}\left(m_{t}=j\right)$
$m_{t}=j$, is an indicator function that returns 1 if feature $j$ is used for splitting in the $t^{t h}$ tree and 0 otherwise. However, Breiman, Friedman, Olshen, and Stone (1983) used it directly rather than as squared influence (Friedman 2001). The proposed algorithm calculates the contribution of each feature to the reduction in the loss function when constructing decision trees. For each feature in the model, it computes the total reduction in impurity gained by splitting on that feature across all decision trees. The impurity measure used in the calculation is typically the Gini entropy, which assesses the homogeneity of the target variable in the subsets created by the split. Finally, the algorithm normalizes the total reduction in impurity for each feature by dividing it by the sum of the total reduction in impurity across all features. This produces the relative importance of each feature. The resulting feature importance scores are typically scaled so that they sum up to 1.0 .

## Algorithm: Relative importance calculation Inputs:

$X$ : input variables, matrix of training data
$Y$ : target variable of training data
$T$ : number of decision trees in Gradient Boosting Machine (GBM) model, which is trained in the previous section.

## Outputs:

Relative feature importance scores: array of feature importance scores, that ranks how each input variables contributed to the reduction of the loss function (Difference between ground truth $Y$ and predicted $\hat{Y}$ by GBM)

- Train a GBM model with $T$ decision trees to predict $Y$ from $X$, using scikit-learn 1.2.1
- For each feature $j$ (location, temp, others) compute the total reduction in the loss:


## for t in 1 to T :

if feature j is used for splitting in tree t :
compute reduction in loss function for tree $t$ using feature $j$
add reduction to total reduction, j

- Normalize the total reduction in the loss function for each feature feature_importance_scores $=$ total reduction / sum (total reduction)
- The resulting feature importance scores are typically scaled so that they sum up to 1.0 .
- Return feature_importance_scores.


### 2.2.3 Multi-layer perceptron network

Partial dependence plots (PDPs) are valuable tools in statistical learning. They can be employed to infer the interaction between target variables and input variables, such as linear or non-linear relationships (scikit-learn). Interactions among some key predictors can lead to higher error rates, which can be diagnosed through two-variable partial dependence plots (Hastie, Tibshirani and Friedman 2017). Plotting partial dependence to describe each predictor's contribution to the fitted GBMR model in the previous section revealed non-linearity and strong interactions between temperature and stop location in particular. As a result, a multilayer perceptron is introduced to capture this non-linearity and complex interactions among predictors.
A multi-layer perceptron (MLP) is a popular type of artificial neural network (ANN) composed of fully connected feedforward neurons or nodes. MLP consists of two or more layers, including an input layer, one or multiple hidden layers, and an output layer, each employing an activation function to map non-linearly separable data (Cybenko 1989). The MLP architecture presented in this study includes an input layer equal to the number of observations, two hidden layers with 100 and 50 neurons respectively, and a single-neuron output layer that estimates the number of passengers arriving at each stop every 30 minutes, using a linear activation function for regression problems. The two hidden layers in this model are designed to learn complex patterns and representations from the input data. Each neuron in the hidden layers receives a weighted sum of the outputs from the neurons in the previous layer, applies an activation function (ReLU in this case), and generates an output (Figure 1). The Rectified Linear Unit (ReLU) activation function is used to produce positive values, choosing the maximum value between zero and X .
$\operatorname{ReLU}(X)=\operatorname{argmax}\left\{\begin{array}{l}0 \\ X\end{array}\right\}$

The MLP model utilizes backpropagation for learning, an algorithm that minimizes error by adjusting the weights and biases of the network through each iteration, calculating the derivative of the loss function concerning the given parameters. In simple terms, fitting neural networks involves finding the best values for the unknown parameters called weights so that the model fits the training data well (Hastie, Tibshirani and Friedman 2017).


Figure 1: Schematic diagram of a multi-layer perceptron (MLP) with an input layer equal to the length of input variables, two hidden layers of 100 and 50 neurons, and a single neuron output layer.

### 2.2.4. Wasea-Lstm (Weather-Aware Smart Exponential Activation LSTM

Long short-term memory (LSTM) is a well-known class of recurrent neural networks (RNNs). Unlike standard feedforward neural networks such as MLP, LSTM can process sequential data and remember its state over time, which makes it ideal for time series and sequential information. LSTM has a wide range of applications in NLP, speech recognition, image captioning, as well as various domains in traffic and transportation, such as demand estimation, arrival time prediction, bunching detection, and others. For example, (Yao et al. 2018) proposed a framework to capture both temporal and spatial correlations in taxi demand estimation, and (Ke et al. 2017) modelled three distinct dependencies: temporal, spatial, and exogenous for on-demand ride services using LSTM. This study proposes an innovative deep neural network architecture called Wasea-Lstm (Weather-Aware Smart Exponential Activation LSTM Estimator) that incorporates spatial and temporal correlations with finer resolutions in time and space.
Time series data analysis has been challenging for years, and using variables that vary in time with those that change in space and/or both is very difficult. The presented methodology has the following advantages over the best state-of-the-art models:

- The proposed method incorporates spatial resolution at the stop level, which is much shorter with higher granular resolutions than the $7 \times 7$ grids, each grid with an approximate length of 4.77 kilometres, as presented in (Ke et al. 2017). In contrast, the stop locations are mostly within 0.2-1.0 kilometres, which produces more than 10 times finer spatial resolutions. As per findings from (Ke et al. 2017), Pearson correlations for demand intensity and travel time drop significantly from $27 \%$ at a grid distance of 1 kilometre to less than $2 \%$ as the grid distance increases to 9 kilometres. Therefore, the finer and higher spatial resolution in this research is expected to significantly enhance the model prediction and identification of non-linearity in complex interactions.
- The methodology in this research incorporates a temporal resolution of 30 minutes, which is twice as high as the best model with a 1-hour interval.
- Lastly, temporal variables, such as time of the day, are incorporated as numerical variables, which can better predict the outcome than categorizing them into aggregated forms such as peak and off-peak periods. Aggregating to a limited number of categories results in losing valuable information between each category and reducing the dimension and resolution.
Therefore, the proposed method is superior in accurately predicting stop-level demand for the tram network in Melbourne.
The majority of proposed methodologies that deal with spatio-temporal correlation in demand estimation modelling combine time series (temporal) and non-time series (spatial) variables into a single LSTM architecture. This architecture consists of LSTM and Convolutional layers for each, respectively. However, this integration makes it challenging to capture spatial correlation in smaller sizes with larger details. When transforming to convolutional layers, some essential interactions between features might be lost. Therefore, in this study, we propose a novel architecture (Wasea-Lstm) that sequentially incorporates both spatial and temporal attributes.
Wasea-Lstm is built using Keras 2.10.0. with a Sequential API, which allows stacking layers in a linear manner. The architecture consists of three layers: two LSTM layers and one Dense output layer. The first LSTM layer has 100 units and is configured to retain the sequences of the input information. This allows the output of this layer to be a sequence of hidden states, which can be fed as input to the second LSTM layer. An exponential activation function is used in the dense layer to strictly penalize negative outputs and preserve the non-linear relationship between input variables and target variables, as shown in Figure 2. Moreover, the mean squared logarithmic error (MSLE) is chosen instead of Mean Squared Error (MSE), as it is suitable for regression and non-negative target values with a large dynamic range, as demonstrated in relation 7 .

MSLE $=(1 / N) \sum\left(\log \left(Y_{i}\right)-\log \left(\widehat{Y}_{l}\right)\right)^{2}$
The model can be scaled to both route and network levels, and it is tested using integrated multi-dimensional large-scale data collected from tram operations in Melbourne over three months, aggregated to 30-minute intervals.

## 3. Case study

### 3.1. Overview of the case study

Melbourne, the capital of Victoria, is home to the world's largest operational tram network, boasting 250 kilometres of double track that facilitates over 200 million trips annually and more than 5,000 services per day (YarraTrams 2023a). The network has been operating for over 100 years, with more than three-quarters of its trams traveling in shared corridors with private vehicles, resulting in low average operating speeds of 15 kph (Currie, Goh and Sarvi 2012). The tram network operates 24 hours a day; however, limited routes are available during the night shift with longer headways. Despite a $0.3 \%$ annual decline in tram patronage between 2017-18 and 2018-19, it remains the second most used mode of public transport after metropolitan trains, with 205.4 million annual boardings (Victoria 2018-19). The proposed methodology in this paper has been trained and tested on tram route 96, which runs from East


Figure 2: Network diagram of Wasea-Lstm with an input layer equal to the length of input variables, two LSTM layers of 100 and 50 units, and a single neuron output layer with exponential activation function.

Brunswick to St Kilda Beach and includes a total of 39 stops, with ten stops located within the free tram zone in the CBD (YarraTrams 2023b); see Figure 3.
Considering multi-dimensional explanatory variables coming from different sources with spatial, temporal, endogenous, and exogenous public transport reliability influencing factors (Rezazada, Nassir and Egemen 2022), Route 96 is chosen that is assumed to have all these variability in order to effectively evaluate the performance of the model. The long route of tram line 96 connects diverse land-use areas and traverses various traffic corridors, featuring multiple attractions, different tracks (exclusive and shared segments), and significantly varying demand profiles, which can represent spatial heterogeneity. For example, some people may travel from northern suburbs to the CBD, while others commute from the CBD to southwest suburbs, resulting in a non-linear demand profile. Key destinations along the route include Melbourne Museum, Carlton Gardens, Royal Exhibition Buildings and IMAX, Bourke Street Mall, Crown Entertainment Complex, Melbourne Sports and Aquatic Centre, St Kilda Beach, and Luna Park and Palais Cinema. Consequently, demand is expected to vary significantly across different spaces and types of land-use, such as CBD, upstream, and downstream areas. Demand may also fluctuate throughout the day, days of the week, and various seasons due to differing event schedules at key attractions. Moreover, the weather data used in this research is obtained for city-wide Melbourne, and since route 96 covers a substantial part of the city, the locality bias of the weather is alleviated.


Figure 3: Tram route 96 East Brunswick to St Kilda Beach. Left figure, re-synthesized from tram network map produced by Public Transport Victoria, and right-hand figure is produced using Google map

### 3.2. Data collection and integration

In this study, we utilized several datasets from different sources, including automatic fare collection (specifically, Myki card data in Melbourne), automatic passenger count (collected during a trial period), and weather data obtained from the Visual Crossing Weather API. We used hourly weather data collected for Melbourne, Australia, during a three-month period from February 1st, 2020, to April 30th, 2020. The dataset was acquired from the Visual Crossing Weather API, which provides historical and real-time meteorological information with high resolution (API 2020). This dataset includes numerous key weather parameters, such as temperature, humidity, precipitation and its probability, wind speed, wind direction, visibility, and UV index, among others (Table 1). The inclusion of this granular weather information, recorded from multiple stations, is crucial for estimating the number of passengers arriving at tram stops. It is hypothesized that the likelihood of using a tram on a rainy day with strong winds may decrease. Similarly, the potential impact of weather attributes on leisure or shopping trips during weekends might be stronger than work trips on weekdays. Furthermore, location proximity to a tram stop can be associated with arrival behaviour; hence, carrying an umbrella can be burdensome for some trips. Therefore, the potential impacts of meteorological conditions on public transport demand profiles are essential. By examining the relationship between weather variables and arrival rates at tram stops, we aim to develop a model that enhances the accuracy of demand estimation under various weather conditions.
Myki data, an Automatic Fare Collection (AFC) system used in Melbourne Transportation services, includes metro, train, tram, and bus data. This study incorporates tram data from Myki cards, spanning a three-month period from February 1st, 2020, to April 30th, 2020. The dataset consists of $1,134,932$ transactions, including tap-on and tap-off events for passengers validating their Myki cards upon boarding and alighting. The data captures vital information such as transaction time (in epoch format), date, transaction type (tap-on or tap-off), unique transaction identifier, route number, card key, vehicle number, and stop name and location (refer to Table 2). Notably, around $5 \%$ of tap-off transactions are missing, which were imputed and filled in the dataset. Nonetheless, sensitivity analyses revealed that the model predictions were not significantly impacted by either imputing or excluding these missing transactions. Furthermore, the gathered data covers the initial lockdown phase implemented in Melbourne.

As a result, a reduction in the count of tram commuters is evident. The decision of whether to incorporate the lockdown period data or not led to the execution of numerous experiments. Despite the deep learning model's sensitivity to capture these variations, the ultimate iteration focused solely on the datasets recorded prior to the lockdown.
By integrating the Myki data with high-resolution weather data into our study, we aim to better understand and model the relationship between public transportation usage and various factors influencing it, ultimately planning, and operating reliable services that balance the trade-off between supply and demand. An algorithm has been developed to pre-process, filter, compute stop-level demand, and integrate data from different sources into an appropriate input format that can be fed to machine and deep learning models. In this algorithm, several data processing tasks are performed for stop-level demand estimation using Automatic Fare Collection (AFC), weather datasets, and Automatic Passenger Count (APC) data.

Table 1: Hourly weather information for Melbourne between February 1st, 2020, to April 30th, 2020, retrieved from multiple station located across the city using Visual Crossing Weather API in Python

| name | datetime | temp | feelslike | dew | humidity | preci* | precipprob | windspeed | winddir | sealevelpressure | visibility | uvindex | conditions | stations |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Melbo... | 2020-02-... | 15.8 | 15.8 | 8.9 | 63.77 | 0.009 | 100 | 16.8 | 152 | 1022 | 10 | 0 | Rain, Pa... | 95874099... |
| Melbo... | 2020-02-... | 11.8 | 11.8 | 8.6 | 80.51 | 0 | 0 | 4.1 | 180 | 1021.8 | 10 | 0 | Clear | 94866099... |
| Melbo... | 2020-02-... | 13.1 | 13.1 | 9.6 | 79.02 | 0 | 0 | 6.1 | 178 | 1023 | 10 | 0 | Partially cloudy | 95874099... |
| Melbo... | 2020-02-... | 11.7 | 11.7 | 9 | 83.8 | 0 | 0 | 7.5 | 191 | 1022 | 10 | 0 | Partially cloudy | 95874099... |
| Melbo... | 2020-02-... | 12.6 | 12.6 | 8.9 | 78.32 | 0 | 0 | 7.8 | 179 | 1022 | 10 | 0 | Partially cloudy | 95874099... |
| Melbo... | 2020-02-... | 12.9 | 12.9 | 9.4 | 79.36 | 0 | 0 | 7.9 | 188 | 1022.8 | 10 | 0 | Clear | 94866099... |
| Melbo... | 2020-02-... | 12.6 | 12.6 | 9.1 | 79.29 | 0 | 0 | 8 | 179 | 1022.4 | 10 | 0 | Partially cloudy | 95874099... |
| Melbo... | 2020-02-... | 14 | 14 | 8.9 | 71.35 | 0 | 0 | 8.1 | 178 | 1023 | 10 | 0 | Partially cloudy | 95874099... |
| Melbo... | 2020-02-... | 14.8 | 14.8 | 9.1 | 68.6 | 0 | 0 | 11.5 | 159 | 1022.9 | 10 | 0 | Clear | 94866099... |
| Melbo... | 2020-02-... | 15.2 | 15.2 | 8.6 | 64.83 | 0 | 0 | 11.6 | 160 | 1022.6 | 10 | 0 | Partially cloudy | 95874099... |
| Melbo... | 2020-02-... | 16.5 | 16.5 | 9.1 | 61.67 | 0 | 0 | 13.7 | 161 | 1021.3 | 10 | 0 | Partially cloudy | 94866099... |

Table 2: Myki datasets from tram operation in Melbourne between February 1st, 2020, to April 30th, 2020

| txn_uid | txn_type | route_number | card_key | txn_ts | vehicle_number | gtfs_stop_id | stop_name | stop_lat | stop_lon | operational_ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0-2448... | IMPUTED SCANOFF | 96 | 30680217 | 1582905976 | 6083 | 17894 | 11-Melbou... | -37.8057 | 144.974 | 2020-02-28 |
| 0-2448... | SCANON | 96 | 30682889 | 1582885494 | 5111 | 6070 | 21-Glenly... | -37.7734 | 144.979 | 2020-02-28 |
| 0-2448... | IMPUTED <br> SCANOFF | 96 | 30682889 | 1582885495 | 5111 | 17882 | 9-Spring ... | -37.8115 | 144.973 | 2020-02-28 |
| 0-2384... | SCANON | 96 | 20543724 | 1583313645 | 6065 | 18196 | 124-Batma... | -37.822 | 144.956 | 2020-03-04 |
| 0-2633... | SCANOFF | 96 | 34178427 | 1582876819 | 6065 | 6147 | 23-Blyth ... | -37.7675 | 144.98 | 2020-02-28 |
| 0-2498... | SCANON | 96 | 34178427 | 1582876937 | 6065 | 6147 | 23-Blyth ... | -37.7675 | 144.98 | 2020-02-28 |
| 0-2498... | IMPUTED SCANOFF | 96 | 34178427 | 1582876938 | 6065 | 6147 | 23-Blyth ..- | -37.7675 | 144,98 | 2020-02-28 |
| 0-2372... | SCANON | 96 | 11845125 | 1583331334 | 6063 | 6064 | 16-Freema... | -37.7897 | 144.976 | 2020-03-04 |
| 0-2602... | SCANOFF | 96 | 11845125 | 1583337448 | 6063 | 17894 | 11-Melbou... | -37.8057 | 144.974 | 2020-03-04 |
| 0-2384... | IMPUTED SCANOFF | 96 | 20543724 | 1583313646 | 6065 | 20496 | 127-South... | -37.8331 | 144.955 | 2020-03-04 |
| 0-2384... | SCANON | 96 | 20543724 | 1583334931 | 6085 | 20497 | 127-South... | -37.8332 | 144.955 | 2020-03-04 |

### 3.2.1. Stop-level demand computation

The algorithm is designed to pre-process and convert each relevant attribute. It then identifies and maps each transaction to a specific date and stop. All corresponding transactions are
aggregated into half-hour intervals and map each transaction type (tap-on or tap-off) for that particular interval to each stop every day during the data collection period. The total number of transactions for each unique combination is counted and stored in a new column. It is important to note that stop-level and trip-level demand are estimated from both APC and AFC datasets. However, since the APC devices were installed on a limited number of vehicles and were only in use during trial periods in early 2020, they underestimate the demand compared to AFC. Therefore, AFC is the core dataset used to estimate the number of passengers arriving at each stop. APC datasets are only used as supplementary sources to provide additional information related to trips and stops, which is then mapped to the output from AFC data for model improvement.

### 3.2.2. Stop coordinates integration

The stop coordinates (latitude and longitude) are extracted from the AFC data and added to the stop-level demand dataset using a left join method. This method combines two dataframes (2D matrices in Pandas) based on a common key, in this case, column(s) while preserving the order and rows of the left dataframe. When performing a left join, the resulting dataframe will include all the rows from the left dataframe (stop-level demand) and matching rows from the right dataframe (stop information), based on the specified key column(s). If there are multiple matching rows in the right dataframe, all of them will be included in the result. For rows from the left dataframe that do not have a matching row in the right dataframe, the columns from the right one will be filled with NaN (Not a Number) or missing values in the resulting output.

### 3.2.3. AFC and APC data fusion

To obtain the stop order information, which is not available in the AFC dataset, a simple algorithm is written to pre-process, extract necessary information from APC datasets, and integrate it with datasets of estimated demand at the stop level using AFC data. Detecting stop orders facilitates the estimation of onboard loads and the reconstruction of sequential load profiles for each successive trip commencing from the first stop (depot). This procedure begins with the computation of the aggregate count of passengers boarding and alighting at every stop along the route, utilizing transaction data from individual users. Subsequently, this information with the extracted stop order is employed to deduce onboard loads for consecutive trips. Furthermore, employing a stop order provides an effective method to assign numerical codes to stop locations. This enables the model to comprehend and capture both the sequence and position of each stop accurately.

### 3.2.4. Data pre-processing for machine learning

A series of data pre-processing tasks is performed to prepare the dataset for different machine and deep learning models, such as Gradient Boosting Regression, Multi-Layer Perceptron, and Long Short-Term Memory that predict passengers' arrival rates at tram stops. We extract weekday and weekend information and convert it to categorical variables using one-hot encoded dummy variables $(0,1)$ to capture arrival behaviour during weekdays and weekends. To prevent models from being misled by non-homogeneous values from each explanatory variable, a MinMax Normalisation from the scikit-learn library is used to scale the values to a range between 0 and 1, which can improve the performance and convergence of the model. The dataset is randomly split into a training set ( $80 \%$ of the data) and a testing set $(20 \%$ of the data). The proposed LSTM model requires an additional data preparation step, which necessitates to convert two dimensional datasets into three dimensional tensors (i.e., samples, timestamps, features) to match the expected input shape for the LSTM model.

## 4. Results and discussion

### 4.1. Model training and feature selection

This section presents the results of the proposed model compared to benchmarks trained on the same datasets. To ensure fairness in the model performance evaluation, the same input parameters and normalization are applied across all models, i.e., MinMax Scaler and an 80$20 \%$ random training and test split using Sklearn train_test module, respectively. The random process during split serves to mitigate potential biases that may arise from the original order of the dataset, while also ensuring an equal chance for every data point to be allocated to either the training or testing subsets. Four categories of input variables are fed into the models:

- Variables that interact both temporally and spatially, preserving sequences such as stop location information. This indicates that these variables exhibit variation in both temporal and spatial dimensions.
- Time series variables that preserve sequences but only change temporally, like hours and minutes of the day
- Variables that partially correlate in space and fully correlate in time, including weatherrelated attributes. To preserve the partial spatial correlation of weather-related attributes, we have collected city-wide weather information from several available weather forecasting stations and taken the average across all stations to capture locality in space. In other words, temperature and other weather attributes vary only temporally and are assumed fixed across space (i.e., along different stops)
- Variables that are categorically fed into the model. These variables are transformed into dummy variables of 0 and 1 , such as weekends and weekdays

A feature selection using relative importance, proposed by (Friedman 2001), is used to calculate the relative contribution of each explanatory variable from each categories in predicting the target variable, will be explained in details in the coming sections. The proposed model and benchmarks are trained on the training datasets ( $80 \%$ ) and validated on the test set ( $20 \%$ ), respectively. The structure of Gradient Boosting Machine Regression (GBMR) is composed of 200 weak learners (decision trees) used in the ensemble to work sequentially, combining these decision trees to create a strong predictive model. Various depths are tested, with 3 found to be optimal for balancing the model from under and overfitting while reaching convergence. Using PDPs, as explained in the previous section, a non-linear interaction between temperature and stop location is identified. Therefore, we propose a Sequential LSTM model along with a state-of-the-art MLP Regression to capture the complex non-linear interaction. The neural network that the MLP Regressor is built on comprises two hidden layers of 100 and 50 neurons. A ReLU activation function is used to introduce non-linearity in the model, and the maximum iterations are set to 500 , meaning the model will go through the training datasets 500 times during training.
A Sequential LSTM model is comprised of two LSTM layers with 100 and 50 units, and a dense layer with an exponential activation function. The model is compiled using Mean Squared Logarithmic Error (MSLE), which, in parallel with the exponential activation function, is found to be a suitable choice for regression problems. This combination strictly penalizes non-negative outputs and ensures that the outputs are always positive. Since the MSLE loss function calculates the squared logarithmic difference between ground truth and model prediction, it puts higher emphasis on relative error than absolute error. In other words, MSLE is more sensitive to underestimation than overestimation, as errors in the lower range have a larger impact on the loss value compared to errors in the higher range. Additionally, the output from the exponential activation function ranges from $(0, \infty)$; the combination of MSLE and the exponential function can lead to underestimation of the output. Because MSLE is
sensitive to underestimation and a small alteration can lead to rapid growth in the exponential non-linear function, which can significantly impact the output. To diagnose this problem, we propose a weight matrix and use its mean to rescale the prediction from the model to the ground truth to optimize a weighting parameter that can be used to tune the model output (Equation 8). We have tested different weights using the minimum, mean, and maximum between predicted and ground truth values. It has been found that the mean weight produces the best and closest prediction.
$Y_{i}=Y_{1}, \ldots Y_{N}$ and $\widehat{Y}_{l}=\widehat{Y_{1}}, \ldots, \widehat{Y_{N}}$
Where, $Y_{i}$ : is set of ground truth values and $\widehat{Y}_{l}$ : is set of predicted values from model.
We computed the mean of all the samples and utilized the weighting parameter obtained through Equation 9 and 10 to adjust the weight of output. $\partial$ is the weight parameter, and $\bar{Y}$ the weighted output of the model.
$\partial=\frac{\sum_{i=1}^{N} Y_{i}}{N} / \frac{\sum_{t=1}^{N} \hat{Y}_{i}}{N}$
$\bar{Y}=\partial * \widehat{Y}_{l}$

### 4.2. Model performance comparison and findings

To conduct a fair comparison between the proposed methodology and state-of-the-art models, we assume identical input parameters and standardization methods across all models. We used Python 3.10.9, scikit-learn 1.2.1, and Keras 2.10 .0 on a base model MacBook Pro 14 with an Apple M1 chip. The evaluation employed four key performance metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R2 (Equations 11-14). Where, $Y_{i}, \widehat{Y}_{l}$, and $\bar{Y}_{i}$ are ground truth, model estimated output, and mean value across all $Y_{i}$, respectively.
$M S E=\frac{1}{N} \sum_{i=1}^{N}\left(Y_{i}-\widehat{Y}_{l}\right)^{2}$
RMSE $=\sqrt{\frac{1}{N} \sum_{i=1}^{N}\left(Y_{i}-\widehat{Y}_{l}\right)^{2}}$
$M A E=\frac{1}{N} \sum_{i=1}^{N}\left|Y_{i}-\widehat{Y}_{\imath}\right|$
$R=1-\frac{\sum_{i=1}^{N}\left(Y_{i}-\widehat{Y}_{L}\right)^{2}}{\sum_{i=1}^{N}\left(Y_{i}-\bar{Y}_{i}\right)^{2}}$
The features importance ranking in Figure 4 reveals that spatial features, such as the stop location, as the most important determinants of the target value. This parameter exhibits variations across both spatial and temporal dimensions, leading to nonlinearity and bidirectional correlations within the temporal and spatial dimensions. The incorporation of spatio-temporally dependent variables, such as stop locations and their sequential arrangements, is achieved through numerical encoding within the model. This encoding process enables the model to comprehend the spatial positioning of each stop based on the GTFS Stop Sequence, leveraged from historical information during training phase. The model's
architecture accommodates the indirect capture of stop characteristics and latent attributes through intricate configurations of hidden layers within deep neural network. These layers process the input of number-coded stop sequences, derived from historical demand information at each stop. Within this framework, even the introduction of a new, previously unobserved stop between two established stops can be accommodated. In such instances, the model extrapolates the characteristics of the new stop by considering the sequential relationships with preceding and succeeding stops, in conjunction with their spatio-temporal correlations with other attributes, such as temperature. Finally, the model adopts an approach where the stop sequence is encoded as an unordered numerical variable rather than an ordered continuous variable. This characteristic enables the model's versatility, making it adaptable to diverse scenarios encompassing different routes, directions, and properties.
The deep learning framework is capable to capture non-linearities and spatio-temporal correlations. For example, we can clearly see that there is a strong interaction between temperature and stop location, as shown in Figure 5. With upstream stops, the number of passenger arrivals is primarily dependent on temperature, while at middle stops (CBD neighborhoods), it relies on both stop location and temperature (Figure 5). This can be expected, as passengers upstream have the option to travel when the weather is favorable to avoid trips during extreme rain, heat, and cold. In the CBD, considering Melbourne's rapidly changing weather, users may prefer to wait for a short period instead of initiating their trips immediately; therefore, both the location of the stop and temperature jointly affect their decision. In contrast, at downstream stops (excluding the last few stops), the location of the stop has diminishing influence, ultimately having no impact at lower temperatures. This is reasonable because users who transfer from another service to travel a few stops on this route are likely to continue their trips. Due to the proximity of various attractions downstream of Route 96 , some users may have less flexibility in when and where to arrive, taking into account the event's timetable, within these locations. These findings confirm that the strong non-linear interactions between temperature and stop location greatly influence the passenger arrival rate at tram stops.
The other important features after stop location information are the time series attribute, hours of the day, and weather parameters. The type of day, wind speed, and other factors contribute less than $5 \%$ to the relative importance. It is safe to conclude that stop location, time of the day, temperature, and humidity are the key influencers of passenger arrival behavior in Melbourne, Australia. The results in Table 3 and Figures 6-8 show that the $\partial$ : Mean weight parameter produces the best predictions, outperforming all metrics, including the difference between the mean values of test data $\left(\mu: Y_{i}\right)$ and predicted data ( $\mu: \widehat{Y}_{l}$ ), MSE, MAE, and $\mathrm{R}^{2}$, respectively. When moving from the minimum to mean weight parameter, the model slightly improves, while in contrast, with the maximum weight parameter, it significantly improves, showing a $32 \%$ enhancement in $\mathrm{R}^{2}$ value.
Last but not least, we compared the performance of the proposed model with the GBMR and MLP Regression using several indicators, as mentioned above. It can be clearly seen that the proposed method surpasses the benchmarks in all the given performance measures (Table 4). For example, Wasea-Lstm achieved a $6 \%$ and $15 \%$ improvement in R2 and MSE compared to GBMR and MLP, respectively. Furthermore, the prediction power of each model is compared with the ground truth, which helps visualize the distribution of the predictions against the tested dataset (Figure 9 and 10). In all scenarios, the Wasea-Lstm model demonstrates superiority over the benchmark models. For example, Wasea-Lstm model exhibits a dense concentration at the bottom, compared to the sparse concentration of GBMR and MLP in the absolute difference plot in Figure 9 (left-hand), and the tallest peak in the residual distribution curve in Figure 9 (right-hand), indicating that Wasea-Lstm produces lower errors. Ultimately, WaseaLstm predictions fit the ground truth the best, as shown in Figure 9C. This superiority can be
attributed to the innovative framework introduced, which combines the strengths of a deep neural network architecture and the adept capturing of various types of dependencies. These dependencies include the spatio-temporal interaction rooted in stop location characteristics, the partially spatial and fully temporal attributes tied to weather features, as well as the temporal relationships inherent in time and day attributes. To summarize, the proposed architecture within the realm of deep learning not only outperforms the benchmark models across all evaluation metrics but also generates precise predictions characterized by the minimal residuals. Furthermore, its applicability extends to a wide array of real-world scenarios and case study applications.


Figure 4: Feature importance ranking by the Mean Decrease Impurity (MDI)
Table 3: Wasea-Lstm model comparison with different weight parameters, including Min, Mean, and Max

| Weight par-LSTM | $\mu: Y_{i}$ | $\mu: \widehat{Y}_{l}$ | MSE: $\widehat{Y}_{l}$ | MAE: $\widehat{Y}_{l}$ | $\mathrm{R}^{2}: \widehat{Y}_{l}$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $\partial:$ Min | 5.543 | 4.338 | 55.915 | 3.455 | 0.468 |
| $\partial:$ Mean | 5.543 | 5.543 | 54.854 | 3.644 | 0.478 |
| $\partial:$ Max | 5.543 | 9.697 | 121.592 | 5.859 | 0.157 |

Table 4: Performance comparison

| Model | MSE | RMSE | MAE | $\mathrm{R}^{2}$ |
| :--- | :--- | :--- | :--- | :--- |
| GBMR | 69.6 | 8.3 | 4.6 | 0.34 |
| MLP | 60.7 | 8.3 | 4.2 | 0.42 |
| Wasea-Lstm | 54.8 | 7.4 | 3.6 | 0.48 |

```
PDP interact for "gtfs_stop_sequence" and "temp"
Number of unique grid points: (gtfs_stop_sequence: 10 , temp: 10)
```



Figure 5: Partial Dependence Plot, shows the dependence of the number of passenger arrival on joint values of temperature and stop location


Figure 6: Predicted (orange) values by Wasea-Lstm versus ground truth (blue), using minimum weight parameter.


Figure 7: Predicted (orange) values by Wasea-Lstm versus ground truth (blue), using mean weight parameter.


Figure 8: Predicted (orange) values by Wasea-Lstm versus ground truth (blue), using Maximum weight parameter.


Figure 9 (left-hand): Absolute difference between predicted and ground truth using GBM, MLP, and Wasea-Lstm, (right-hand): Comparison of the distribution of the absolute residuals


Figure 9C: The correlation of model prediction versus ground truth

## 5. Conclusion

In this study, we introduce a novel deep learning model, the Wasea-Lstm (Weather-Aware Smart Exponential Activation LSTM Estimator), designed for predicting passenger arrival rates at tram stops in Melbourne, Australia. This model utilizes multi-dimensional highresolution datasets obtained from automatic fare collection (AFC), automatic passenger count (APC), and weather data sources for one of the largest tram routes in Melbourne, Australia. The research proposes a deep neural network architecture capable of capturing spatial, temporal, and weather correlations at finer spatio-temporal resolutions. The developed model successfully captures demand non-linearity and variability along the route through the integration of multi-source data.
Key factors influencing passenger arrival behaviour in Melbourne, Australia include stop location, time of day, temperature, and humidity. The study uncovers a strong interaction between temperature and stop location, with upstream stops primarily influenced by temperature, while middle stops (CBD neighbourhoods) are influenced by both stop location and temperature (Figure 5). This is expected, as passengers upstream have the option to travel when weather conditions are favourable, avoiding extreme heat or cold. In contrast, CBD passengers may choose to wait briefly rather than begin their trips immediately, considering Melbourne's rapid weather changes, thus both location and temperature jointly affect their decisions.
Although the proposed Wasea-Lstm model surpasses benchmark models like Gradient Boosting Machine Regression (GBMR) and Multi-Layer Perceptron Regression (MLP) across all performance metrics, including R2 value, Mean Squared Error (MSE), and Mean Absolute Error, yet it has some limitations that can be explored in the future. For instance, this study does not account for service frequency's impact on passenger arrival, which could be extended to different services with varying or unique headways to determine frequency's effect on demand estimation and prediction. Moreover, weather data is collected at the city level for Melbourne, not at the stop level, due to the limited number of weathers forecast stations and the impracticality of obtaining local weather data for each route or stop. However, it is reasonable to assume that weather fluctuations within the same city have a negligible effect on behaviour at each stop, and results from multiple stations are averaged to capture this locality. As local weather forecasts become available in the future, exploring the spatial correlation of weather attributes would be interesting. Furthermore, the historical demand information comprises datasets gathered during the Covid-19 lockdown, exhibiting a noticeable decrease in user numbers. Utilizing uninterrupted data spanning the entire year will hold significant importance. Lastly, the look-back time window in the Wasea-Lstm architecture is set to 30 minutes, but investigating the impact of alternative look-back intervals would be a valuable research topic.

## References

Ahn, S, Kim, J, Bekti, A, Cheng, LC, Clark, E, Robertson, M \& Salita, R 'Real-time information system for spreading rail passengers across train carriages: Agent-based simulation study',
API, VCW 2020, Hourly Weather information for Melbourne, Australia, 2020 edn, Visual Crossing Weather, https://www.visualcrossing.com/weather/weather-data-services/Melbourne?v=api, 2023.
Arshad, M \& Ahmed, M 2021, 'Train delay estimation in Indian railways by including weather factors through machine learning techniques', Recent Advances in Computer Science and Communications, vol. 14, no. 4, pp. 1300-1307.
Bao, Y, Zhang, C \& Shi, Q 'Mining and Analysis Based on Big Data in Public Transportation', pp. 681-688.
Breiman, L 1984, Classification and Regression Trees, Routledge.
Ceder, A 2007, Public Transit Planning and Operation: Theory, modelling and practice, Butterworth-Heinemann, Elsevier, UK.

Currie, G, Goh, KC \& Sarvi, M 2012, 'Exploring the impacts of transit priority measures using automatic vehicle monitoring (AVM) data', in Australasian Transport Research Forum, Perth.
Cybenko, G 1989, 'Approximation by superpositions of a sigmoidal function', Mathematics of Control, Signals and Systems, vol. 2, no. 4, pp. 303-314.
Downey, L, Fonzone, A, Fountas, G \& Semple, T 2022, 'The impact of COVID-19 on future public transport use in Scotland', Transportation Research Part A: Policy and Practice, vol. 163, pp. 338-352.
Friedman, JH 2001, 'Greedy function approximation: A gradient boosting machine', The Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 1144.
Frumin, M \& Zhao, J 2012, 'Analyzing Passenger Incidence Behavior in Heterogeneous Transit Services Using Smartcard Data and Schedule-Based Assignment', Transportation Research Record, vol. 2274, no. 1, pp. 52-60.
Hastie, T, Tibshirani, R \& Friedman, JH 2017, The Elements of Statistical Learning Data Mining, Inference, and Prediction, Second Edition edn, Springer.
Hensher, DA 2020, 'Chapter 24 - The effects of passenger crowding on public transport demand and supply', in DA Hensher (ed.), Bus Elsevier, [https://www.sciencedirect.com/science/article/pii/B9780128201329000248](https://www.sciencedirect.com/science/article/pii/B9780128201329000248), pp. 295-308.
Hörcher, D \& Graham, DJ 2018, 'Demand imbalances and multi-period public transport supply', Transportation Research Part B: Methodological, vol. 108, pp. 106-126.
Ingvardson, JB, Nielsen, OA, Raveau, S \& Nielsen, BF 2018, 'Passenger arrival and waiting time distributions dependent on train service frequency and station characteristics: A smart card data analysis', Transportation Research Part C: Emerging Technologies, vol. 90, pp. 292-306.
Ke, J, Zheng, H, Yang, H \& Chen, X 2017, 'Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach', Transportation Research Part C: Emerging Technologies, vol. 85, pp. 591-608.
Kholodov, Y, Jenelius, E, Cats, O, van Oort, N, Mouter, N, Cebecauer, M \& Vermeulen, A 2021, 'Public transport fare elasticities from smartcard data: Evidence from a natural experiment', Transport Policy, vol. 105, pp. 35-43. Liu, Y, Xu, H, Yu, X \& Zhou, J 2022, 'Heuristic feeder-bus operation strategy considering weather information: a chance-constrained model', International Transactions in Operational Research, vol.
Mai, E, List, G \& Hranac, R 2012, 'Simulating the travel time impact of missed transit connections', Transportation Research Record, vol. no. 2274, pp. 69-76.
Rezazada, M, Nassir, N \& Egemen, T 'Public transport bunching: A critical review with focus on methods and findings for implications for policy and future research', Adelaide, Australia, ATRF,
scikit-learn, 4.1. Partial Dependence and Individual Conditional Expectation plots, scikit-learn, scikit-learn.org, Smith, DA \& Sherry, L 'Decision support tool for predicting aircraft arrival rates from weather forecasts',
Soza-Parra, J, Raveau, S \& Muñoz, JC 2021, 'Travel preferences of public transport users under uneven headways', Transportation Research Part A: Policy and Practice, vol. 147, pp. 61-75.
Toro-González, D, Cantillo, V \& Cantillo-García, V 2020, 'Factors influencing demand for public transport in Colombia', Research in Transportation Business \& Management, vol. 36, p. 100514.
Venkatesh, V, Arya, A, Agarwal, P, Lakshmi, S \& Balana, S 'Iterative machine and deep learning approach for aviation delay prediction', pp. 562-567.
Victoria, PT 2018-19, Public Transport Victoria, Annual Report, Public Transport Development Authority operating as Public Transport Victoria.
Yang, M, Chen, C, Wang, L, Yan, X \& Zhou, L 2016, 'Bus arrival time prediction using support vector machine with genetic algorithm', Neural Network World, vol. 26, no. 3, pp. 205-217.
Yao, H, Wu, F, Ke, J, Tang, X, Jia, Y, Lu, S, Gong, P, Li, Z, Ye, J \& Chuxing, D 'Deep multi-view spatialtemporal network for taxi demand prediction', pp. 2588-2595.
YarraTrams 2023a, Facts \& figures https://yarratrams.com.au/facts-figures, viewed 11/04/2023,
YarraTrams 2023b, Timetables \& routes, https://yarratrams.com.au/route-guides/route-96, viewed 11/04/2023,
Zhang, MJ, Chen, C \& Han, MX 'Passenger waiting time and behavioral adaption to suburban bus timetable', pp. 1199-1203.

