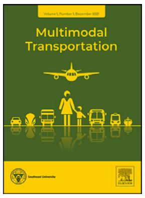
Multimodal Transportation 3 (2024) 100120

Contents lists available at ScienceDirect

Multimodal Transportation

journal homepage: www.elsevier.com/locate/multra



Full Length Article

Prediction of rail transit delays with machine learning: How to exploit open data sources

Malek Sarhani [a,b], Stefan Voß [a,*]

[a] Institute of Information Systems (IWI), University of Hamburg, Von-Melle-Park 5, Hamburg 20146, Germany
[b] School of Business Administration, Al Akhawayn University in Ifrane, Avenue Hassan II, P.O. Box 104, Ifrane 53000, Morocco

ARTICLE INFO

Keywords:
Public transport
Machine learning
Arrival time prediction
Delay prediction
Open data
Rail transit

ABSTRACT

The use of public transport data has evolved rapidly over the past decades. Indeed, the availability of diverse data sources and advances in analytics have led to a greater emphasis on utilizing data to enhance public transport services. Rail transit systems have increasingly become the preferred mode of travel due to their comfort, speed, and (mostly) emission-free nature. However, persistent delays continue to be a concern. Machine learning-based prediction of transit delays is an emerging field gaining recognition. The first contribution of this paper is to illustrate how to exploit available open data to improve the prediction of rail transit delays using machine learning. Moreover, through a comparison of various well-known machine learning approaches, we show that they can yield significantly different results. Notably, the improved support vector machine method presented in this study exhibits exceptional performance and is well-suited for long-term predictions. Furthermore, we have incorporated explainable artificial intelligence techniques to identify and assess the most significant factors influencing delays. To perform experiments with the method and draw robust conclusions, three case studies featuring different rail services in major cities are provided.

1. Introduction

Public transport provides an important service whose necessity is becoming more widely recognized. In fact, it has emerged as one of the most important modes of transportation, particularly for reducing motorized individual transportation and achieving sustainability while lowering emissions. However, managing the public daily commute network is a demanding challenge, especially in today's world of fast urbanization and associated population growth. As a result, public transportation systems must implement proper tools and make use of accessible data to address these challenges. Indeed, it is illustrated in several studies (e.g. Schneidereit et al. (1998), Jevinger and Persson (2019) and Berggren et al. (2021)) that the gathering and use of trustworthy information and data is critical for the successful operation of public transport systems (for system operators as well as users, alike).

In the ideal case, when focusing on scheduled services, intended arrival and departure times provided by agencies should match the real ones. Nevertheless, delays often occur as they are influenced by several (outside and unforeseen) factors and then have to be managed by the agencies. Therefore, predicting transit travel time and delays at each stop is a rising topic that attracts agencies around the globe. In fact, accurate prediction of transit delays can improve transit service and increase passenger use and satisfaction.

In the past, predicting and analyzing public transit delays has been difficult due to a lack of availability of real-time data on public transit schedules and related data sources, in addition to the shortcomings of classical data analysis approaches. However, in recent years we have experienced a huge advance in both data acquisition and processing. On the one hand, transit agencies have adopted

[*] Corresponding author.
E-mail address: stefan.voss@uni-hamburg.de (S. Voß).
URL: https://www.bwl.uni-hamburg.de/en/iwi.html (S. Voß)

https://doi.org/10.1016/j.multra.2024.100120
Received 12 April 2023; Received in revised form 22 October 2023; Accepted 25 October 2023

and deployed vehicle-based global positioning system (GPS)-based tracking systems to provide real-time updates on transit locations and arrival times at stops. On the other hand, data analysis approaches are emerging and evolving quickly these days to allow new perspectives on improving public transportation systems.

As a result, the availability of data, coupled with advanced approaches to data analysis, has paved the way for the use of advanced data analysis approaches in public transport, which aims to leverage the different available data sources. Therefore, the public transport community is increasingly emphasizing the development of commonly collected data sources on public transport as well as powerful analytical tools.

More specifically, many agencies aim to publish openly their updates on transit arrival following the rise of the general transit feed specification (GTFS) standard and are updating their policies for that. In fact, over the past decade, GTFS has emerged as an industry standard for publishing data about transit operations. GTFS feeds are divided into static and real-time transit feeds. GTFS static data (Google (2021)) is mainly about timetables of scheduled trips, while GTFS real-time data (Google (2022)) contains constantly updated trip information, including arrival prediction, as well as live vehicle location reports based on GPS (Wessel et al. (2017)).

One way that agencies are adopting to cope with service delays and disruptions is to issue updates on their transit schedules using the GTFS standard. However, providing accurately predicted information on the arrival time is still an ongoing issue in the literature, and the provided predictions are questioned in several occasions. In particular, machine learning (ML) seems to become the most promising way to tackle such data and provide accurate predictions. ML uses different factors, known as features, as an input to achieve the best possible prediction. ML outperformed conventional methods in several occasions (e.g. Nair et al. (2019) and Tang et al. (2020)). However, the accuracy of ML approaches mainly relies on the acquired data and features.

In particular, to enhance delay prediction, researchers can make use of the wealth of data available to improve prediction accuracy. To achieve this purpose, identifying the features influencing transit delays is a crucial step needed to improve the reliability of transit systems. It is known that one source of data will not be enough to have accurate predictions. Liu and Miller (2020) suggest that real-time data is not sufficient for prediction and recommend to consider other data to enhance it. Therefore, researchers are increasingly considering multiple factors, potentially from different sources, to reach more accurate predictions. An example of work in this regard is Wu et al. (2021).

While GTFS is the typical standard for open data, other open data sources, not integrated so far in GTFS, provide rich sources of information. The main contribution of this paper is to show how open data can be exploited to better predict the transit arrival times. Open data is a hot concept these days that needs further consideration by the agencies. To our knowledge, this paper is among the first to adopt open data to provide accurate predictions of delays.[1] In addition to GTFS data information, which are increasingly shared by agencies with the public via websites using data standards, we use in this paper weather data. Weather information can be collected via Application Programming Interfaces (APIs). Furthermore, we exploit the statistics about passengers which are provided by agencies. To investigate the use of ML to predict transit delays with these data, we compare the performance of the most adopted ML models for predicting transit delays. The aim is to systematically assess the predictive performance of main ML models that take different available open data.

The experiments in this paper focus on rail transit systems, which have gradually become the preferred mode of travel due to their comfort, speed, and emission-free features. However, the persistent occurrence of delays, especially during peak times, remains a significant issue. In the absence of a reliable estimate of delays, some passengers may choose private cars over rail transport. It is worth noting that research on rail transit is relatively limited compared to buses; see, e.g., Kuo et al. (2023). Our objective in this paper is to provide a more accurate prediction for rail transit services than what is currently proposed.

In this paper, we show that it is possible in the public transport domain to integrate openly available data sources to enable the extended use of ML in transport policy to improve prediction accuracy, finally allowing a better linkage between transport service providers and their customers. We adopt proper feature engineering techniques in addition to recent trends in explainable and interpretable ML. This helps to have the most accurate prediction of arrival time and to understand the factors that affect it.

In summary, this paper presents three significant contributions that, to the best of our knowledge, have not been previously applied in the context of rail transit:

- Our proposal introduces a unified approach that combines internal and external data sources to enhance rail transit prediction.
- We apply the best practices feature engineering approach in conjunction with a well-used ML approach for improved results.
- To gain insights into the crucial features, we employ state-of-the-art explainable artificial intelligence techniques and conduct three distinct case studies focusing on rail transit.

The remainder of the paper is organized as follows: In the next section, we present a literature review. In section 3, we expose the included ML approaches. Section 4 is dedicated to the data acquisition and construction process. Section 5 presents the experiments conducted, starting with the main case study of Canberra, Australia, and followed by two other major cities. Finally, a conclusion is presented in Section 6.

[1] We should point the reader to the very recent developments in MobilityDB as a free and open source novel moving object database, developed as a PostgreSQL and PostGIS extension, that adds spatial and temporal data types along with a large number of functions, that facilitate the analysis of mobility data actually also incorporating delay determination; see Godfrid et al. (2022). In addition, we should note that data-availability problems seem diminishing as pointed out in the cases below. Another example emphasizing transparency efforts including delay data relates to Zurich (Switzerland) where data availability is shown, e.g., on https://data.stadt-zuerich.ch/dataset/vbz_fahrzeiten_ogd_2019 and https://data.stadt-zuerich.ch/dataset/vbz_fahrzeiten_ogd_2021 (access date Feb 28, 2022).

2. Literature review

In this section, we aim to review the related literature. We start by showing how the aforementioned data sources are exploited to improve transportation in general and transit in particular. Then, we illustrate how ML is exploited in predicting transit delays. Before going into detail, we point to some general exposition regarding mobility data as it can be found, e.g., in Pelekis and Theodoridis (2014); Renso et al. (2013). A recent survey is Ge et al. (2021).

First, public transit data come from a variety of sources. The issue of the exploitation of the different data is extensively discussed and illustrated in Ge et al. (2021). In that paper, the authors summarize and analyze the potentials and challenges of the main data sources. In addition, they emphasize the complementary aspect of these data sources and how to merge them to broaden their contributions and face their challenges.

In particular, the prediction of arrival times using available data sources is attracting researchers these days. Typical data sources are automatic vehicle location (AVL) and automatic passenger counting (APC). For example, Barabino et al. (2017) propose an approach that collects and handles AVL data, computes passenger patterns from passenger arrival data, and integrates AVL data and patterns. However, although powerful in analyzing performance, AVL/APC data is not often made publicly available and lacks a standardized format. Moreover, the information on these data has to be validated as illustrated in Gilmore and Reijsbergen (2015). In that paper, the authors stress the importance of the validation of AVL data and provide an insight for this. In addition to AVL and APC, information on the passengers' preferences can also provide insightful information (e.g. Kumar et al. (2018)); however, they are mostly confidential and can not be openly available.

For rail transit, disturbances in general (see, e.g., Ge et al. (2022b) for a survey) and delays in particular are influenced by many factors. The main information are available in GTFS data. Other openly available information concerns weather and the number of passengers. Below, we briefly review them.

First, regarding GTFS, Wessel et al. (2017) aim to incorporate real-time data into GTFS (GTFS real-time was not frequently used at that time) and to standardize it. Moreover, the paper argues that deviation from the schedule does not appear to be random, which is a main motivation for the prediction of delays. GTFS is used for a related purpose in Park et al. (2020). In that paper, the authors analyze the spatio-temporal patterns of propagating delays using publicly available schedule and real-time location data from the Central Ohio Transit Authority. The authors point out that arrival time updates provided by the transit agencies should be further validated with more precise actual arrival-time observations. At a practical side, a crucial issue within GTFS, especially with GTFS real-time, is how to handle it and extract the needed information. We refer to Lim et al. (2019) as an example of work which can help understand how GTFS data can be extracted, stored and processed. Also, Chondrodima et al. (2022) propose a ML approach for the prediction of delays.

Second, in addition to GTFS, weather is another source of data that, while important, has not received much interest for delay prediction. In fact, it is affirmed in several papers (e.g. Miao et al. (2019)) that weather disturbances have a negative impact on transit service. However, such an impact varies according to the different modes and the initiatives of the agencies towards them. Moreover, it depends on the meteorological nature of the region. It is then impractical to estimate the weather impact manually, especially when combining with other data. Indeed, Wei et al. (2019) show that weather influence on the transit varies across the course of a day.

Although weather is not yet used to predict rail transit delays, it is used for similar prediction problems, as in Schultz et al. (2021). For this purpose, it has to be integrated with other data, as shown in Huang et al. (2020). For rail transit, both papers Wu and Liao (2020) and Zhao et al. (2018) suggest that the subway is less vulnerable to inclement weather and can replace other travel modes in this case. This may lead to an increase of the number of passengers, which is another feature that can be considered.

Third, the number of passengers is an important factor that needs to be considered. Despite that APC data is often not openly available, agencies provide useful open data regarding passengers such as the Hamburger Verkehrsverbund (HVV).[2] Many papers are actually interested in getting more accurate information about the passengers such as Ni et al. (2016). In this paper, we focus on the number of passengers using a transit service, known as patronage or ridership. In fact, some agencies publicly provide data on patronage, which can offer relevant information for analysis (as described in Section 4).

In this paper, we aim to use ML to have more accurate GTFS arrival predictions based on other available data. In the literature, several approaches are used for delay prediction such as state-space models, e.g., Kalman Filter (Kumar et al. (2017)) and statistical approaches, like ARIMA (Alzyout and Alsmirat (2020)). The main particularity of ML compared to these approaches is its ability to incorporate different heterogeneous and massive data sources (Zhou et al. (2017)). This is in line with Ge et al. (2021), who stress that by fusing different data sources, the information on one data source, such as GTFS, can be validated by others and new knowledge can be mutually derived. This is illustrated, for the case of delay prediction, in Wang and Zhang (2019), who adopt mixed datasets of weather, train delay and train schedule. These data sources are recorded, collected and analyzed in order to understand the patterns of train delays and to predict the delay time.

[2] Various transport companies provide an annual customer satisfaction report, such as the city of Hamburg, Germany at https://www.hvv.de/de/ueber-uns/publikationen. Related reports can also be found in other cities. An example for Qingdao, China is given in Qdbus (2014). We should also note the availability of transit data in many countries and regions based on extensive questionnaire and interview works. While these questionnaires are important for transit companies, one might argue whether there should be as many as can be found in academic literature rather than at the companies/associations; see, e.g., the discussion in Voß et al. (2020).

Different ML models can be applied to improve the prediction accuracy. The main ML approaches used for delay prediction are linear regression (LR), neural networks and particularly deep neural networks (DNN), gradient boosting (GR), random forests (RF) and support vector machines (SVM).

LR is a traditional approach for prediction and is used for delay prediction for a long time (e.g. Sun et al. (2003)). But, in the past decade, the focus moved to other advanced approaches, such as SVM, RF, GR and DNN, which are gaining much attention these days. In Shoman et al. (2020), the authors propose a framework which is fueled by large, heterogeneous bus transit data (GTFS) and vehicle probe data. In that paper, the authors utilize entity embeddings to enable the framework to simultaneously fit functions and learn patterns from both categorical and continuous data streams. Moreover, in Nithishwer et al. (2022), the authors propose a DNN variant that outperforms a historical average approach, LR, ANN, long short-term memory (LSTM) and Conventional LSTM methods. Also, in Shi et al. (2021), a GB prediction model is established to capture the relation between the train arrival delays and various railway system characteristics. RF is another approach that is showing promise in prediction. For example, Li et al. (2020) found that the RF model exhibits high prediction accuracy for short-term train delay prediction. The hybridization of ML approaches is also promising. For example, in Nimpanomprasert et al. (2022), a comparison between two hybrid neural network models for bus travel time prediction can be found. Another promising ML approach is SVM. Indeed, it is shown in Yu et al. (2011) that SVM yields better results than other ML approaches for a bus arrival prediction case study. In this paper, we propose an enhancement of SVM as described below.

Understanding the importance of different features is of utmost importance. Explainable artificial intelligence techniques are rapidly improving to provide better-explained models, which are more helpful for decision-making tasks. An example of related work in this regard is Wagner et al. (2022).

We can conclude from the literature that ML is widely adopted these days for delay prediction. Nevertheless, to the best of our knowledge, this is the first paper that leverages open source data to enhance existing rail transit systems and analyze the causes of delays.

To make the review more comprehensive, we like to comment on a few additional issues related to the topic. For instance, He et al. (2019) emphasize that the prediction of travel times may be distinguished according to the passenger's riding time on multiple bus trips. While this does not account for transfer synchronization, it provides some indication of this topic as an optimization problem (see, e.g., Daduna and Voß (1995)), which requires further research. Surveys on trip duration prediction, as well as prediction methods for arrival times, are provided in Al-Naim and Lytkin (2021).

3. Machine learning for predicting transit delay

The aim of this section is to highlight the adopted ML approaches to predict transit delay, which is a continuous variable. In fact, there are a number of ML methods that yield satisfactory results for regression (continuous) problems such as SVM, RF, GB, DNN and LR. In our study, we focus on these approaches as they are widely adopted for regression problems.

In this paper, we begin by presenting SVM and the pre-processing techniques we utilize with it. Subsequently, we provide a brief overview of the other methods mentioned.

3.1. Support vector machines and feature pre-processing

SVM is a ML approach that has been successfully applied to many fields such as pattern recognition and economics. Indeed, it provides an effective tool for both classification and regression problems. In particular, for regression problems such as transit delay prediction, the regression version of SVM is appropriate. SVM has shown effectiveness for accurate arrival delay prediction as shown, e.g., in Marković et al. (2015). Below, we describe the SVM version for regression.

Given a training dataset with input vectors $\mathbf{x}_i \in \mathbb{R}^p$ and corresponding target values $y_i \in \mathbb{R}$, SVM seeks to find a function $f(\mathbf{x})$ that predicts the target value y for a given input vector $\mathbf{x}$. The SVM formulation involves finding the optimal values of the function parameters and the bias term.

$$\min_{\mathbf{w},b,\xi,\xi^*} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{n}(\xi_i + \xi_i^*)\right)$$

subject to:

$$\begin{cases} y_i - \mathbf{w}^T\phi(\mathbf{x}_i) - b & \leq \varepsilon + \xi_i^* \\ \mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i & \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* & \geq 0 \quad \forall i \end{cases}.$$

Where: $\mathbf{w}$ represents the weight vector, b is the bias term, ξ and ξ^* are the slack variables, $\phi(\mathbf{x})$ denotes the feature mapping function, C is the penalty parameter that controls the trade-off between the margin violations and the training error, ε is the width of the epsilon-insensitive tube that defines the tolerance around the regression line. The objective of the optimization problem is to minimize the regularization term $\frac{1}{2}\|\mathbf{w}\|^2$ while maintaining the errors within the tolerance ε and penalizing the margin violations through the slack variables ξ and ξ^*.

Once the optimization problem is solved, the predicted value for a new input vector $\mathbf{x}$ can be calculated as $f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$. The value of b is determined during the training process of SVM. (For more details on SVM for regression, the interested reader is referred to Smola and Schölkopf (2004)).

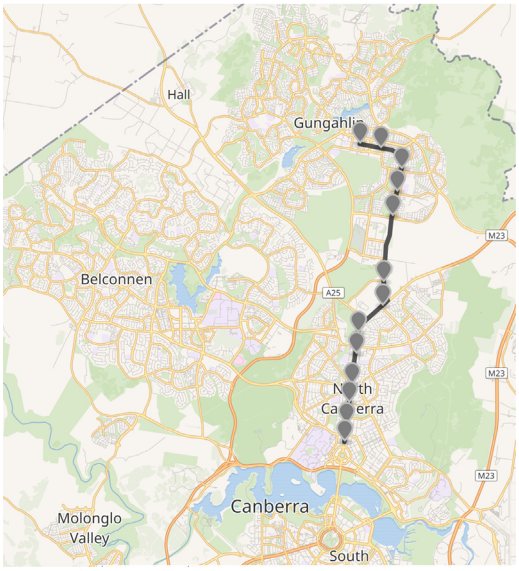


Fig. 1. Map of the Canberra metro system.

In this paper, we particularly emphasize our aim to study the impact of feature selection (FS) on SVM (e.g., Sarhani and Voß (2021)). The goal of FS is to improve the generalization capacity of learning algorithms by removing redundant, irrelevant, and noisy features (Guyon and Elisseeff (2003)). Previous literature has shown the effectiveness of FS in enhancing SVM prediction (Sarhani and Voß (2021)), which is why we are investigating it in the experiment described below. Another essential pre-processing phase is parameter tuning. Indeed, finding the best parameters for SVM is of utmost importance (Sánchez A (2003)) and can have a crucial impact on its performance, as indicated in previous work (e.g., Sarhani and Afia (2016)). In this paper, we follow best practices for both FS and parameter tuning to achieve the best results.

3.2. Other machine learning techniques

In this part, we briefly present the ML techniques compared to SVM. First, RF is a ML approach that uses ensemble learning methods for regression (Breiman (2001)). An ensemble learning method is a technique which combines predictions from various ML algorithms.

Second, GB is a sort of boosting used in ML. It is based on the assumption that when the next best potential model is coupled with earlier models, it minimizes the overall prediction error (Mason et al. (1999)).

Third, regarding DNN, the concept of deep learning originated from the study on artificial neural networks (ANNs). ANNs have become an active area of research over the past decades. In particular, DNN has recently made massive strides with successful applications. More details on DNN and their usage can be found in Zhang et al. (2019).

Fourth, LR is one of the most well known and simplistic ML algorithms. LR is based on a linear model that assumes a linear connection between the input variables and the output variable (Olive (2017)). The details of the adopted parameters of the different methods are presented in Section 5.1.

4. Data acquisition and construction

The aim of this part is to describe the different data used to predict transit delay for the Canberra case study. The data acquisition for the two other case studies is similar and the differences are highlighted below. We start by describing the case study. Then, we show how we collected the data and we describe the different features.

4.1. Case study

In this paper, we take the Canberra Light Rail System also known as Canberra Metro as a case study. The Canberra Metro is a light rail system serving the city of Canberra, Australia. The original 12-kilometer line connects the northern town center of Gungahlin to the city center and has 13 stops. Fig. 1 shows a graphical overview of the line transit.[3]

In this study, different data sources are adopted in order to study the possible delay patterns of the transit and to extract the different factors affecting its delay. More specifically, we adopt the ACT Government Open Data Portal (the data is available at

[3] More information on the network is available at https://cmet.com.au/frequency-guide/.

https://www.data.act.gov.au/). The portal contains different open data sources available in separate files and we aim to merge them in the appropriate way. The data sources used relate to (static) stop times, trip updates and vehicle positions, which are provided under the GTFS specification. The first data source is part of GTFS static while the second and third belong to GTFS real-time. The other data used are the weather and the patronage of the transit. Below, we describe the different data.

4.2. GTFS static

The static data sources included in this study concern the arrival times at each stop as well as the corresponding information (e.g. location) of the different stops. The time information are available in the “Stop times” file, which aims to provide the arrival and departure times at a specific stop for a specific trip on a route. To further enrich the information, we have added the information from location on each stop, which is available on the “Stop” file. The static GTFS data are extracted from the open data portal available at https://www.transport.act.gov.au/contact-us/information-for-developers. (Accessed on January 9, 2023.)

4.3. GTFS real-time

GTFS real-time information is mainly divided in trip update and vehicle positions, and is often displayed in two separate files.

Trip update

In general, in the ideal case, the expected arrival times mentioned above should correspond to the actual and published times of the transit. However, delays are inevitable. To be updated about them, the necessary information is provided in the trip update data source. In other words, its aim is to display the actual transit situation and to represent fluctuations in the timetable.

Typically, as proposed in the GTFS specification, these updates give a predicted arrival or departure time for stops along the route. But, for some data, agencies provide historical information on arrival and prediction times. This is the case with our studied data in which the information on arrivals and departures is provided after their occurrences.

This fact allows us to compare the actual values with those predicted. In fact, the trip update file includes the predicted output, which is the arrival delay. We also include other variables such as the arrivals’ incertitude.

We note that in this study, we are only interested in arrival delays, and we do not consider any other strongly correlated information (mainly the actual time of arrival and the times and delays of departure). Therefore, the dwell time is not considered in this study.

The trip update data is extracted from the following URL: https://www.data.act.gov.au/Transport/Canberra-Metro-Light-Rail-Transit-Feed-Trip-Update/jxpp-4iiz. (Accessed on January 9, 2023.)

Vehicle positions

Vehicle positions are often used to provide automatically generated information about the location of a vehicle, e.g., from an on-board GPS device. A single vehicle position should be provided for each vehicle capable of providing it. Our goal in introducing this data source is to study the impact of vehicle information. Indeed, it is well known that the location of vehicles plays an important role in keeping them on schedule and can cause significant delays. Such information is not available for predicted instances, but the other information may also impact ML prediction as is investigated in this paper. The provided information in this paper includes the vehicle ID and label, current status, and whether it is running smoothly. The data is available at the following URL: https://www.data.act.gov.au/Transport/Canberra-Metro-Light-Rail-Transit-Feed-Vehicle-Upd/92fy-xvmy. (Accessed on October 9, 2021.)

4.4. Weather

In addition to GTFS real-time, weather is another real-time information which is valuable in many fields. Thereby, many APIs have been proposed. An API is a set of instructions that allows software programs to interact with each other and they are used in this context to extract meteorological information in real time. Here, we adopt an existing API for this purpose.[4] As indicated in the literature, the weather influence that can affect the transit concerns the temperature, precipitation, and winds. Indeed, such information can affect the number of passengers and the delays. These weather information can be extracted from the following URL: https://www.wunderground.com/. (Accessed on January 31, 2022.)

4.5. Patronage

Patronage (or ridership) is another important data source, which attracted researchers for a while (e.g. FitzRoy and Smith (1998)). In recent years, agencies are increasingly publishing their data online.[5] In particular, for Canberra, the agency provides data that contains the light rail patronage for every 15 minutes, starting from the interval (00:00 to 00:14) to the interval (23:45 to 23:59). The data is available on the portal at the following URL: https://www.data.act.gov.au/Transport/Light-Rail-Patronage-15-min-interval/xvid-q4du. (Accessed on January 9, 2023.)

[4] More information on the data collection process can be found at the following URL: https://medium.com/@dd93/collecting-weather-data-to-boost-data-science-models-with-selenium-390d9db88210. (Accessed on January 31, 2022.)

[5] In addition to our case study, an example of an agency providing this data is Transport for New South Wales https://opendata.transport.nsw.gov.au/nsw-public-transport-patronage-data-on-open-data-hub.

Table 1
Description of the features used in the study.

	Feature	Description
GTFS static	Trip ID	A specific trip identification
	Stop ID	A stop identification
	Stop sequence	Order of a stop for a particular trip
	Stop headsign	Text that appears on signage identifying the trip's destination
	Pick-up type	Pick-up method
	Drop-off type	Drop-off method indication
	Arrival time	Planned arrival time at a specific stop for a specific trip
	Stop latitude	Latitude of the stop location
	Stop Longitude	Longitude of the stop location
	Stop Name	Name of the location of the stop
	Date	Date of the trip
GTFS real-time	Arrival uncertainty	Uncertainty provided by the Canberra transport agency
	Vehicle ID	Internal identification for the vehicle
	Vehicle label	A visible label by the user
	Latitude	Latitude for the vehicle position
	Longitude	Longitude for the vehicle position
	Bearing	Bearing for the vehicle position
	Congestion level	Specification of the congestion level that the vehicle is experiencing
	Current status	Differentiation between in-status vehicles and stopped ones
	Timestamp	The time when the position reading was taken (used for merging data)
Weather	Temperature	Minimum (Min), Maximum (Max) and average (avg) temperature of the day
	Precipitation	Precipitation of the day
	Other information	Wind and pressure-related information
Patronage	Patronage	The number of the passengers at the corresponding interval
	Time interval	The 15 minutes time interval
	Weekday	Day of the week of the trip

4.6. *Description of the features*

In this part, we summarize the features used in this study and describe them in Table 1.

4.7. *Data description and pre-processing*

Before describing the data, the first task to be done is data pre-processing. In fact, an important issue when processing agency data, as pointed out in Barbeau (2018), is its accuracy, as some agencies might fill in some data with random values just to fill the gap instead of providing missing values. In this experiment, we removed the illogical values displayed in the file. The most apparent outlier in the data is that the year of several arrival times does not belong to the period studied. Another issue is to eliminate the various redundant columns in the merged data frame. Also, we extracted another variable linked to the day of the week to enrich the study. The data used in this study covered the period from 8 January 2019 to 31 December 2020. After pre-processing, the number of instances is 116703. We have uploaded the adopted data for more details and an illustration of the fusion and pre-processing approach; the corresponding information is available on Github (https://github.com/Malek-01/Transit-delay-prediction).[6]

5. Experiment

In this section, we shall define the experimental setup for the three case studies. Subsequently, we present and discuss the results, beginning with the Canberra case study, followed by the other two case studies.

5.1. *Experimentation setup*

First, to implement the different algorithms, we adopt the Scikit-learn library (Pedregosa et al., 2011) (version 1.0.2). In particular, for DNN, we use Keras 2.2.5 with TensorFlow as the back-end for its implementation. Table 2 shows the adopted parameters for DNN, RF, GB, and LR.

[6] When dealing with delays, a specific interest should be addressed to delays causing bunching. Bunching refers to a group of two or more transit vehicles which are scheduled to be evenly spaced, running in the same location at the same time (see, e.g., Voß (2023)). After checking the data, we have found examples in which vehicles are in very close location at the same time ("timestamp"). This information can be checked in the "vehicle updates" file. Examples of vehicles affected by that are vehicle 2 and 14 (vehicle ID).

Table 2
Hyperparameters for Machine Learning Models.

Algorithm	Parameter	Value
DNN	Optimizer	Adam
	Loss Function	Mean Squared Error
	Number of Epochs	100
	Number of Hidden Layers	3
	Neurons per Hidden Layer	128
	Activation Function	ReLU
	Learning Rate	0.001
	Batch Size	32
RF and GB	Number of Trees	100
	Subsample	1.0
	Min Samples for Split	2
	Min Samples at Leaf	1
	Max Depth	3
	Learning Rate (for GB)	0.1
	Max Features (for RF)	Sqrt
LR	Fit Intercept	True
	Normalization	False

Regarding SVM, the parameter tuning process utilizes cross-validation, which is a standard technique for adjusting hyperparameters of predictive models. In this paper, we adopted 10-fold cross-validation to select the value of C (regularization factor), and ϵ in such a way to minimize the error on the validation set. More precisely, an SVM model for regression with a radial basis function (RBF) kernel is initialized with specific hyperparameters. A grid search is performed using GridSearchCV with cross-validation using the RepeatedKFold method. The goal is to find the best combination of hyperparameters from the parameter list that minimizes the negative mean absolute error for the given data. (The kernel coefficient γ is defined using the Scikit-learn “auto” function).

In particular for FS, we select the features based on the highest scores using the Scikit-learn library. The adoption of parameter tuning and FS enables a balance between overfitting and underfitting. More information on the implementation of them can be found on the Github link.

Second, in order to have a fair assessment of the ML prediction, the data must be divided into a training set and a test set. In our experiment, the first 80% of the data are used for training and the remaining 20% are used as test data. Our aim in using this division is to see how accurate we can get predictions on the (unseen) test data. The implementation of the algorithms and processing can be found in the same Github repository.

All the experiments are carried out on a computer equipped with an Intel i7-9750H and 16GB of RAM. The measures adopted in this paper are: R^2 (coefficient of determination), the mean absolute error (MAE), the root mean square error (RMSE) and, in addition, the symmetric mean absolute percentage error (sMAPE).[7] Moreover, we also compare the computational time (CPU time) of the different methods. The processing time encompasses the entire training and testing process, including model and feature selection for SVM. The algorithms are executed on a computer equipped with an Intel i7-9750H processor and 16GB of RAM.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

$$sMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}$$

Where:
y_i represents the actual target values.
$\hat{y}_i$ represents the predicted values.
$\bar{y}$ represents the mean value of the dependent variable across all data points.
n represents the number of samples in the dataset.

5.2. Prediction comparison of machine learning techniques

First, for the different algorithms described in Section 3, we display in Table 3 a comparison of their prediction capabilities in the test set. In other words, we show in Table 3 the R^2, MAE, RMSE and sMAPE values obtained by comparing the algorithm predictions

[7] This measure is adopted instead of the classical MAPE due to the potential existence of zero or near zero values (Kolassa, 2020).

Table 3
Comparison of ML predictions for the transit delay.

	SVM-FS	GB	DNN	RF	LR
R^2	**0.91588**	0.240137	0.03667	0.0895	0.01441
MAE	**0.64239**	22.62667	25.01354	23.0080	23.72723
RMSE	**9.75349**	32.53128	36.62865	28.8057	33.38596
sMAPE	**3.02225**	33.42303	143.14395	23.0080	163.51487
CPU time	1167.1683	51.5855	843.3476	429.7497	**1.2986**

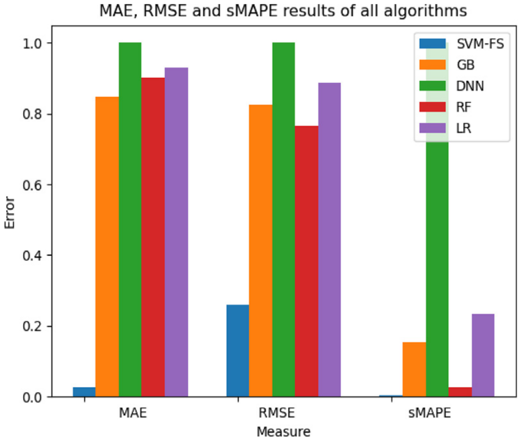


Fig. 2. Results for the different approaches.

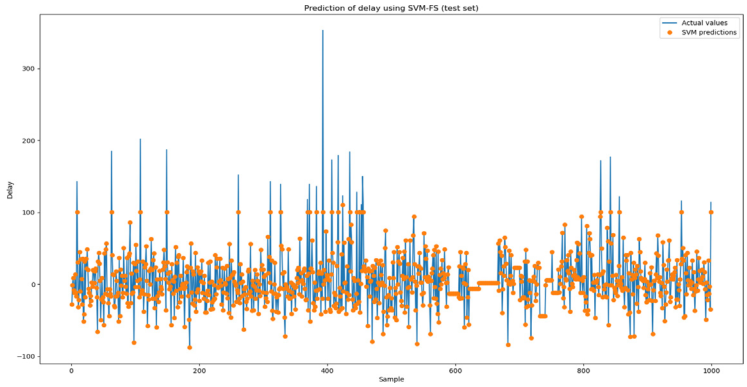


Fig. 3. Results for the SVM with FS approach.

with the real values in the test set in addition to the CPU time. Second, to give a better comparison of the results, we depict in Figure 2 the MAE, RMSE and sMAPE obtained by the different algorithms. (To be able to compare the three measures in the same graph, we have weighted the values between 0 and 1.)

We can conclude from Table 3 and Figure 2 that the predictive ability of ML algorithms differs significantly. The best results are obtained using SVM with FS. We can conclude that the adopted features are meaningful, and the parameter tuning is effective. The worst results are found when using DNN. However, as shown in Section 2, there are a number of DNN approaches and other variants are also worth investigating for this data. GB is showing acceptable results, and these results are slightly better than those of RF and LR. This is logical as LR is not appropriate to deal with some features like the timestamp (cyclical feature) and typically contradicts the linear behavior of LR. However, the main issue with SVM predictions is the CPU time. We can conclude that the proposed approach for prediction is more suitable for long-term predictions.

Moreover, to give a better overview of the results, we display in Figure 3 the predictions given by the different algorithms in the test set along with the real values for the first 1000 instances of the test set. (Due to the number of instances compared, we limit the graphs to this number to be able to detect differences between methods.)

Figure 3 illustrates a notable correlation in the prediction shape, particularly in the case of SVM with FS. This correlation is significantly stronger compared to the other methods. Note that we applied a more rigorous model selection and an FS approach for SVM, which may account for these superior results when compared to other models. In other words, the SVM model demonstrates

Table 4
Feature importance.

Feature	GB	RF	Permutation
Stop Sequence	0.09698	0.02118	0.024603
Stop Latitude	0.07650	0.07639	0.088871
Stop Longitude	0.03333	0.04580	0.085763
Stop Bearing	0.00544	0.02368	0.053025
Trip ID	0.01393	0.02368	0.08900
Arrival time (planned)	0.21874	0.20356	0.089827
Stop ID	0.01679	0.01165	0.089827
Stop Name	0.00000	0.00000	0.00000
Stop Headsign	0.00257	0.16062	0.00000
Pickup Type	0.00059	0.00048	0.00000
Drop-Off Type	0.00000	0.00000	0.00000
Date	0.162032	0.16062	0.09529
Timestamp	0.14945	0.14710	0.18900
Arrival Uncertainty	0.00000	0.00000	0.00000
Congestion Level	0.00000	0.00000	0.00000
Vehicle ID	0.00525	0.02531	0.038819
Vehicle Label	0.00262	0.02411	0.025963
Current Status	0.00458	0.02225	0.0027576
Patronage	0.08993	0.05588	0.090341
Weekday	0.00192	0.00212	0.037893
Time Interval	0.01	0.02	0.03
Temperature	0.08	0.09	0.17
Precipitation	0.00381	- 0.07	0.06456359
Other weather information	0.04	0.06	0.13

the highest generalization capacity for this case study. The chosen values for SVM parameters C and ϵ, after model selection, are 100 and 0.1, respectively.

5.3. *Feature importance and time arrival reasons*

The aim of this section is to analyze the importance of different features and to identify the most important for the prediction process.

First, we show in Table 4 the importance of the features obtained for different algorithms. Since there are many ways to measure feature importance and it can be algorithm-specific, we have adopted multiple techniques. The first two correspond to GB and RF in addition to a permutation-based feature importance approach. This last technique is more generic and we adopt it when using SVM. We chose these three algorithms because they gave the best prediction results (Table 3). More information on these techniques can be found in Brownlee (2022). The purpose of their adoption is to give a glance of the impact of the various features. More precisely, in Table 4, we show the corresponding relative value for each feature. (The different feature types are separated by lines).[8]

We can notice from Table 4 that most of the GTFS static features have a significant impact. The most important features are the planned arrival time and the date. Some of the GTFS static features like Pickup Type, Stop Headsign, and Drop-Off Type have no or negligible impact. This also applies to some GTFS real-time information such as the congestion level (this is only valid for the adopted case study). For GTFS real-time, the most important feature is the timestamp (the time the data is recorded). Also, weather features have shown a relatively significant impact. Patronage is another significant feature, and its inclusion is justifiable based on the analysis.

Moreover, in Figure 4, we show the bar plot of the Shapley values (Marcílio-Jr and Eler, 2021) of the most significant features. Shapley values have become widely used for this purpose (Wagner et al., 2022).

We can see that the results shown in Figure 4 are consistent with the discussion in the previous paragraph. We can notice that several GTFS static features (e.g. Planned arrival time, Stop sequence and trip ID) in addition to time-related features are providing the most attributions to the model. Temperature is the most important weather-related feature in this case study (‘tavg’ is the average temperature and is the most important weather feature). These key findings need to be validated and their generalizability assessed, for which we will adopt two additional case studies described below.

5.4. *Further case studies and results validation*

In this section, we conduct additional case studies to validate the results of our ML comparison and gain a better understanding of the impact of different features. The first additional case study focuses on the rail service in Dublin, Ireland. The data can be obtained

[8] For weather and patronage-related features (e.g. temperature and time interval), multiple features can correspond to a row in the table. We add them together and display the approximate sum. More specific information can be found in the GitHub link.

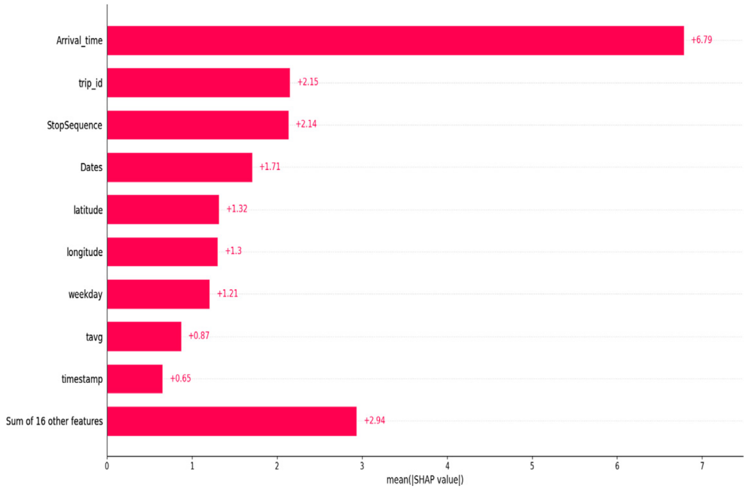


Fig. 4. Bar plot of the Shapley values of the most significant features.

Table 5
Comparison of ML predictions for the Dublin transit delay.

	SVM-FS	GB	DNN	RF	LR
R^2	0.7994	**0.9145**	0.2496	0.8402	0.0133
MAE	160.2316	**64.4737**	421.0676	76.6310	123.7272
RMSE	312.0834	**255.7368**	757.9450	349.6831	1042.3859
sMAPE	8.5231	**3.0805**	10.2085	3.7628	82.5148
CPU time	754.3458	59.4937	214.4687	116.6621	**1.5760**

from the National Transport website at https://www.nationaltransport.ie/news/attention-developers-upgrade-to-gtfs-realtime-api/. The dataset comprises 168,156 observations, covering the period between May 15, 2023, and May 21, 2023. The second additional case study concerns the Sydney public transport system. This case study is adopted in Wu et al. (2023). For this study, we utilize Sydney Trains GTFS static and real-time data obtained from the Transport for NSWs open data portal, which publishes its data regularly (Transport for NSW, https://opendata.transport.nsw.gov.au). This dataset encompasses two services, namely metro and light rail, and consists of 61,780 observations from May 9, 2023, to May 21, 2023. The merging of data in these two datasets follows a similar procedure to the initial case study, with some variations in the available features (e.g., some vehicle information is missing in the Dublin data and some stop information is not available in the Sydney data). Patronage data is presented just by the interval of the day (hour) and the weekday. For the sake of readability, we will not provide a detailed list of features but include the relevant information for our analysis. Complete information on the data and code can be found in the GitHub link mentioned earlier. It is important to note that the actual departure delay or time is excluded from both the training and test datasets. It is also worth noting that in Dublin, we use the delay as a label while for Sydney, we use the time corresponding to the GPS position as the label.

5.4.1. *Dublin case study*

We present the comparison of results for the Dublin case study in Table 5. In this table, we use the same notation as in Table 3. Table 5 demonstrates that the methods yield satisfactory results. The highest performance is achieved by GB, while RF and SVM-FS also produce good results.

Additionally, Figure 5 highlights the most significant features for the prediction process. The analysis of Figure 5 is described in conjunction with the following figure (Figure 6) and explained below.

5.4.2. *Sydney case study*

In this section, we present a similar study for the case of Sydney. Table 6 provides a comparison of the results obtained in this analysis.

From Table 6, it is clear that SVM-FS yields the best results for the Sydney case study. RF and GB also demonstrate acceptable performance. However, the issue of higher CPU time still remains, and therefore the SVM-based prediction approach is mainly useful for long-term predictions.

Furthermore, Figure 6 highlights the most significant features in the prediction process.

5.4.3. *Analysis of feature importance for additional case studies*

We observe that date-related features play a significant role in delays, with the day of the week being particularly influential. Time-related features also have a considerable impact, indicating that delays tend to occur at specific times, possibly during rush

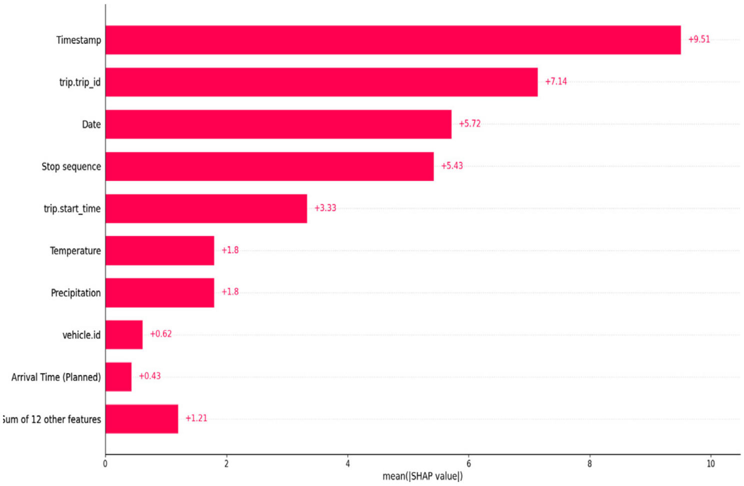


Fig. 5. Bar plot of the Shapley values of the most significant features for the Dublin case.

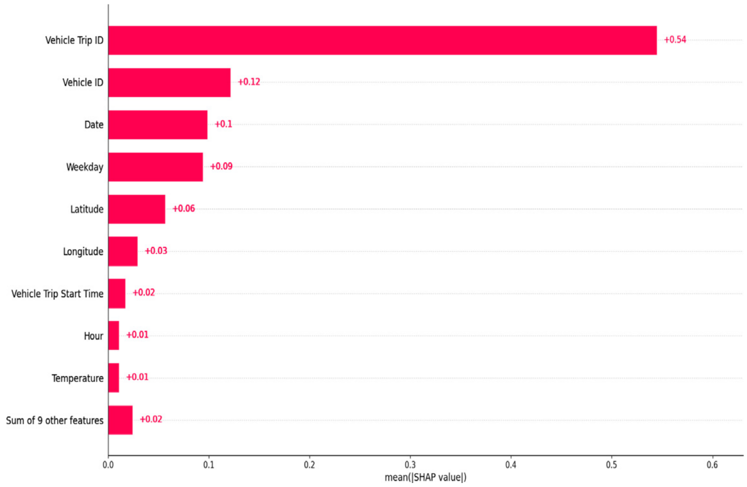


Fig. 6. Bar plot of the Shapley values of the most significant features for the Sydney case.

Table 6
Comparison of ML predictions for the Sydney transit delay.

	SVM-FS	GB	DNN	RF	LR
R^2	**0.1579**	0.1112	0.0541	0.1160	0.0704
MAE	**7104.2354**	92077.5010	19087.8473	8206.6044	8806.9821
RMSE	**1823.1453**	23754.4339	94461.1281	8186.1413	82265.2504
sMAPE	**30.5698**	34.2254	77.2145	35.3404	70.0195
CPU time	134.1849	18.0432	88.3576	52.2813	**0.3062**

hours. Even if we lack information about passenger numbers for the Dublin and Sydney case studies, we can infer that this factor is expected to influence delays, as seen in the Canberra case study. Indeed, we note that patronage-related features such as the weekday and time interval exhibit a significant impact.

Furthermore, the specific stop location also influences delays. However, it is worth mentioning that some stop information is missing in the Sydney case study, which resulted in less accurate predictions compared to the other cases.

When considering weather features, the average temperature of the day emerges as the most influential factor, potentially linked to the overall impact of different days. Precipitation also has a noteworthy effect on rail transit delays. In the Sydney case study, where no rain was recorded, precipitation does not appear to have an impact. But, in the Dublin case study, it has a significant influence on delays.

The trip ID also plays a significant role in predicting delays. This can be attributed to the fact that if a delay occurs during a trip, it tends to persist and cannot be easily compensated for in these case studies.

It can also be concluded that many of the GTFS features do not have a direct impact on rail transit delays. For example, stop-related information such as Stop Name, Stop Headsign, Pickup Type, and Drop-Off Type do not significantly affect delays. Similarly, in terms of vehicle position, features such as Vehicle Label and Congestion Level do not have a substantial impact on rail transit delays, although Congestion Level does impact delays in the case of buses (Hu et al., 2022). Additionally, weather information such as wind and pressure also do not show a significant impact on delays.

While some of these conclusions may seem intuitive, it is crucial to validate them using sufficient data, as we performed, to ensure accurate recommendations and informed decision-making.

6. Conclusion

Transit delay prediction has recently become a recurring topic in the intelligent transportation literature. Transit delay results from the influences of several factors. This study proposed a systematic approach to predicting rail transit delays, based on publicly available timetables and real-time transit service feed data, both in the GTFS format, in addition to weather and patronage (ridership) data. In this paper, we have merged the different data sources to estimate time deviations in rail transit arrivals. Then, we adopted machine learning techniques, to have an accurate prediction of the delays.

Our proposed approach based on machine learning integrates heterogeneous data sources and, hence, can serve a wide range of agencies worldwide. It can be easily adopted by a large number of cities, thanks to the increasingly applied standardized data. Machine learning approaches, as utilized in this paper, can be used to improve the quality and effectiveness of GTFS real-time feed information for each stop. In fact, the reliability of GTFS feeds provided in agencies often seemed questionable. This work provided an insight on how to improve the accuracy and effectiveness of GTFS transit feeds through machine learning.

We have also shown how to exploit open data to improve the information provided to passengers. In this context, we compared various machine learning techniques for prediction. These methods demonstrated different performances, and in this paper, we have identified SVM with feature selection as the most effective method for our case study. However, it is important to note that the SVM-based prediction approach comes with a drawback of high CPU time, making it particularly advantageous for long-term predictions.

The significance of our work lies in being among the first to achieve accurate predictions solely using open data sources for rail transit, despite the widespread adoption of machine learning for public transport prediction. This finding holds great importance as it highlights the potential for richer and more precise information in smart cities. Such information empowers planners to develop and manage public transport systems that better cater to the needs of travelers (Kuo et al., 2023). Additionally, we have identified key factors influencing delay predictions, providing transportation agencies with valuable insights to address them effectively. The most relevant features in this regard can be extracted from methods like Shapley values. Examples of such features include time and trip ID.

In essence, we have demonstrated the crucial role of open data in shaping data-driven policies aimed at preventing delay causes and delivering accurate predictions to passengers. Nevertheless, it is worth noting that despite the increasing prevalence of open data and data-driven approaches, many transportation agencies worldwide, particularly in developing countries, do not prioritize openly publishing their data when formulating policies. Additionally, national transport policies often do not emphasize the recording, reporting, and publication of data, thereby limiting data exploitation by research institutes. Our work aims to pave the way for greater motivation and utilization of open data.

From an agency perspective, the main crux of our findings is the ability to achieve highly accurate predictions on unseen data. Machine learning can be leveraged to enhance the quality of the provided service, as elaborated in the introduction.

Further research should prioritize the adoption of these approaches to optimize the transport service. While this paper demonstrates the effectiveness of using machine learning for long-term prediction, practical implementation for other prediction cases requires additional study. It is also important to leverage the insights from this paper for better optimization of public transport systems, as well as for modern transport methods like ridesharing. For example, improved prediction and optimization of delays can significantly enhance service quality, as demonstrated in Malucelli and Tresoldi (2019). Furthermore, feature explainability is essential for evaluating the robustness of the schedule, as discussed in Müller-Hannemann et al. (2022).

Additionally, conducting a comprehensive analysis of the causes of delays is necessary. In this paper, we have identified relevant features, but further investigation is required, particularly regarding the trip ID feature. It is important to pinpoint the specific trips that contribute to delays. Moreover, including other socio-economic factors that impact transit, as suggested in Bree et al. (2020) and Jevinger and Persson (2019), and conducting a more in-depth analysis of ridership, are essential steps for a holistic understanding. Another issue, that has not yet been fully explored, refers to buffer times. Different features may be influenced by the question in which way and to what extent buffer times have been incorporated as a feature (see, e.g., Ge et al. (2022a)). This might need further exploration, too.

Declarations

Funding: This work was supported by the Alexander von Humboldt Foundation.

Conflicts of interest/Competing interests: The authors declare that there are no conflicts of interest/competing interests.

Ethics approval: Not applicable.

Consent to participate: Not applicable.

Availability of code, data and material: All references to used data and code are provided in the paper. Copies of data and code are provided upon reasonable request.

Authors' contributions: All authors contributed to the study conception and design. Data collection was performed by Malek Sarhani. Data analysis was performed by all authors. All authors contributed to the writing of all versions of the manuscript.

Consent for publication All authors read and approved the final manuscript.

References

Al-Naim, R., Lytkin, Y., 2021. Review and comparison of prediction algorithms for the estimated time of arrival using geospatial transportation data. Procedia Comput. Sci. 193, 13–21. doi:10.1016/j.procs.2021.11.003.

Alzyout, M.S., Alsmirat, M.A., 2020. Performance of design options of automated ARIMA model construction for dynamic vehicle GPS location prediction. Simul. Modell. Practice Theory 104, 102148. doi:10.1016/j.simpat.2020.102148.

Barabino, B., Lai, C., Casari, C., Demontis, R., Mozzoni, S., 2017. Rethinking transit time reliability by integrating automated vehicle location data, passenger patterns, and web tools. IEEE Trans. Intell. Transp. Syst. 18 (4), 756–766. doi:10.1109/tits.2016.2585342.

Barbeau, S.J., 2018. Quality control-lessons learned from the deployment and evaluation of GTFS-realtime feeds. In: 97th Annual Meeting of the Transportation Research Board, Washington, DC.

Berggren, U., Brundell-Freij, K., Svensson, H., Wetstrand, A., 2021. Effects from usage of pre-trip information and passenger scheduling strategies on waiting times in public transport: an empirical survey based on a dedicated smartphone application. Public Transport 13, 503–531. doi:10.1007/s12469-019-00220-1.

Bree, S., Fuller, D., Diab, E., 2020. Access to transit? Validating local transit accessibility measures using transit ridership. Transp. Res. Part A: Policy Practice 141, 430–442. doi:10.1016/j.tra.2020.09.019.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Brownlee, J., 2022. https://machinelearningmastery.com/calculate-feature-importance-with-python/. Accessed: February 27, 2022.

Chondrodima, E., Georgiou, H., Pelekis, N., Theodoridis, Y., 2022. Particle swarm optimization and RBF neural networks for public transport arrival time prediction using GTFS data. Int. J. Inf. Manag. Data Insight. 2 (2), 100086. doi:10.1016/j.jjimei.2022.100086.

Daduna, J.R., Voß, S., 1995. Practical experiences in schedule synchronization. Lect. Note. Econ. Math. Syst. 430, 39–55. doi:10.1007/978-3-642-57762-8_4.

FitzRoy, F., Smith, I., 1998. Public transport demand in Freiburg: why did patronage double in a decade? Transp. Policy 5 (3), 163–173. doi:10.1016/s0967-070x(98)00024-9.

Ge, L., Kliewer, N., Nourmohammadzadeh, A., Voß, S., Xie, L., 2022a. Revisiting the richness of integrated vehicle and crew scheduling. Public Transport doi:10.1007/s12469-022-00292-6.

Ge, L., Sarhani, M., Voß, S., Xie, L., 2021. Review of transit data sources: potentials, challenges and complementarity. Sustainability 13 (20), 11450. doi:10.3390/su132011450.

Ge, L., Voß, S., Xie, L., 2022b. Robustness and disturbances in public transport. Public Transport 14, 191–261. doi:10.1007/s12469-022-00301-8.

Gilmore, S., Reijsbergen, D., 2015. Validation of automatic vehicle location data in public transport systems. Electron. Note. Theor. Comput. Sci. 318, 31–51. doi:10.1016/j.entcs.2015.10.018.

Godfrid, J., Radnic, P., Vaisman, A., Zimányi, E., 2022. Analyzing public transport in the city of Buenos Aires with MobilityDB. Public Transport 14 (2), 287–321. doi:10.1007/s12469-022-00290-8.

Google, 2021. GTFS static overview, https://developers.google.com/transit/gtfs.

Google, 2022. GTFS realtime reference, https://developers.google.com/transit/gtfs-realtime/reference/.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182. doi:10.1162/153244303322753616.

He, P., Jiang, G., Lam, S.-K., Tang, D., 2019. Travel-time prediction of bus journey with multiple bus trips. IEEE Trans. Intell. Transp. Syst. 20 (11), 4192–4205. doi:10.1109/TITS.2018.2883342.

Hu, J., Zhang, Z., Feng, Y., Sun, Z., Li, X., Yang, X., 2022. Transit signal priority enabling connected and automated buses to cut through traffic. IEEE Trans. Intell. Transp. Syst. 23 (7), 8782–8792. doi:10.1109/tits.2021.3086110.

Huang, P., Wen, C., Fu, L., Lessan, J., Jiang, C., Peng, Q., Xu, X., 2020. Modeling train operation as sequences: a study of delay prediction with operation and weather data. Transp. Res. Part E: Logist. Transp. Rev. 141, 102022. doi:10.1016/j.tre.2020.102022.

Jevinger, A., Persson, J.A., 2019. Exploring the potential of using real-time traveler data in public transport disturbance management. Public Transport 11 (2), 413–441. doi:10.1007/s12469-019-00209-w.

Kolassa, S., 2020. Why the "best" point forecast depends on the error or accuracy measure. Int. J. Forecast. 36 (1), 208–211. doi:10.1016/j.ijforecast.2019.02.017.

Kumar, B.A., Vanajakshi, L., Subramanian, S.C., 2017. Bus travel time prediction using a time-space discretization approach. Transp. Res. Part C: Emerg. Technol. 79, 308–332. doi:10.1016/j.trc.2017.04.002.

Kumar, P., Khani, A., He, Q., 2018. A robust method for estimating transit passenger trajectories using automated data. Transp. Res. Part C: Emerg. Technol. 95, 731–747. doi:10.1016/j.trc.2018.08.006.

Kuo, Y.-H., Leung, J.M.Y., Yan, Y., 2023. Public transport for smart cities: recent innovations and future challenges. Eur. J. Oper. Res. 306 (3), 1001–1026. doi:10.1016/j.ejor.2022.06.057.

Li, Z., Wen, C., Hu, R., Xu, C., Huang, P., Jiang, X., 2020. Near-term train delay prediction in the Dutch railways network. Int. J. Rail Transp. 9 (6), 520–539. doi:10.1080/23248378.2020.1843194.

Lim, A., Sharma, S., Bhaskar, A., Arkatkar, S., 2019. An open source framework for GTFS data analytics: case study using the Brisbane TransLink network. In: 41st Australian Transport Research Forum. National Academy of Science.

Liu, L., Miller, H.J., 2020. Does real-time transit information reduce waiting time? An empirical analysis. Transp. Res. Part A: Policy Practice 141, 167–179. doi:10.1016/j.tra.2020.09.014.

Malucelli, F., Tresoldi, E., 2019. Delay and disruption management in local public transportation via real-time vehicle and crew re-scheduling: a case study. Public Transport 11 (1), 1–25. doi:10.1007/s12469-019-00196-y.

Marcílio-Jr, W.E., Eler, D.M., 2021. Explaining dimensionality reduction results using Shapley values. Expert Syst. Appl. 178, 115020. doi:10.1016/j.eswa.2021.115020.

Marković, N., Milinković, S., Tikhonov, K.S., Schonfeld, P., 2015. Analyzing passenger train arrival delays with support vector regression. Transp. Res. Part C: Emerg. Technol. 56, 251–262. doi:10.1016/j.trc.2015.04.004.

Mason, L., Baxter, J., Bartlett, P., Frean, M., 1999. Boosting algorithms as gradient descent in function space. In: Proc. NIPS, Vol. 12, pp. 512–518.

Miao, Q., Welch, E.W., Sriraj, P.S., 2019. Extreme weather, public transport ridership and moderating effect of bus stop shelters. J. Transp. Geography 74, 125–133. doi:10.1016/j.jtrangeo.2018.11.007.

Müller-Hannemann, M., Rückert, R., Schiewe, A., Schöbel, A., 2022. Estimating the robustness of public transport schedules using machine learning. Transp. Res. Part C: Emerg. Technol. 137, 103566. doi:10.1016/j.trc.2022.103566.

Nair, R., Hoang, T.L., Laumanns, M., Chen, B., Cogill, R., Szabó, J., Walter, T., 2019. An ensemble prediction model for train delays. Transp. Res. Part C: Emerg. Technol. 104, 196–209. doi:10.1016/j.trc.2019.04.026.

Ni, M., He, Q., Gao, J., 2016. Forecasting the subway passenger flow under event occurrences with social media. IEEE Trans. Intell. Transp. Syst. 18 (6), 1623–1632. doi:10.1109/tits.2016.2611644.

Nimpanomprasert, T., Xie, L., Kliewer, N., 2022. Comparing two hybrid neural network models to predict real-world bus travel time. Transp. Res. Procedia 62, 393–400. doi:10.1016/j.trpro.2022.02.049.

Nithishwer, M.A., Kumar, B.A., Vanajakshi, L., 2022. Deep learning– just data or domain related knowledge adds value?: Bus travel time prediction as a case study. Transp. Lett. 14, 863–873. doi:10.1080/19427867.2021.1952042.

Olive, D.J., 2017. Linear Regression. Springer International Publishing doi:10.1007/978-3-319-55252-1.

Park, Y., Mount, J., Liu, L., Xiao, N., Miller, H.J., 2020. Assessing public transit performance using real-time data: spatiotemporal patterns of bus operation delays in Columbus, Ohio, USA. Int. J. Geogr. Inf. Sci. 34 (2), 367–392. doi:10.1080/13658816.2019.1608997.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., et al., 2011. Scikit-learn: machine learning in Python. J. Machine Learn. Res. 12, 2825–2830.

Pelekis, N., Theodoridis, Y., 2014. Mobility Data Management and Exploration. Springer, New York doi:10.1007/978-1-4939-0392-4.

Qdbus, 2014. Qingdao bus: customer satisfaction and loyalty evaluation report. http://gzw.qingdao.gov.cn/n28356025/n30142503/140813145100327435.html.

Renso, C., Spaccapietra, S., Zimànyi, E. (Eds.), 2013. Mobility Data. Cambridge University Press, New York.

Sánchez A, V.D., 2003. Advanced support vector machines and kernel methods. Neurocomputing 55 (1-2), 5–20. doi:10.1016/s0925-2312(03)00373-4.

Sarhani, M., Afia, A.E., 2016. Simultaneous feature selection and parameter optimisation of support vector machine using adaptive particle swarm gravitational search algorithm. Int. J. Metaheuristic. 5 (1), 51–66. doi:10.1504/ijmheur.2016.079112.

Sarhani, M., Voß, S., 2021. Chunking and cooperation in particle swarm optimization for feature selection. Annal. Math. Artific. Intell. 90 (7-9), 893–913. doi:10.1007/s10472-021-09752-4.

Schneidereit, G., Daduna, J.R., Voß, S., 1998. Informationsdistribution über Netzdienste am Beispiel des Öffentlichen Personenverkehrs. VDI-Berichte 1372, 217–236.

Schultz, M., Reitmann, S., Alam, S., 2021. Predictive classification and understanding of weather impact on airport performance through machine learning. Transp. Res. Part C: Emerg. Technol. 131, 103119. doi:10.1016/j.trc.2021.103119.

Shi, R., Xu, X., Li, J., Li, Y., 2021. Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization. Appl. Soft Comput. 109, 107538. doi:10.1016/j.asoc.2021.107538.

Shoman, M., Aboah, A., Adu-Gyamfi, Y., 2020. Deep learning framework for predicting bus delays on multiple routes using heterogenous datasets. J. Big Data Anal. Transp. 2 (3), 275–290. doi:10.1007/s42421-020-00031-y.

Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Stat. Comput. 14 (3), 199–222. doi:10.1023/b:stco.0000035301.49549.88.

Sun, H., Liu, H.X., Xiao, H., He, R.R., Ran, B., 2003. Use of local linear regression model for short-term traffic forecasting. Transp. Res. Record: J. Transp. Res. Board 1836 (1), 143–150. doi:10.3141/1836-18.

Tang, J., Zheng, L., Han, C., Yin, W., Zhang, Y., Zou, Y., Huang, H., 2020. Statistical and machine-learning methods for clearance time prediction of road incidents: A methodology review. Anal. Method. Accident Res. 27, 100123. doi:10.1016/j.amar.2020.100123.

Voß, S., 2023. Bus bunching and bus bridging: What can we learn from generative AI tools like ChatGPT? Sustainability 15 (12). doi:10.3390/su15129625.

Voß, S., Mejia, G., Voß, A., 2020. Mystery shopping in public transport: The case of bus station design. Lect. Note. Comput. Sci. 12423, 527–542. doi:10.1007/978-3-030-60114-0_36.

Wagner, F., Milojevic-Dupont, N., Franken, L., Zekar, A., Thies, B., Koch, N., Creutzig, F., 2022. Using explainable machine learning to understand how urban form shapes sustainable mobility. Transp. Res. Part D: Transp. Environ. 111, 103442. doi:10.1016/j.trd.2022.103442.

Wang, P., Zhang, Q.-P., 2019. Train delay analysis and prediction based on big data fusion. Transp. Saf. Environ. 1 (1), 79–88. doi:10.1093/tse/tdy001.

Wei, M., Liu, Y., Sigler, T., Liu, X., Corcoran, J., 2019. The influence of weather conditions on adult transit ridership in the sub-tropics. Transp. Res. Part A: Policy Practice 125, 106–118. doi:10.1016/j.tra.2019.05.003.

Wessel, N., Allen, J., Farber, S., 2017. Constructing a routable retrospective transit timetable from a real-time vehicle location feed and GTFS. J. Transp. Geogr. 62, 92–97. doi:10.1016/j.jtrangeo.2017.04.012.

Wu, J., Du, B., Gong, Z., Wu, Q., Shen, J., Zhou, L., Cai, C., 2023. A GTFS data acquisition and processing framework and its application to train delay prediction. Int. J. Transp. Sci. Technol. 12 (1), 201–216. doi:10.1016/j.ijtst.2022.01.005.

Wu, J., Liao, H., 2020. Weather, travel mode choice, and impacts on subway ridership in Beijing. Transp. Res. Part A: Policy Practice 135, 264–279. doi:10.1016/j.tra.2020.03.020.

Wu, W., Xia, Y., Jin, W., 2021. Predicting bus passenger flow and prioritizing influential factors using multi-source data: Scaled stacking gradient boosting decision trees. IEEE Trans. Intell. Transp. Syst. 22 (4), 2510–2523. doi:10.1109/tits.2020.3035647.

Yu, B., Lam, W.H.K., Tam, M.L., 2011. Bus arrival time prediction at bus stop with multiple routes. Transp. Res. Part C: Emerg. Technol. 19 (6), 1157–1170. doi:10.1016/j.trc.2011.01.003.

Zhang, A., Lipton, Z. C., Li, M., Smola, A. J., 2019. Dive into deep learning. Unpublished book, Accessed: October 09, 2021. https://d2l.ai/.

Zhao, J., Wang, J., Xing, Z., Luan, X., Jiang, Y., 2018. Weather and cycling: Mining big data to have an in-depth understanding of the association of weather variability with cycling on an off-road trail and an on-road bike lane. Transp. Res. Part A: Policy Practice 111, 119–135. doi:10.1016/j.tra.2018.03.001.

Zhou, M., Wang, D., Li, Q., Yue, Y., Tu, W., Cao, R., 2017. Impacts of weather on public transport ridership: Results from mining data from different sources. Transp. Res. Part C: Emerg. Technol. 75, 17–29. doi:10.1016/j.trc.2016.12.001.