Journal of Big Data Analytics in Transportation (2020) 2:275–290
https://doi.org/10.1007/s42421-020-00031-y

ORIGINAL PAPER

Deep Learning Framework for Predicting Bus Delays on Multiple Routes Using Heterogenous Datasets

Maged Shoman[1] · **Armstrong Aboah**[1] · **Yaw Adu-Gyamfi**[1]

Received: 31 July 2020 / Revised: 10 November 2020 / Accepted: 23 November 2020 / Published online: 4 January 2021

Abstract

Accurate prediction of bus delays improves transit service delivery and can potentially increase passenger use and satisfaction. To date, models developed for predicting bus delays have been restricted to single routes because of their poor performance on a wide network, due to reliance on simplistic model architectures and limited sources of data. This paper proposes a deep learning-based framework for predicting bus delays at the network level. The framework is fueled by large, heterogenous bus transit data (GTFS) and vehicle probe data. We utilize entity embeddings to enable the framework to simultaneously fit functions and learn patterns from both categorical and continuous data streams. The framework results in a single model that is able to characterize the factors influencing delays on multiple routes, for multiple stations at a time, at different times of the day, and during different seasons. A case study was conducted in Saint Louis, Missouri, with data collected over a 1-month period. The results indicate that the developed modeling framework is high performing, predicting delays for multiple stops at a mean absolute percentage error (MAPE) of about 6%. For different routes and trips, the observed prediction errors were stable across days of week, bus stops, holidays, and time of day. Although peak hour factors and the distance to bus stop influenced the model's prediction errors for some routes, the observed differences were not significant. Compared to previous research, the ability to simultaneously model continuous and categorical data with deep learning and the use of heterogenous data contributed to such high performance on multiple routes.

Keywords Transit · GPS tracking · Delay prediction · Prediction algorithm · Deep learning

Introduction

As the demand for transportation increases every day, efficient operation of transport systems is critical. The USA has seen a 13.8% rise in vehicle miles traveled on its roads over the past 10 years reaching 253.7 billion in January 2020 (FHWA 2020). During the same period, the vehicle miles traveled in the state of Missouri has more than doubled. St. Louis, the second most populated city in Missouri, experienced a decrease in public transit use by 8% in 2018 with 64% of public transport trips performed by bus (One STL 2020). With the continuous rising demand for transportation and yet decreasing use of public transit, it is of critical importance to make the service reliable and attractive to users. It has been shown that accurate estimation of bus delays can improve users' trust and willingness to pay for transit services (Dziekan 2008), as it shortens waiting times at bus stops, lowers travel time uncertainties and, overall, improves the image of bus service.

The accurate prediction of bus delays is an essential component of an Intelligent Public Transport System (IPTS). Bus arrival times are however influenced by the interaction between multiple non-linear factors including traffic conditions, incidents, dwell times, etc. These relationships are usually difficult to capture through conventional modeling techniques. Additionally, diverse datasets needed for building accurate predictive models are usually not available. While the use of big data can present valuable results, their advancements should be critically reviewed before implementation (Griffin et al. 2020). As a result, conventional

✉ Maged Shoman
mas5nh@mail.missouri.edu

Armstrong Aboah
aa5mv@mail.missouri.edu

Yaw Adu-Gyamfi
adugyamfiy@missouri.edu

[1] Department of Civil and Environmental Engineering, University of Missouri, W1024 Lafferre Hall, Columbia, Missouri 65211, USA

models have only been successful when predicting delays at the project level: for single routes, over a short period of time, one station at a time. To overcome some of these challenges, the current paper develops a framework that leverages recent advances in deep machine learning and the current generation of graphical processing units (GPUs) to characterize the relationship between the non-linear factors influencing bus arrival times on large network of routes. The developed modeling scheme learns from a large, heterogenous datasets acquired from regional transit and traffic databases. The resulting model predicts bus transit delays at the network level, for multiple routes, multiple buses, for multiple stop locations at a time and at different times of the day.

The use of artificial neural network (ANN) has proven success in prediction models where Ma et al. (2019) and Xu and Ying (2017) used it in predicting bus arrival times reporting better results compared to other methods. ANN is motivated by emulating the human brain neurons with interconnected group of nodes that has proven success in many complex problems in recent times, but it demands a larger training dataset. Other methods include support vector machine (SVM), which is a supervised machine learning, but also demands a large dataset for higher accuracies. For smaller data sizes, K-nearest neighbors (KNN) was used in building prediction models for one route because of its limited accuracy on larger datasets. Numerous models for predicting bus arrival times have been developed in the past decade. They can be categorized into four main groups: statistical, filtering (Kalman), machine learning and historical data models. Balasubramanian and Rao (2015) used seasonal variations such as time of day and day of week to predict bus arrival with cyclic variations. To incorporate changing historical traffic situations, Deng et al. (2013) predicted travel times of buses on the Bayesian network. Zhang and Teng (2013), on the other hand, explored a similar approach but incorporated bus dwell times into the prediction process. A different approach was presented by Chen et al. (2004), where they used automatic passenger counts as a key input in predicting bus arrival times and found the model powerful in modeling bus arrival times. Cats and Loutos (2016) proposed the following models to estimate transit travel times: calculate arrival time of the bus using its current location and the scheduled timetable, using time of day and day of week as an estimate to travel times and using speed information from the preceding buses on the same route as a computation of traffic conditions between and at bus stops. Traditionally, prediction models were used for one or few routes with limited use of input variables. Recent developments aimed at handling delay predictions proved successful when predicting very few routes. The current approach provides a practical means for taming the 'burden' of big data, but it limits the robustness of the model to be extended to other applications. The primary objective of this paper is to develop a framework for network-level prediction of bus transit delays. Subobjectives include the influence of different variables such as spatial influence, days of week, bus stops, time of day, holidays, distance to bus stop. The framework uses variable conditions along road segments to fuse both real-time and historical information. A deep learning-based architecture for stop-based and segment-based delays is proposed. The reliability analysis for the proposed models shall be assessed by comparing predicted and observed delays at transit stops.

Deep machine learning has achieved impressive performance on a variety of tasks; however, to the best of our knowledge, there has been no studies that used deep machine learning to develop a framework for network-level prediction of bus transit delays using heterogenous traffic conditions such as time of day and holidays. We make three main contributions to the state of the art for predicting bus transit delays. First, we present a deep learning architecture that utilizes entity embeddings to capture the influence of variables such as bus schedule, traffic congestion and road segment characteristics on bus transit delays. This enables the developed framework to learn rich information from tabular datasets with continuous and categorical attributes. Second, we design a modeling framework that can be implemented on a large scale to simultaneously predict delays on multiple routes, buses, and stations. Lastly, we develop a mechanism for conflating and fusing heterogenous datasets to assist the deep learning model to learn the underlying patterns influencing bus delays. Bus schedules, real-time bus locations, real-time traffic feed and probe datasets are used to incorporate external and partially dependent factors in the modeling process.

The remainder of the study is structured as follows: "Related works" discusses the methodology and results in literature. "Research methodology" follows discussing our developed framework contribution to the state of the art, methodology section then outlines the approaching steps to build our prediction model and data cleaning and preprocessing section follows discussing the different datasets fused together. This is then followed by "Results and analysis" where we present general and detailed model performance. Finally, "Conclusion" summarizes the research done and future direction.

Related Works

In the transportation research world, Huang et al. (2014) started with using deep learning methods for the prediction of freeway traffic world. Expanding on the prediction of freeway traffic flow, Lv et al. (2015) proposed a stacked autoencoder model. Jia et al. (2016) and Yi et al. (2017)

then introduced deep belief networks (DBF) and deep neural networks (DNN) to predict more traffic parameters such as speed. DNNs and algorithm constructions from DNN have been widely used to solve series problems (Tsoi and Back 1997) and recent works in the transportation field have been using DNNs to predict: traffic flow, traffic speed and travel time (Li et al. 2018; Duan et al. 2016; Liu et al. 2018). Short- and long-term predictions can have different impacts on the performance of a prediction model; however, in most instances long-term models seem to perform better because of fewer disturbances and more regular patterns (Mrgole and Sever 2016). Disturbances and irregular patterns are caused by impact factors such as weather changes (Zhou et al. 2017), peak times (Zhenliang et al. 2017), passenger count (Zhou et al. 2017) and traffic conditions (Brakewood et al. 2015; Rahman et al. 2018; Ramakrishna et al. 2008). Studying the general trends in literature, SVM and ANN had close results and a much better performance than KNN. DNN, which is an ANN with multiple input layers, was used by Petersen et al. (2019), and their results show high accuracy with capabilities to deal with seasonal changes. Table 1 presents various models and results by authors in different geographical locations. While existing prediction approaches were able to reflect a good performance, they are limited by one or a few routes rather than a full network.

Research Methodology

The research methodology adopted for predicting bus transit delays consists of several key components as illustrated by Fig. 1. First, different datasets are fused together using spatio-temporal conflation which maps the datasets into a unified data layer. The second component involves constructing route shapes from the fused datasets, and in the third component we pre-process data where we clean and merge files to calculate parameters such as bus speed, shape distance traveled and delays. The fourth and fifth components involve the use of our impact factor (traffic conditions) and conflating our stops to the route segmented along its road units. The inclusion of traffic impacts on scheduled transit is discussed in the sixth component. The seventh and last component discusses our prediction model and the setup of our training and testing data.

Data

Three main sources of data were used to develop the predictive framework: bus transit schedule data from GTFS, real-time bus location data, and cell-phone probe data from the geographical mapping software: Regional Integrated Transportation Information System (RITIS).

Real-Time Bus Locations

The real-time location of buses was obtained through an API provided by St. Louis Metro (STL Metro 2019) Requests for data were made every 30 s from December 13th 2019 to January 8th 2020. Each request returns information on the bus ID, route ID, trip ID, vehicle ID, vehicle label, start date, start time, latitude, longitude and time stamp. The way data is collected and stored can be thought of as an image taken every 30 s of all buses operating in the network. For a total of over 5 million records archived with a size of 1 GB, around 70 routes were used and 200 buses served at one time. Figure 2a shows the map for our case study area colored by routes and Fig. 2b shows the same map but colored by traffic speeds on routes with colors ranging from red (speed = 0mph) to green (speed = 40mph).

GTFS Data

The General Transit Feed Specification (GTFS) data contains schedules and arrival times for all buses serving the STL metro region. Compared to the bus location data, the GTFS data is small in size and static in nature, with very few updates every month. GTFS data was requested for the same period during which the real-time feed data of buses was collected. The GTFS data consists of text files of transit aspects such as: stops, stop times, routes, trips, schedule, shapes, calendar and agency. The text files were merged together to output scheduled arrival times at stop locations through trips along their respective routes. In the current study, bus delays are estimated by comparing bus schedule data with real-time bus location data.

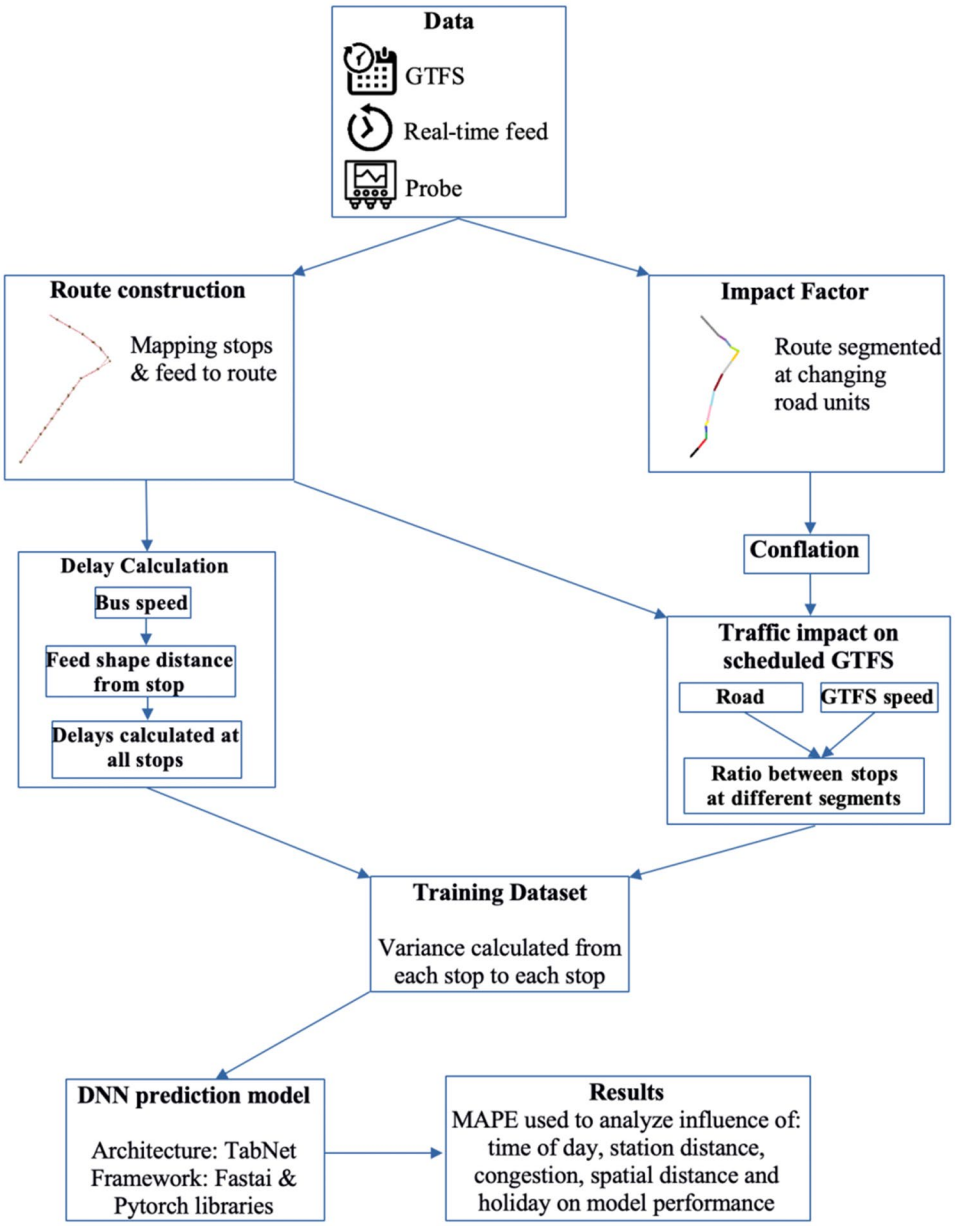
Probe Data

This data provides traffic data such as speed and travel times for about 35,000 road segments at a resolution of 60 s throughout the state of Missouri. The data is provided by a third-party provider: Regional Integrated Transportation Information System (RITIS). Probe data provides us with road segment information such as traffic speed and travel time. Traffic speed is later used in our model when compared to bus travel speed. Probe data includes lots of road information, but our data of interest is speed and measurement time stamp which will be used as inputs to our model while tmc code will be conflated when mapping bus stops (point) to road segments (line). Table 2 presents a sample (one row) of the real-time transit feed, GTFS dataset after merging the text files, and probe data.

Table 1 Detailed literature review of previous work

Author	Title	Location	Methodology	Findings
Chien et al. (2002)	Dynamic bus arrival time prediction with artificial neural networks	New Jersey	Two (ANN) trained by link-based and stop-based data Adaptive algorithm to adapt to the prediction error in real time and improve prediction accuracy	RMSE of the enhanced link-based ANN increases from 14.5 to 72.26 s. RMSE of the stop-based ANN increases from 19.71 to 61.43 s. This indicates that the enhanced link-based ANN outperforms the stop-based one if the number of intersections between a pair of stops is relatively small
Yu et al. (2011)	Bus arrival time prediction at bus stop with multiple routes	Hong Kong	Support vector machine (SVM), ANN, *k* nearest neighbors algorithm (*k*-NN) and linear regression (LR)	MAPE: (SVM: 4.49–13.23%, ANN: 6.84–15.11%, k-NN: 6.94–16.89%, LR: 6.78–24.99%)
Kumar et al. (2014)	Pattern-based bus travel time prediction under heterogeneous traffic conditions	Chennai, India	Kalman filter with pattern-based model and heterogeneous traffic states for 1 route	MAPE = 12.34 s for a subsection index of 500 m with an average travel time of 100 s
Bai et al.(2015)	Dynamic bus travel time prediction models on road with multiple bus routes	Shenzen, China	SVM–Kalman filter algorithm on multiple bus routes	SVM–Kalman outperforms pure SVM, ANN and Kalman models with MAPE for one road segment reduced from 10.24% to 4.33%
Yang et al. (2015)	Bus arrival time prediction using support vector machine with genetic algorithm	Shenyang, China	Genetic algorithm–support vector machine (GA–SVM) using character of the time period, the length of road, the weather, the bus speed and the rate of road usage	GA–SVM may be globally optimal, while the ANN models are a local optimal solution. The performance of the GA–SVM has higher accuracy than SVM
Xu and Ying (2017)	Bus arrival time prediction with real-time and historic data	Hangzhou, China	SVM, ANN for a stop-based model with heterogeneous traffic state for multiple routes	SVM and ANN had very close results MAE: (SVM: 37, ANN: 38) RMSE: (SVM: 25, ANN: 26)
Celan and Lep (2018)	Bus arrival time prediction using bus network data model and time periods	City of Ljubljana and Maribor	Prediction of travel times using average travel times in each time period	MAE: (Maribor: 55.7 s, Ljubljana: 63.7 s) MRE: (Maribor: 14.4–17.7%, Ljubljana: 10.5–12.1%)
Petersen et al. (2019)	Multi-output deep learning for bus arrival time predictions	Greater Copenhagen area	A multi-output, multi-time-step, deep neural network using convolutional and long- and short-term memory (LSTM) layers is used for travel time, and simplistic models are used for dwell time and seasonal components	RMSE: (historical average: 6.05–6.28%, linear reg: 4.77–5.45%, multi-modal: 4.38–4.53%)
Ma et al. (2019)	Bus travel time prediction with real-time traffic information	Xi'an, China	Comparing the performance of SVM, ANN, and k-NN	SVM performed best, followed by ANN and K-NN. ANN preformed similar to SVM at smaller transit segments. Best route MAE (s): stop-based: 61.76 link-based: 56.88 segment-based: 40.07

Fig. 1 Overview of approach



Route Construction and Delay Calculation

Inspection of the collected real-time feed showed that some trips had location errors where the record would include multiple points very far from the scheduled stops before ending its service. Figure 3 presents an example of the error in a trip when comparing the schedule to the real-time feed. Missing feed points near a stop can indicate one of the following:

- Feed error not returning feed at that time.
- Bus passed the stop quickly.
- Stop was ignored by the driver.

Our first approach in calculating delays is building a program with Python and using the Vincenty library that calculates the geographical distance between two points with extreme accuracy. We started by creating a buffer of radius 0.01 miles, since buses do not usually stop exactly at the stop location, around a stop and returning to the closest feed point. Closest feed point must be on the same route and trip as the stop. While this approach is accurate in determining when the bus actually stopped at the stop, it was deemed unsuccessful since most stops did not have any feed points within the set buffer. This can be explained by the reasons mentioned earlier. To mitigate such a problem, we decided to eliminate the use of a

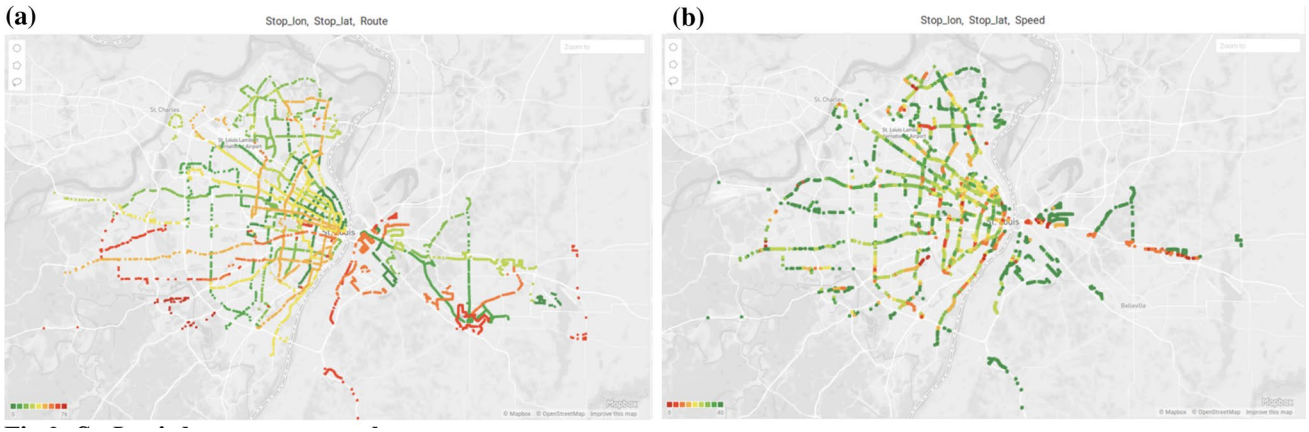


Fig. 2 St. Louis bus stops network

Table 2 Sample of integrated datasets

Real-time transit feed data		GTFS data		Probe data	
bus_id	1	stop_id	14,818	tmc_code	119P04640
trip_id	2592413	stop_lat stop_lon	38.646949 − 90.309436	Measurement_tstamp	2019-12-13 00:00:00
start_time	14:19:00	arrival_time	19:45:00		
start_date	20191213	stop_name stop_sequence	MALLINCKRODT CENTER 2	speed	68
route_id	15784	Direction direction_id	CC 0	average_speed	64
Latitude	38.536922454834	block_id	a_652	reference_speed	71
Longitude	− 89.8747482299805	trip_id	2520433		
Timestamp	2019-12-13 14:21:41-06:00	route_id	15302	travel_time_minutes	0.58
vehicle_id	2077	shape_dist_traveled	539.646516		
vehicle_label	21 Scott AFB-Main Base Shuttle—CLOCKWISE	trip_headsign	TO CENTRAL WEST END TC	data_density	A

Fig. 3 Scheduled stops (red) compared to real-time feed points (blue)

buffer radius and instead return the closest feed point at all stops. ArcGIS is then used to map all the stops and calculate the shape distance to the closest feed point found which is then used to calculate the bus speed and expected delay at stop. Delays at stops are calculated using the following formula:

$$D = A_t - T_s,$$

where D represents the delay calculated in minutes with positive values indicating bus arriving early and negative values indicating a late arrival to the stop. A_t represents the expected arrival time of bus which is provided by the GTFS data and T_s represents the time stamp of the actual arrival time of bus provided by the real-time bus location data.

Figure 4 presents an example of the shape created by ArcGIS for one route and one trip. It is worth noting that the shape created is not necessarily a representation of the actual route taken by the bus.

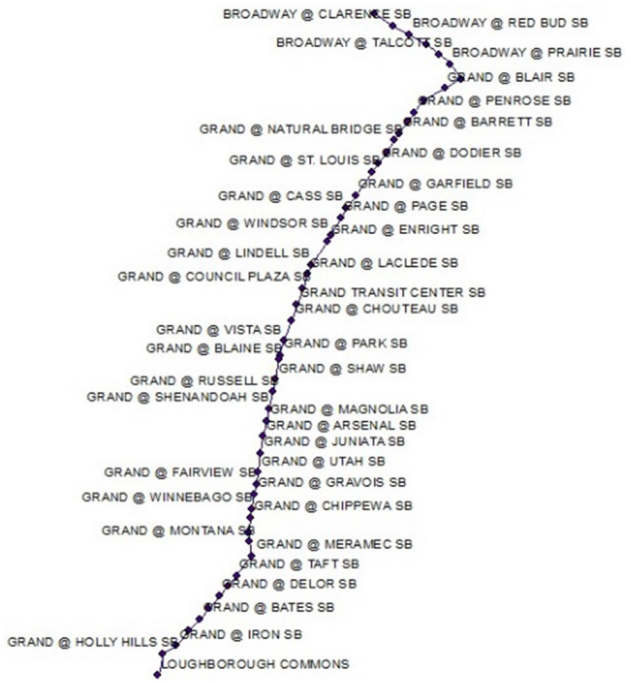


Fig. 4 Route shape created along bus stops

To understand the performance of our prediction model, it is important to understand the distribution of delays at different days or times of day. Figure 5 presents analytics of delays at different days of the week and hours of the day with buses arriving late to a stop as negative delays, while buses arriving early to a stop are positive delays. Figure 5a presents the heatmap of average delays along various days of the week and times of the day, which can be used as a reflection of the traffic congestions caused by high delays. The heatmap colors range from green (delay = − 0.5) to red (delay = − 5.00). Weekdays seem to have much higher average delays compared to weekends, explained by the fact that traffic is generally much less during weekends. Inspecting the varying average delays during times of day shows higher delays during very early hours of weekdays explained by buses not operating due to no demand. Common weekday delays can be observed at peak times between 6 and 9am and 4 pm and 8 pm as expected since the transportation network is at its peak use. Figure 5b presents a combo analytics of the average, standard deviation, minimum and maximum delays along various hours of the day.

Conflation

Point to line conflation was carried out to integrate the bus location (point) and probe (line segment) datasets. First, we used ESRI's "Generate Near Table" tool to generate a point to line proximity table that maps each bus location to the corresponding road segment using a buffer of 100 feet. Since the "Generate Near Table" tool does not take direction and name of road into consideration, generated segment pairings were post-processed to filter out pairings with opposite directions and different road names. A road name similarity function was developed to verify if paired road segments had similar road names

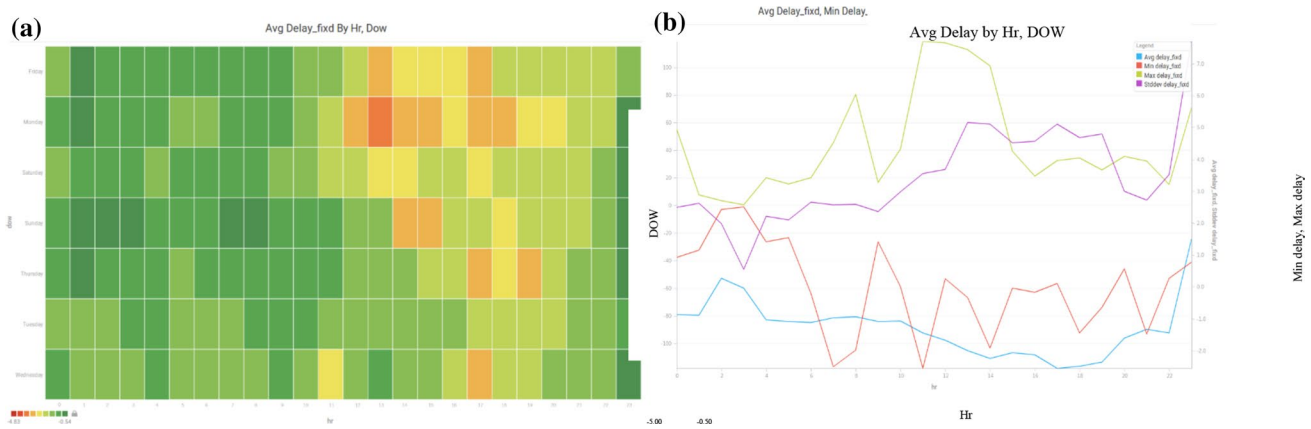


Fig. 5 Analytics of delays along days of week and day hours: **a** Heatmap, **b** combo analytics

and bearing. The final metric for accepting or rejecting proximity-based pairings is designed as follows:

$$si = k_1 d + k_2 s + k_3 n,$$

where k_1,k_2, k_3 are constants whose sum equals 1. d is direction similarity, s is text similarity and n is road number similarity. If $si > 0.8$, pairing is accepted, else, rejected. Table 3 shows the output conflated table. About 95% of bus locations were conflated automatically to probe road segments using this technique.

Training Dataset

As mentioned earlier in the probe data section, average speed and real-time speed of traffic data were provided. Both speed components were used in building the training dataset. Since each stop has its own scheduled or GTFS speed, each segment also has its own road speed. We estimate the impact of road speeds on transit by calculating (GTFS speed/road speed) at each stop. Figure 6 presents a sample transit route with stop-based and segment-based speeds with average and real-time speeds for each model.

In the seventh component, we built the training model from the real-time transit, GTFS (average and real-time traffic) datasets and the results from the previous components: delays and conflation. After stops were mapped to each road segment they belong to, the arrival times at stops are compared to the measurement time stamp at that particular road segment so that the traffic speed at the nearest time stamp is returned. For each trip, the GTFS to road speed ratio at the stop is calculated along the trip. The reason behind training the model on such data is because such attributes consider the change in speeds from one stop to another and one segment to another, achieving our goal in predicting stop-based and segment-based delays. Figure 7 presents point-map analytics of the GTFS to road speeds along all routes in our network. The point-map colors range from green (gtfs/road speed = 1) to red (gtfs/road speed = 0). Smaller ratios indicate buses traveling at a speed slower than traffic.

For our training data, we use 1-h historical average and variance of speed ratios (gtfs/rd). Our historical data also includes the past trips on historical similar days of week on the same route. Our data for each trip is then restructured so that prediction can be made at each stop from preceding stops using the variance calculated from speed ratios between stops. Figure 8 provides an easy explanation of the calculated speed variances between stops. Table 4 presents a sample of the training dataset (units in second row), followed by an explanation for each column.

Table 3 Sample of conflated table

TMC	START_LAT	START_LONG	END_LAT	END_LONG	ROAD	DIRECTION	Bearing	Stop_id	stop_lat	stop_lon	direction	trip_id	route_id	direction_id	arrival_time
119–13694	38.64603	− 90.30175	38.64738	− 90.31581	FORSYTH BLVD	WESTBOUND	175	14818	38.646949	− 90.309436	CC	2520433	15302	0	19:45:00
119 + 13695	38.64738	− 90.31581	38.64603	− 90.30175	FORSYTH BLVD	EASTBOUND	355	14818	38.646949	− 90.309436	CC	2520433	15302	0	19:45:00
119 + 13695	38.64738	− 90.31581	38.64603	− 90.30175	FORSYTH BLVD	EASTBOUND	355	15018	38.64612	− 90.303546	EB	2520433	15302	0	19:46:00
119–13694	38.64603	− 90.30175	38.64738	− 90.31581	FORSYTH BLVD	WESTBOUND	175	15018	38.64612	− 90.303546	EB	2520433	15302	0	19:46:00
119P13695	38.64603	− 90.30175	38.64598	− 90.30122	FORSYTH BLVD	EASTBOUND	355	2732	38.646101	− 90.301151	NB	2520433	15302	0	19:47:00

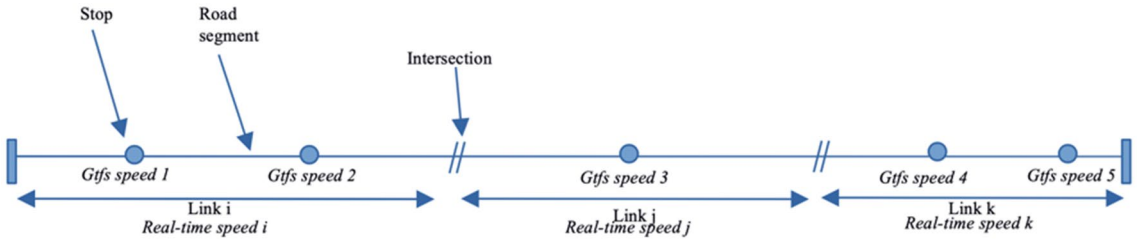


Fig. 6 Transit route segment with stop-based and link-based speeds

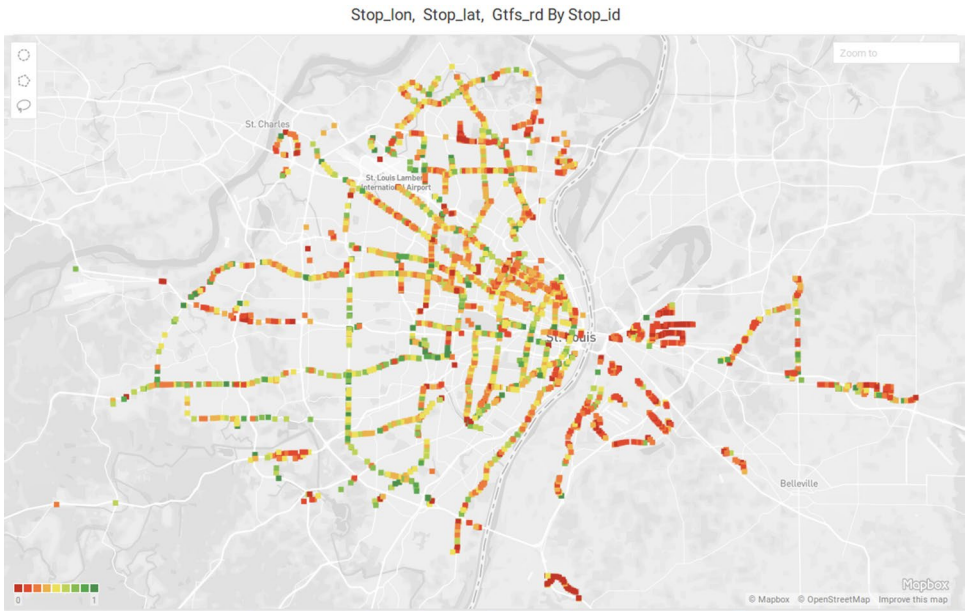


Fig. 7 Point-map analytics of GTFS/road speeds along routes on network

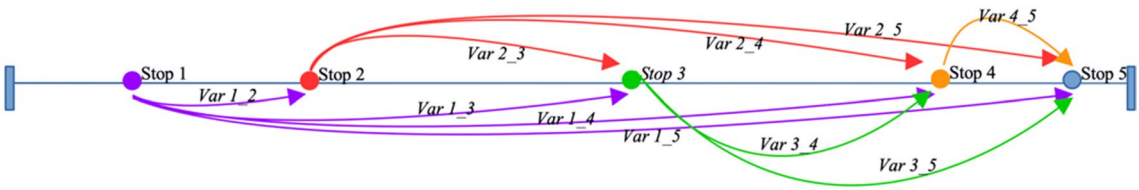


Fig. 8 Transit route segment with variances calculated between stops

Model Building Background

Arik and Pfister (2019) proposed the first self-supervised-learning DNN architecture for tabular data called TabNet. Tabular data consists of continuous and categorical features. The developed architecture analyzes continuous features with their raw values and maps categorical features with trainable embeddings. The embedding increases the learning ability of the categorical features. TabNet was designed to learn using a decision-tree-like mapping to inherit important characteristics of both tree-based methods and DNN-based methods. Unlike tree-based methods, TabNet uses gradient descent to train the raw data without feature preprocessing. This enables flexible representation and integration into end-to-end learning. Also, TabNet uses sequential attention to choose which features to reason from at each decision step. This enables TabNet to interpret and better learn relevant features in the dataset. The results from the study indicates that TabNet outperforms previous studies across tabular datasets from different domains.

The current model developed for predicting network-level bus delays is based on the TabNet architecture explained above. The model was implement using Fastai's deep learning framework and Pytorch libraries. The model architecture

Table 4 Sample of training dataset

Route_id	Trip_id	Dow	Start_date	Time	From stop	To_stop	tmc_code	GTFS/ rd__30 min	GTFS/ rd_60 min	var_30 min	var_60 min	hist_avg	hist_var	observed_ GTFS/rd	delay
Id	Id	Day	Date	Time	count	count	Id	ratio	ratio	ratio	ratio	ratio	ratio	ratio	mins
15776	2591589	Friday	2019-12-13	14:02:00	1	2	119-04612	0.2504	0.2504	0.0331	0.0331	0.2504	0.0331	0.2504	− 20.32
15776	2591589	Friday	2019-12-13	14:02:20	1	3	119-04612	0.2504	0.2504	0.0331	0.0331	0.2504	0.0331	0.2504	− 20.31
15776	2591589	Friday	2019-12-13	14:02:35	1	4	119-04612	0.5075	0.5075	0.0331	0.0331	0.5075	0.0331	0.5075	− 19.31
15776	2591589	Friday	2019-12-13	14:03:00	1	5	119-04612	0.3633	0.3633	0.0166	0.0166	0.3633	0.0166	0.3638	− 18.63
15776	2591589	Friday	2019-12-13	14:03:10	1	6	119-04612	0.1393	0.1393	0.0248	0.0248	0.1393	0.0248	0.1393	− 18.63

Route_id id of the path along network that the bus takes during its trip, *Trip_id* id of the trip performed along a route, *dow* day of week where the trip is performed, *Start_date* date when the trip was performed, *hr* hour when the trip was performed, *min* minute when the trip was performed, *sec* second when the trip was performed, *from_stop* origin stop where the prediction is made from, *To_stop* destination stop where the prediction is happening, *tmc_code* id of the road segment along the path/route. Each route is made from a collection of road segments, *gtfs/rd_30 min* ratio of bus speed to road speed of the trips performed on the same route over the past 30 mins, *gtfs/rd_60min* ratio of bus speed to road speed of the trips performed on the same route over the past 60 mins, *var_30min* variance of ratio of bus speed to road speed of the trips performed on the same route over the past 30 mins, *var_60 min* variance of ratio of bus speed to road speed of the trips performed on the same route over the past 60 mins, *hist_avg* historical average of the ratio of bus speed to road speed of past trips on past similar day of week on the same route (so if we are predicting on a Friday, the historical will be the Fridays before the current Friday), *hist_var* the historical variance of the ratio of bus speed to road speed of the trips performed on the same route over the past days of week, (so if we are predicting on a Friday, the historical will be the Fridays before the current Friday), *Observed_GTFS/rd* the actual observed gtfs/rd value at the current stop (stop_sequence), *Delay* the actual observed delay value at the current stop (stop_sequence) (min)

used in this study first learns an embedding of unique routes, trips, stops, road segments, time of day and day of week signals. Subsequently, the learned embeddings are combined with other continuous information from historical delay trends, prevailing traffic conditions and other unique local representations which may influence bus transit delays. For each categorical attribute (Table 5), the embedding layer learns an N by four-dimensional vector which characterizes the influence of the variable on the model outcomes. The number of rows in the embedding layer is dependent on the number of unique values in the categorical data variable. The TabNetModel takes in input data as a pandas DataFrame and output one-dimensional tensor. The continuous variables (Table 5) on the other hand are passed through a series of dense layers and their activation is combined with outputs from embedding layers to predict outcomes. That is, a TabularPandas data frame that contained the training dataset and the validation dataset were built. The tabular data frame was parsed through the TabNetModel. The TabNetModel was also fed with the embedding matrix, batch size of 256 and length of the continuous variables. The best learning rate was found to be 3e-2. The number of epochs used to train the model was 70, which took approximately 490 min to complete. Finally, the metric used for assessing the performance of the model was accuracy Fig. 9.

Entity Embeddings

Typically, models which have both continuous and categorical variables as input variables are designed to have one function to fit both variables during model development. The use of entity embedding give room to fit different functions to both continuous and categorical variables. This helps to improve the training effect of the categorical variables. The embedding model structure used in the current paper is shown in Fig. 10. The embedding layers serve as an alternative to the one-hot encoding. For very large cardinality, one-hot encoding for categorical variables becomes practically difficult. Embedding, therefore, reduces the vector's dimension for easy analysis by putting words with similar meaning together. The study used the category variables (Table 5) to create the embedding matrix. Table 6 shows the algorithm used to generate the embedding sizes.

Model Evaluation

The final model developed in our paper utilizes all input variables in Table 5, since they were all highly influential in predicting the outcome (delay). Input variables were added to the model one at a time. The accuracy of the model was used as a metric for determining variables which were influential in the prediction of the outcome. Insignificant variables were decided based on their little to no impact on the

Table 5 List of continuous and categorical variables

Continuous variables	Categorical variables
Historical average	Route identification number
Historical variance	Trip identification number
Variance of ratio of bus speed to road speed of the trips performed on the same route over the past 30 mins	Day of the week
Variance of ratio of bus speed to road speed of the trips performed on the same route over the past 60 mins	Hour of day
Ratio of bus speed to road speed of the trips performed on the same route over the past 60 mins	Minutes
Ratio of bus speed to road speed of the trips performed on the same route over the past 30 mins	Stop sequence
	Road segment identification number

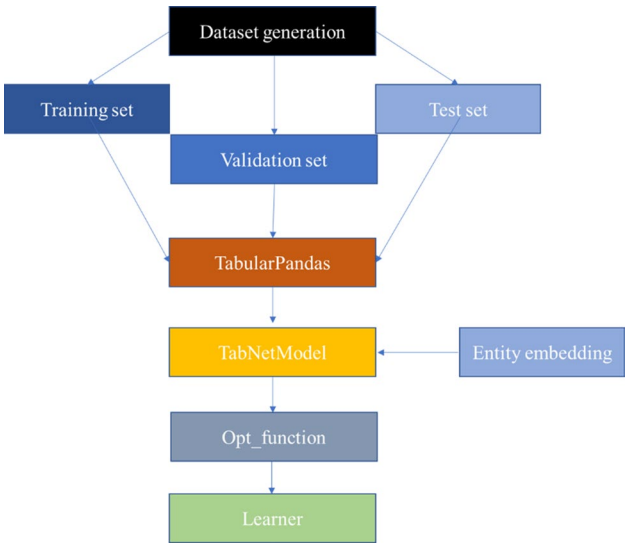


Fig. 9 Flowchart of model development

model accuracy. The overall performance of the developed model was assessed using accuracy. The accuracy of the model refers to the number of times the estimated delays are of the same values as the observed delays expressed as a fraction of the size of the dataset. The accuracy of the developed model was 0.504. This means that approximately 50.4% of the time, the model accurately estimates the delays observed. The developed model had a training loss of 1.576 and a validation loss of 1.576. As a result, the model is said to be performing equally well on both the validation dataset and the training dataset. The model converged after 70 epochs which is shown by the plot of the losses in Fig. 11.

Results and Analysis

This section evaluates the performance of the trained prediction model against a test dataset which consisted of bus feed speed, road speed and observed delays at different routes, trips, segments, stops, days of week and hours of day. The test dataset consists of the last week (7 days) from the data collected for 1 month. First, we will present the general results for the model performance, followed by the influence of different conditions: time of day, station distance, congestion, spatial distance and holidays. While predicting bus transit delays, prediction errors from the models are calculated from the observed bus transit delays and shall be used to justify prediction results. Table 7 summarizes the model performance at peak hours, non-peak hours, weekdays and weekends. Mean absolute percentage error (MAPE) is the performance metric we use in evaluating our model because of its very intuitive interpretation in terms of relative error to our predicted values, with smaller MAPE values indicating higher model accuracy. MAPE for the full tested dataset is 5.32%, which means that on average the predicted values are off by 5.32%, comparable to the results of previous studies by Yu et al. (2011) for multiple routes and by Kumar et al. (2014) for one route. We can conclude that our prediction model was robust and not strongly influenced by time of day or day of week, where we still had a MAPE of less than 6%.

Influence of Time of Day

To understand the performance of the prediction model at different hours of day and days of week, we plot MAPE at hours of day as presented in Fig. 12a and a heatmap colored by MAPE per hour at different days of week. The heatmap colors range from green (MAPE = 0) to red (MAPE = 18%). The minimum MAPE occurs at 4 am when buses are operating at minimum service. Maximum MAPE occurs at hour 16 which is the evening peak hour when delays are also the highest as presented earlier in Fig. 5b. Inspection of this MAPE spike in Fig. 12b shows that evening time across all days of week is when MAPE is higher compared to other times of delay, which indicates that our prediction model is sensitive to high delays occurring on multiple days.

To dig deeper into the prediction errors at the stop level, we use a heatmap to study the average and standard deviation of MAPE at stops (stop id) during hours of day as presented in Fig. 13a, b, respectively. The map shows a sample for a few of the stops we tested and the range of colors on the map are from green (MAPE = 0) to red (MAPE = 18) with

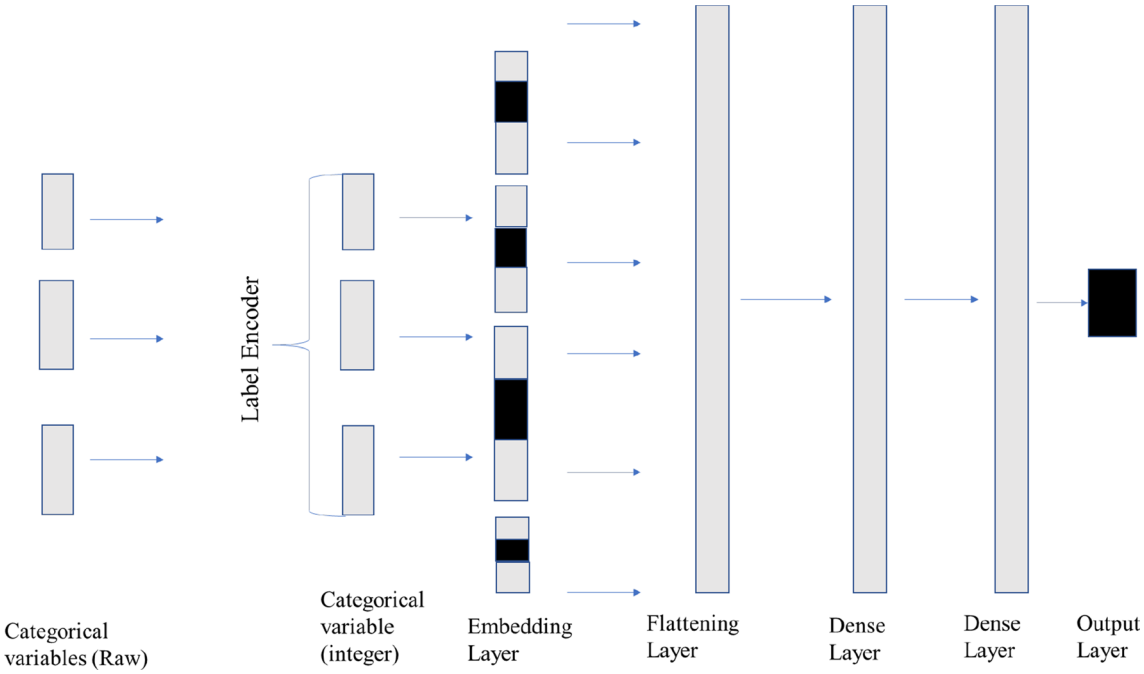


Fig. 10 Model structure of embedding

Table 6 Algorithm for generating embedding sizes

Algorithm
Input = cat_vars
Output = emb_sz
Data = data_1
Initialize cat_vars = []
For i **in** cat_vars
len = len(data_1[i].cat. categories) + 1)
cat_name = i
cat_sz = [cat_name, len]
Initialize emb_szs = []
For j in cat_szs
len = j (Altinkaya and Zontul 2013)
emb_ = (len + 1)/2
emb_len = min (50, emb_)
emb_szs = [len, emb_szs]

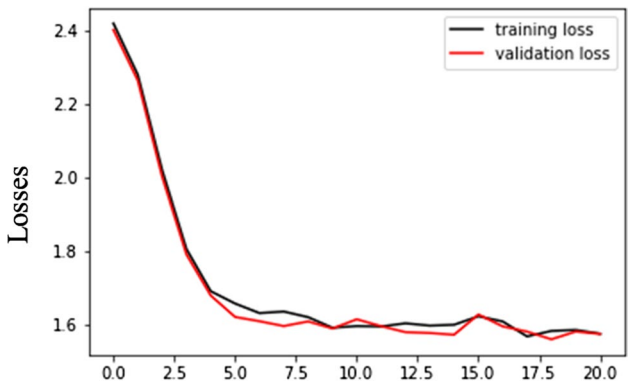


Fig. 11 Model accuracy

Table 7 Prediction model performance

Data (record count)	MAPE (%)
Full dataset (611,325)	5.32
Peak hours (261,228)	5.94
Non-peak hours (350,097)	4.95
Weekdays (605,774)	5.32
Weekends (5551)	5.17

blank fields indicating stop not used. It appears that when high errors (red) occurred at certain stops, they happened at the same hour. High average error with high standard deviation at the same stop and time means that it was an isolated case, indicating that it is a stop or data problem rather than a model problem. If a higher MAPE is observed but the standard deviation is low, it will indicate that the inconsistent underperformance is coming from the model.

Influence of Station Distance

Figure 14a presents the average delays vs average predictions at varying stop distance on the primary axis and the record count of predictions on the secondary axis. Stop distance refers to the number of stops between a specific stop and the stop where delay prediction is made. The record count of predictions was added to understand the amount of data influencing the average. Generally, the difference between predicted and observed was quite consistent at the higher

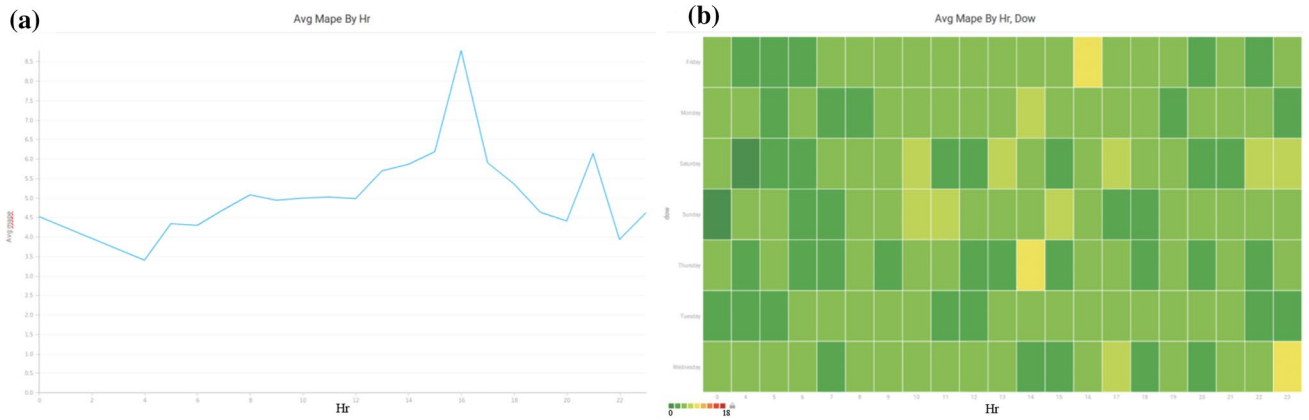


Fig. 12 MAPE between observed and predicted delays: **a** MAPE at varying times of day, **b** MAPE heatmap at varying days and times

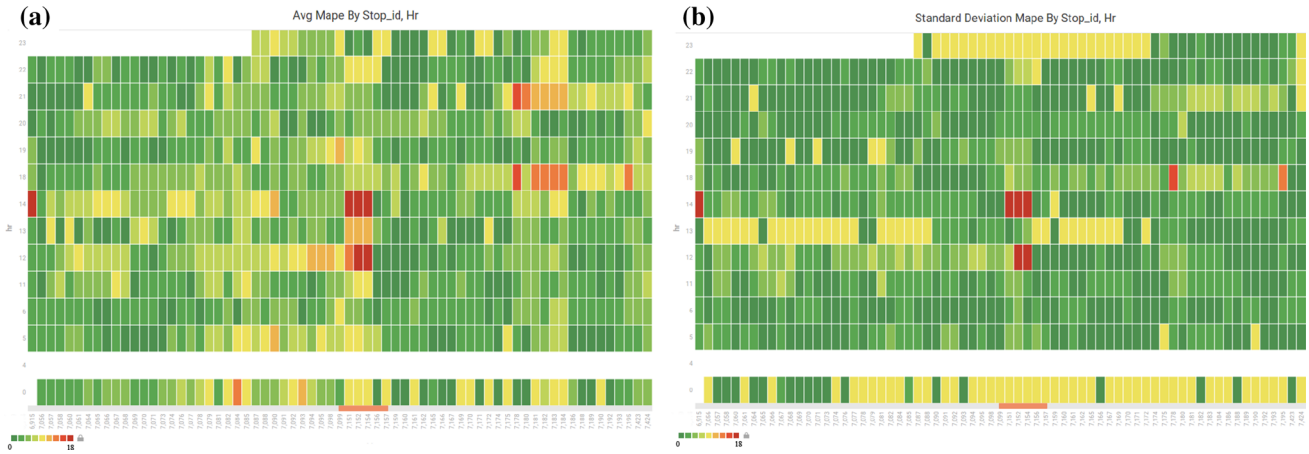


Fig. 13 MAPE along predicted stops: **a** average, **b** standard deviation

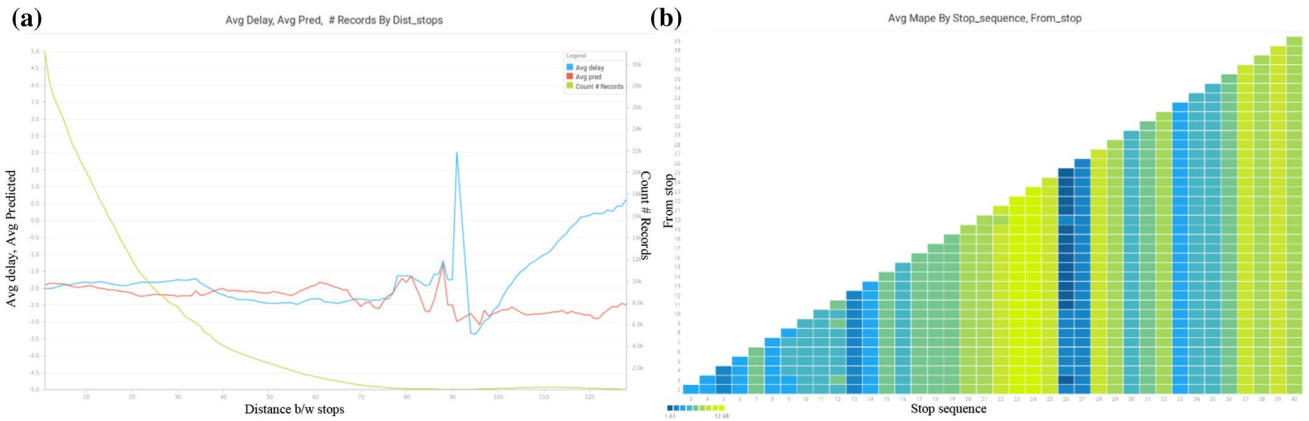


Fig. 14 **a** Delays vs. predictions at varying stops distance (primary axis) with predicted record count (secondary axis), **b** example of MAPE at various stop distances

record counts, which concludes that our model is robust irrespective of distance to prediction station. Although, in general, the prediction slightly increases as the distance of stop station increases, the average delay prediction error stays within 0.5–1% (of the observed delay) when predicting up to 80 stations ahead. It is worth noting that the average number

of stops per route is around 35 stops and very few routes have a higher number of stops as shown on the secondary axis, which explains the higher spikes occurring at very high stop distances, since there is much fewer training data and such routes usually operate at a much smaller frequency. To analyze the prediction accuracy at station distances, Fig. 14b presents a sample of the change in MAPE at varying origin and prediction stops. The heatmap colors range from dark blue (MAPE = 0%) to light green (MAPE = 13%). It appears that in most cases, the error was consistent regardless of the origin stop where prediction is happening from, indicating the models' consistent performance irrespective of the stop distances.

Influence of Congestion

To understand the influence of congestion on our prediction model, we study how MAPE is influenced by peak (between 6 and 9am and 4 pm and 8 pm) and non-peak hours as presented in Fig. 15a, b respectively. It can be observed that in general the MAPEs are slightly better in less congested conditions (non-peak hours). The overall influence of congestion is however not significant, as it falls below 0.5% of the error. The model is therefore robust to peak hour factors. There were a few anomalies, however: for example, on Monday, the MAPE during non-peak hours was about 0.5% higher. The lack of real-time bus feed data during non-peak hours could have contributed to this discrepancy.

Spatial Influence

The spatial influence on the prediction model at peak and non-peak hours can be studied through Fig. 16a, b, respectively. The geo-heatmap is a plot of stops grouped by polygons and colored by MAPE to understand the model performance at different regions. The colors range from red (MAPE = 0%) to green (MAPE = 13%). While no clear conclusion can be made from the maps about which areas cause higher errors, an observation can be made about the higher errors (darker colors) on the city outskirts. Trips generally begin from downtown and end at city outskirts. For long duration trips which end in the city outskirts, it is more challenging (due to limited data) to predict fluctuations in traffic conditions that will be encountered by the bus.

Holiday Effect

Holiday travel patterns are significantly different from normal days. It was therefore important to evaluate the model's performance during such periods. Since data was collected from December through the first week of January, two big holidays were captured. The model was trained on data captured during Christmas holiday, then its performance was evaluated by testing on bus data collected during New Year's holiday. Figure 17 presents the MAPE pie chart of the tested days sized by the magnitude of MAPE and colored by the average delay for each date with colors ranging from green (delay = − 0.26 min) to red (delay = − 2.31 min). We can see that the holiday on Jan 1, 2020 still had average delays comparable to other days and that the prediction model results were not influenced by the holiday, since errors were within 0.5% indicating that changes in schedule or operation because of a particular event do not influence the model performance.

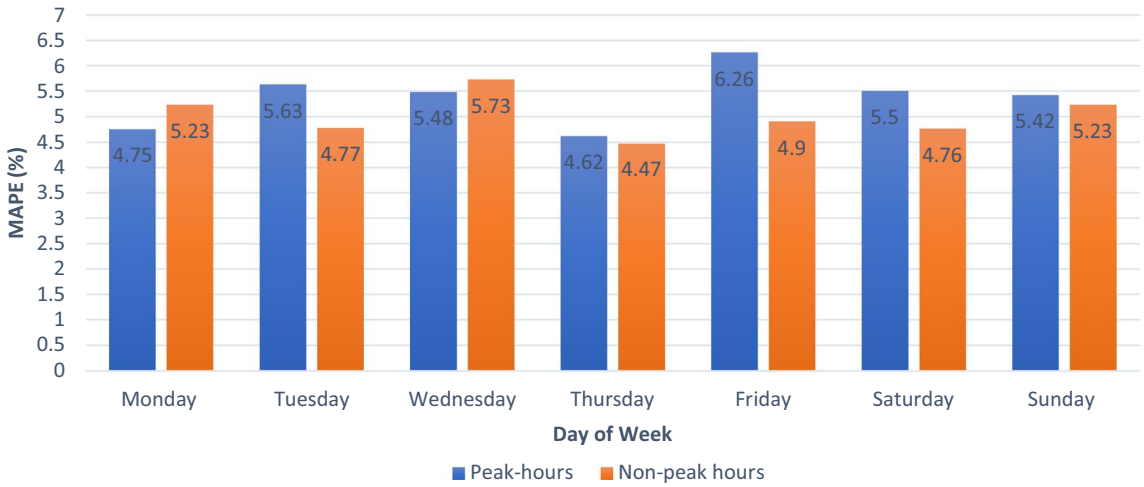


Fig. 15 MAPE at days of week during peak and non-peak hours

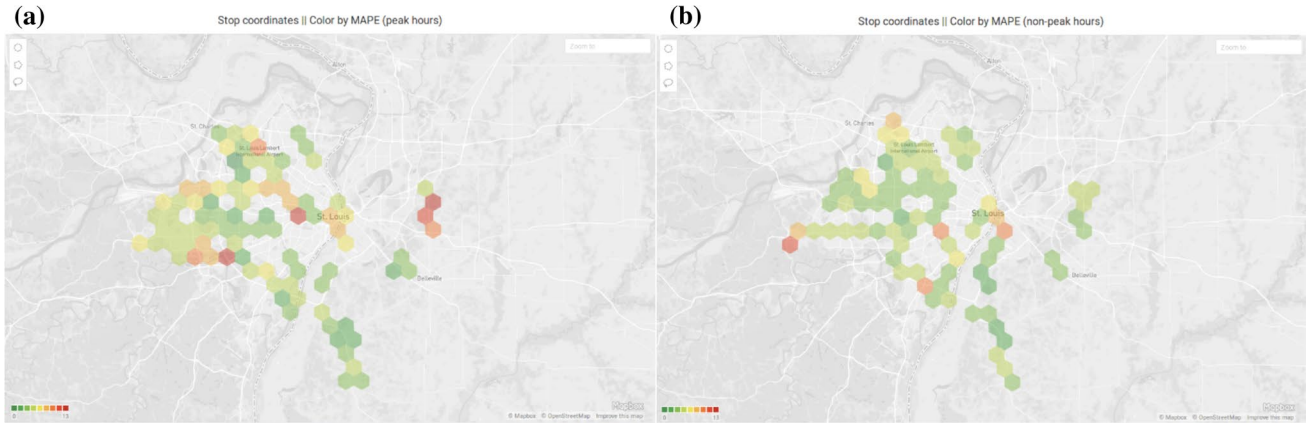


Fig. 16 Geo-heatmap at network level: **a** peak hours, **b** non-peak hours

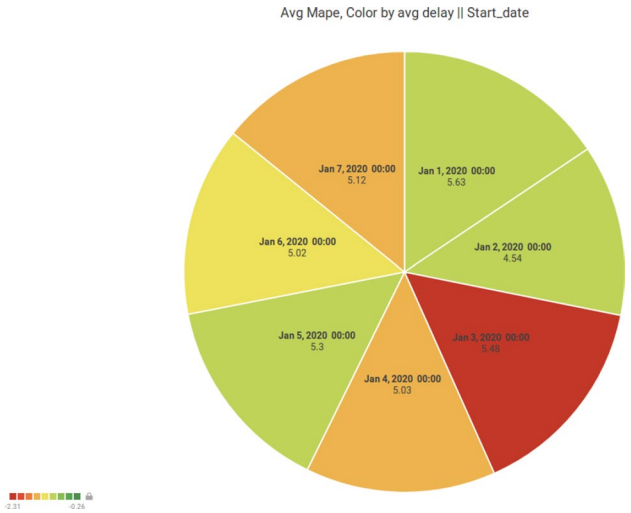


Fig. 17 MAPE pie chart of tested days

Conclusion

Although a wide range of models have been developed using bus locations in predicting delays, most of them were limited by data for one or few routes only because of the limited accuracy in using larger datasets with heterogenous traffic conditions. In aiming toward an intelligent public transport system (IPTS), accurate prediction of transit bus delays is essential. The current paper outlines the development of a network-level model that can be implemented on a large scale to simultaneously predict delays on multiple routes, buses, and stations. It leveraged the large heterogenous bus and traffic data in a deep machine learning architecture that utilizes entity embeddings to capture the influence of underlying variables such as bus schedule, traffic congestion and road segment characteristics. The variables datasets were fused together using spatio-temporal conflation to map them into a unified data layer.

The developed model was able to learn rich information from tabular datasets with continuous and categorical attributes, and a few interesting conclusions can be extracted from the tested dataset. First, it is reasonable to say that the use of real-time traffic speed seemed to have a good overall performance, predicting delays for multiple stops at a mean absolute percentage error (MAPE) of about 6%. The model was robust and high performing since the prediction errors were stable across days of week, bus stops, holidays, and time of day. Secondly, predictions at peak hour factors and the distance to bus stops influenced the model's prediction errors for some routes; however, the observed differences were not significant. The results, compared to previous studies, highlight the ability to simultaneously model continuous and categorical data with deep learning and the use of heterogenous data can attribute to such high performance on multiple routes.

A key limitation to this study is that it was designed for Saint Louis City, so its robustness when expanded to other cities is yet to be tested. The demographics of a different city can differentiate the variables used in prediction models. Further studies to test transferability of the model shall be explored. Additional performance measures can be tested next to delay such as headways, trip durations and cycle times. Further developments can also be done to this model by including additional factors that can influence predictions such as weather and passenger count.

Compliance with Ethical Standards

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Altinkaya M, Zontul M (2013) Urban bus arrival time prediction: a review of computational models. Int J Recent Technol Eng 2:164–169

Arik S, Pfister T (2019). TabNet: Attentive Interpretable Tabular Learning. http://arxiv.org/abs/1908.07442

Bai C, Peng Z, Lu Q, Sun J (2015) Dynamic bus travel time prediction models on road with multiple bus routes. Comput Intell Neurosci. https://doi.org/10.1155/2015/432389

Balasubramanian P, Rao K (2015) An adaptive long-term bus arrival time prediction model with cyclic variations. J Public Transp 18(1):1–18. https://doi.org/10.5038/2375-0901.18.1.6

Brakewood C, Macfarlane G, Watkins K (2015) The impact of real-time information on bus ridership in New York city. Transp Res Part C Emerg Technol 53:59–75. https://doi.org/10.1016/j.trc.2015.01.021

Cats O, Loutos G (2016) Evaluating the added-value of online bus arrival prediction schemes. Transp Res A Policy Pract 86:35–55. https://doi.org/10.1016/j.tra.2016.02.004

Celan M, Lep M (2018) Bus-arrival time prediction using bus network data model and time periods. Future Gener Comput Syst 110:364–371. https://doi.org/10.1016/j.future.2018.04.077

Chen M, Liu X, Xia J, Chien S (2004) A dynamic bus-arrival time prediction model based on APC data. Comput Aided Civ Infrastruct Eng 19:364–376. https://doi.org/10.1111/j.1467-8667.2004.00363.x

Chien S, Ding Y, Wei C (2002) Dynamic bus arrival time prediction with artificial neural networks. J Transp Eng 128(5):429–438. https://doi.org/10.1061/(ASCE)0733-947X(2002)128:5(429)

Deng L, He Z, Zhong R (2013) The bus travel time prediction based on Bayesian networks. IEEE: International Conference on Information Technology and Applications. http://doi.org/https://doi.org/10.1109/ITA.2013.73

Duan Y, Lv Y, Wang F (2016) Travel time prediction with LSTM neural network. IEEE 19th International conference on intelligent transportation systems, Rio de janeiro, Brazil

Dziekan K (2008) Ease-of-use in public transportation—a user perspective on information and orientation aspects. Stockholm: Department of Transport and Economics, Royal Institute of Technology

FHWA (2020) US Department of Transportation—Federal highway administration: traffic volume trends (January 2020). https://www.fhwa.dot.gov/policyinformation/travel_monitoring/20jantvt/20jantvt.pdf. Accessed 25 Jun 2020

Griffin G, Mulhall M, Simek C, Riggs W (2020) Mitigating bias in big data for transportation. J Big Data Anal Transp 2:49–59. https://doi.org/10.1007/s42421-020-00013-0

Huang W, Song G, Hong H, Xie K (2014) Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEE Trans Intell Transp Syst 15:2191–2201

Jia Y, Wu J, Du Y (2016) Traffic speed prediction using deep learning method. IEEE 19th conference on intelligent transportation systems, 1217–1222

Kumar B, Vanajakshi L, Subramanian S (2014) Pattern-based bus travel time prediction under heterogeneous traffic conditions. https://doi.org/https://doi.org/10.13140/RG.2.1.2338.5448

Li C, Wang J, Ye X (2018) Using a recurrent neural network and restricted Boltzmann machines for malicious traffic detection. Neuroquantology 16:823–831

Liu Q, Wang B, Zhu Y (2018) Short-term traffic speed forecasting based on attention convolutional neural network for arterials. Comput Civ Infrastruct 33:999–1016

Lv Y, Duan Y, Kang W, Li Z, Wang F (2015) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16:865–873

Ma J, Chan J, Ristanoski G, Ragasegarar S, Leckie C (2019) Bus travel time prediction with real-time traffic information. Transp Res Part C Emerg Technol 105:536–549. https://doi.org/10.1016/j.trc.2019.06.008

STL Metro 2019. Retrieved from: https://www.metrostlouis.org/

Mrgole A, Sever D (2016) Incorporation of duffing oscillator and Wigner-Ville distribution in traffic flow prediction. Sci Traffic Transp 29(1):13. https://doi.org/10.7307/ptt.v29i1.2116

One STL (2020) https://www.onestl.org/indicators/connected/metric/transit-ridership. Accessed 25 Jun 2020

Petersen N, Rodrigues F, Pereira F (2019) Multi-output deep learning for bus arrival time predictions. Transp Res Proc 41:138–145. https://doi.org/10.1016/j.trpro.2019.09.025

Rahman M, Wirasinghe S, Kattan L (2018) Analysis of bus travel time distributions for varying horizons and real-time applications. Transp Res Part C Emerg Technol 86:453–466. https://doi.org/10.1016/j.trc.2017.11.023

Ramakrishna Y, Ramakrishna P, Lakshmanan V, Sivanandan R (2008) Use of GPS probe data and passenger data for prediction of bus transit travel time. Transp Land Use Plan Air Qual. https://doi.org/10.1061/40960(320)13

Tsoi A, Back A (1997) Discrete time recurrent neural network architectures: a unifying review. Neurocomputing 15:183–223

Xu H, Ying J (2017) Bus arrival time prediction with real-time and historic data. Clust Comput 20:3099–3106. https://doi.org/10.1007/s10586-017-1006-1

Yang M, Chen C, Wang L, Yan X, Zhou L (2015) Bus arrival time prediction using support vector machine with genetic algorithm. Comput Sci 26:205–217. https://doi.org/10.14311/nnw.2016.26.011

Yi H, Jung H, Bae S (2017) Deep neural networks for traffic flow prediction. IEEE international conference on big data and smart computing, 328–331

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

Zhang C, Teng J (2013) Bus dwell time estimation and prediction: a study case in Shanghai–China. Proc Soc Behav Sci 96:1329–1340. https://doi.org/10.1016/j.sbspro.2013.08.151

Zhenliang M, Koutsopoulos H, Ferreira L, Mesbah M (2017) Estimation of trip travel time distribution using a generalized Markov chain approach. Transp Res Part C Emerg Technol 74:1–21. https://doi.org/10.1016/j.trc.2016.11.008

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. https://doi.org/10.1016/j.trc.2016.12.001