Real-time passenger train delay prediction using machine learning

A case study with Amtrak passenger train routes

Real-Time Passenger Train Delay Prediction using Machine Learning: A Case Study with Amtrak Passenger Train Routes

Pipatphon Lapamonpinyo*[a], Sybil Derrible [a], Francesco Corman [b]

[a] Department of Civil, Materials, and Environmental Engineering, University of Illinois at Chicago, Chicago, USA; [b] Department of Civil, Environmental, and Geomatic Engineering, ETH Zürich, Zurich, Switzerland

*Corresponding author: plapam2@uic.edu

Abstract—Passenger train delay significantly influences riders' decision to choose rail transport as their mode choice. This article proposes real-time passenger train delay prediction (PTDP) models using the following machine learning techniques: random forest (RF), gradient boosting machine (GBM), and multi-layer perceptron (MLP). In this article, the impact on the PTPD models using Real-time based Data-frame Structure (RT-DFS) and Real-time with Historical based Data-frame Structure (RWH-DFS) is investigated. The results show that PTDP models using MLP with RWH-DFS outperformed all other models. The influence of the external variables such as historical delay profiles at the destination (HDPD), ridership, population, day of the week, geography, and weather information on the real-time PTPD models are also further analyzed and discussed.

Keywords: big data, data science, gradient-boosting, machine learning, multi-layer perceptron, neural network, random forest, train delay prediction

1. INTRODUCTION

TRANSPORT systems are critical pieces of infrastructure and they have substantially increased in size in many countries worldwide [1]. This includes rail transport systems that have evolved significantly, including to provide long-distance travel services. In Sweden, the total distance travelled by trains increased by 8% between 2013 and 2016 [2]. In the United States (US), ridership on state-supported routes increased by more than 10%, making it the fastest growing segment of Amtrak's services [3]. On long-distance routes, both ridership and revenue increased in fiscal year 2018 by 6.2% and 7.3%, respectively. To sustain its competitiveness and attract more riders, ensuring a high on-time performance is critical [4]–[6]. Poor on-time performance can impact passenger trust and their satisfaction, and it may result in a shift to other modes of transport, especially private vehicles and air transport [5].

Service disruption is a root cause of lower rail punctuality and customer satisfaction. Major service disruptions result from various conditions or factors such as accidents, problems in train operation, malfunctioning or damaged equipment, routine maintenance, construction, passenger boarding or alighting, and even extreme weather conditions [4], [6], [7]. Rail service disruptions directly affect scheduled timetable and inevitably cause train delay [8]. Significant train delay can eventually lead to service loss or even cancellation [7]. In addition, train delay can also negatively affect connecting trains and passengers' journeys or activities [6]. Thus, delay estimations or predictions can help train operators develop better plans to manage, reschedule, or adjust the timetable of the current and consecutive trains more effectively, as well as to inform passengers in advance so they themselves can adjust their travel plans in time. Using or referring to historical average delay is insufficient to estimate future train delay as passenger train can potentially be affected by different factors such as ridership, accumulated delay from prior trains, or weather conditions.

In light of these problems, the main objective of this article is to model real-time passenger train delay prediction (PTDP) based on three ML techniques: random forest (RF), gradient boosting machine (GBM), and multi-layer perceptron (MLP) for Amtrak by using different types of endogenous and exogenous data from several sources from 2008 to 2019. Linear Regression (LR) is also applied as a benchmark. This article offers at least two contributions. First, it proposes and compares two data input structures: (a) a real-time based data-frame structure (RT-DFS) and (b) a real-time with historical based data structure (RWH-DFS). Second, it comprehensively evaluates the significance of several external variables, including historical delay profiles at the destination (HDPD), ridership, population, day of the

week, infrastructure and geography, as well as weather information. To ensure that the models can be applied to real-world systems, three different Amtrak passenger train routes with different delay profiles and rail-host performances are used as a case study. They are R364, R370, and R350 servicing Chicago to Port Huron, Chicago to Grand Rapids, and Chicago to Pontiac. All three routes provide service from or to Chicago Union station which is the Amtrak largest rail hub in the Midwest.

The rest of this article is organized as follows. Section 2 reviews the literature on train delay prediction modelling. Section 3 explains how the data was collected and prepared, and it properly defines how passenger train delay (PTD) is defined. Section 4 briefly introduces the technical background of LR, RF, GBM, and MLP that are used in this work. Section 5 explains how the data-frames were structured and how the models are evaluated. Section 6 presents the results and a discussion of the results. Finally, the conclusion summarizes the main findings of this work and offers a brief discussion of future research.

2. LITERATURE REVIEWS

Passenger train delay prediction (PTDP) has been made and modelled in several ways using a variety of approaches and techniques. Milinković et al. [9] proposed a fuzzy Petri net (FPN) model to estimate train delay of the Belgrade rail service (the train primary delays were simulated by a fuzzy Petri net module). Schlake et al. [10] used dispatch simulation software to simulate traffic volume and estimate train delays on single and double track rail lines. Ren Wang et al. [11] indicated that although simulation methods can be used to estimate complex train operations, they require tremendous effort to configure parameters such as dispatching rules as well as calibrating the models for complex train systems.

Thanks to technological advances in sensing, communication, and computing, real-time train position data has become available online. Besides, many train operators have also been recording their train delay performance for a long time. Thanks to this abundance of real-time and historical data, several types of regression and machine learning (ML) approaches have been applied to construct train delay prediction models. Ren Wang et al. [11] proposed a historical regression model designed to predict train delays before the current trip starts and an online regression model aimed at providing a more accurate train delay estimation after the trip begins by using the delay recorded at the upstream station on the current trip and the delay recorded by other nearby train. Oneto et al. [12] proposed a train delay prediction model based on Extreme Learning Machine (ELM), Kernel Regularized Least Squares (KRLS), and Random Forest (RF) by using historical train data from Rete Ferroviaria Italiana (RFI) and exogenous weather data from the Italian national weather services for Italian railway network. The study concluded that the RF based model consistently outperformed the others.

In 2017, Gal et al. [13] applied AdaBoost (AB) and GBM together with snapshot method from Queueing Theory to improve ensembles of regression trees for predicting travel time based on historical data of scheduled bus journals in Dublin, Ireland. Gal et al. [13] proved that combining snapshot rule with GBM leads to a more robust and better predictions regardless of the increase in trip length. In the same year, Estes et al. [14] applied GBM and RF to solve delay prediction problem in air transport, and their work demonstrated that GBM outperformed RF. Afterwards, Oneto et al. [7] proposed a new optimization algorithm using Stochastic Gradient Descent (SGD) and built the Train Delay Prediction System (TDPS) model using Shallow and Deep Extreme Learning Machines (SELM and DELM) techniques for large-scale Italian railway network. Six-months of Italian train historical data from RFI was also used in the TDPS model in this work, but without weather information. In 2018, Gaurav and Srivastava [15] also applied RF and Ridge Regression (RR) to estimate train delays, and their results showed that RF outperformed RR. Nair et al. [16] also applied RF and Kernel regression to developed passenger train delay models for Deutsche Bahn passenger rail network in Germany. Nair et al. [16] used both historical delays and weather data to develop the prediction models. Other static features such as holiday, infrastructure, and network related data are also included into the models. The results showed that overall RF performed better than others, and accuracy depends on several factors such as service type and current delay of the operational trains. Taleongpong et al. [17] proposed a delay prediction system with the application of extreme gradient boosting and neural network for Great Western Railway journeys in UK by utilizing data from Darwin (UK railway delay prediction system) in 2016-2017. The study pointed out that extreme gradient boosting, and neural network models outperformed decision tree, linear regression, and Darwin's delay prediction models.

In previous studies, regression-based models and ML—especially ensemble methods and neural networks—play more important roles in train delay prediction (TDP) for large-scale rail network worldwide. However, to the authors' knowledge, ML has never been applied to model PTDP for Amtrak, but linear regression has [11]. Train historical data is also essential as one of the key features or inputs to develop TDP models. Besides, external information such as weather information is also believed to be related to variation of train delay. However, only a few historical and external weather-related features have been examined in previous works. Moreover, a comprehensive examination and influence analysis of those features on real-time PTDP models have not been thoroughly studied yet, especially for national railways like Amtrak where each route has different delay profiles.

3. DATA

In this work, several endogenous and exogenous types of data are considered and examined, including historical train delay and weather information from 2008 to 2019.

3.1. Endogenous Data

The proposed real-time PTDP models are mainly formed by using departure and/or arrival time data from the Amtrak Status Maps Archive Database (ASMAD) from 2008 to 2019. In this work, passenger train delay is directly extracted from the difference between scheduled and actual departure and/or arrival times. More details on extracting passenger train delay from ASMAD data are given in section 3.4.

A case study of Amtrak passenger train routes R364, R370, and R350 are used to build, train, and evaluate the proposed real-time PTDP models. Figure 1 shows the details of Amtrak passenger train route R364 (in blue) providing service from Chicago to Port Huron that covers 11 stations over a total distance of about 513 kilometers (319 miles). Amtrak passenger train route R350 (in red) provides service from Chicago to Grand Rapids (MI), covering 5 stations over a total distance of about 283 kilometers (176 miles). Amtrak passenger train route R350 (in green) provides service from Chicago to Pontiac (MI), covering 13 stations over a total distance of about 489 kilometers (304 miles).

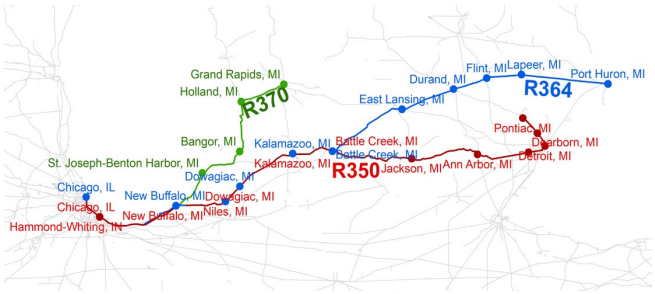


Fig 1. Amtrak routes of passenger trains R364, R370, and R350

These three routes depart from or arrive in Chicago which is one of the busiest stations and the biggest hub in the Midwest connecting passengers to major cities throughout the US. Furthermore, these three routes were selected as they have different delay profiles as shown in Figure 2, which enables us to develop a model that performs well regardless of delay profile. Finally, they also have different rail-host performance grades enabling us to develop a model that performs well regardless of rail-host performance grade as well. Rail-host performance grade is a measurement in term of minutes per 10,000 train-miles that Amtrak uses to evaluate how much delay each private freight company, known as railroad "host," causes to Amtrak passenger trains. This is because most Amtrak passenger trains in the US are operated via rail networks or railroads owned, maintained, and dispatched by freight companies [18]. The railroad hosts of Amtrak route R364, R370, and R350 are Norfolk Southern Railway (NS), CSX Transportation (CSX), and Canadian National Railway (CN), respectively. The four-year average rail-host performances of these routes evaluated by Amtrak in 2020 are more than 1,500 (graded F), in between 900-1,200 (graded B), and in between 1350-1500 (graded D) minutes per 10,000 train-miles, respectively [19].

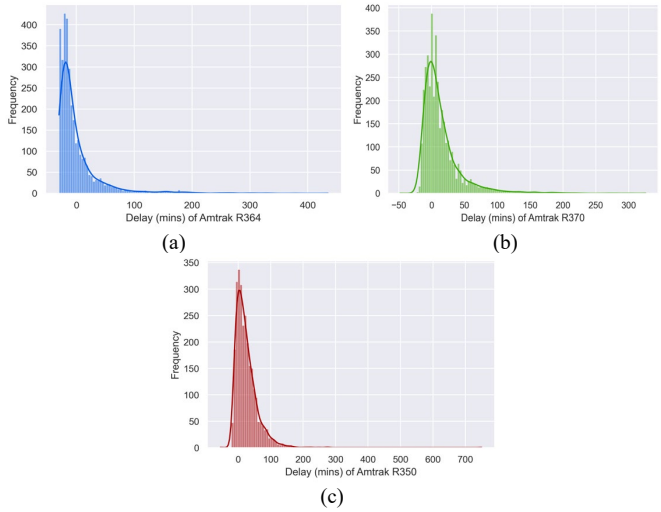


Fig 2. Delay profiles of Amtrak passenger train routes (a) R364, (b) R370, and (c) R350

3.2. Exogenous Data

In addition to departure and arrival time data from ASMAD, other external variables such as ridership, population, infrastructure, and geography, as well as weather information are hypothesized to have some influences on passenger train delay. For example, a higher number of passengers at a station is expected to increase the dwell time at the station to let riders board and alight. Severe weather conditions such as heavy rain and snow may affect train operation or change the behavior of passengers or train operations, possibly affecting train delay. The day of the week a train is running is also expected to influence passenger train delay. Freight trains may be more active on a particular weekday or passenger behavior may differ on weekends. Finally, the total distance or distance between stations is also included since a longer distance may result in a higher probability of malfunction, which would increase passenger train delay.

The summary of all endogenous and exogenous variables considered and examined in this work are shown in table 1. More details are given in the section 3.3 on retrieving and processing raw endogenous and exogenous data from various sources that are stored in different data formats.

TABLE 1
SUMMARY OF ENDOGENOUS (1-2) AND EXOGENOUS (3-6) FEATURES USED AND EXAMINED FOR BUILDING REAL-TIME PTDP MODELS

No	Data Type	Feature	Description
1	Scheduled and actual timetable related	Scheduled departure or arrival times	Retrieved from ASMAD
		Actual departure or arrival times	
2	Historical delay profile at destination (HDPD) related	Historical delay at the destination	Derived from No.1
		Average historical delay at the destination (duration ranging from 2, 3, 5, 7, 14, 21, 30 days)	
		Median historical delay at the destination (duration ranging from 2, 3, 5, 7, 14, 21, 30 days)	
3	Passenger related	Ridership	Ridership data at each station are provided by

No	Data Type	Feature	Description
			Amtrak through Michigan Department of Transportation (MDOT)
		Population	Annual population data at the county level is retrieved from the American Community Survey (ACS)
4	Day of the week	Weekdays (0-6)	Calendar related data in term of weekday or weekend indicators is derived from Amtrak departure or arrival dates of each train
		Weekday indicator (0 or 1)	
		Weekend indicator (0 or 1)	
5	Infrastructure and Geographic related	Distance between stations	Distances between stations or for the entire route are extracted from Federal Railroad Administration's (FRA) Amtrak Station database by using ArcMap
		Accumulated distance from the origin	
		Receded distance from the destination	
		Total distance	
6	Weather information related	Rain indicator (1 if it rains, otherwise, 0)	Weather information from the closest weather station to each Amtrak station and correspond to the scheduled departure or arrival time from 2008 to 2019 is retrieved from Weather Underground
		Snow indicator (1 if it snows, otherwise, 0)	
		Fog Indicator (1 if there is fog, otherwise, 0)	
		Precipitation hourly	
		Snow hourly	
		Temperature	
		Feels-like temperature	
		Heat index	
		Wind chill (WC)	
		Pressure Tend	
		Dew Point	
		Relative Humidity (RH)	
		Visibility	
		Wind Direction	
		Wind Gust Speed (Gust)	
		Wind Speed (WSPD)	
		UV Index	

3.3. Data Retrieval and Processing

In this article, a database first needs to be built that mainly combines Amtrak and Weather Underground raw data. The original departure and arrival time data from ASMAD is in the form of HTML tables provided by its PHP scripts, whereas the raw data from Weather Underground provided by its API service is in the form of JSON objects with nested dictionary structure. Because the data structure and format from the two sources are different, they initially need to be manipulated to be suitable for building, training, and testing the train delay prediction models. Therefore, in this work, we apply the Python-based Amtrak and Weather Underground (PAWU) data collecting tool provided by Lapamonpinyo, Derrible, and Corman [20] to retrieve, convert, and simplify raw data from the two sources to be the same format and store it on MySQL database. The data retrieved from ASMAD and Weather Underground stored in MySQL database is queried to construct a data-frame for building, training, and testing the train delay prediction models via the Python-based Pandas data analysis and manipulation tool.

Ridership and population raw data provided by the Michigan Department of Transportation (MDOT), and ACS can be retrieved to local memory as comma-separated values (CSV) files from the data providers websites to be later converted into data-frames. Afterward, the ridership and population data-frames need to be mapped with the corresponding date and time of Amtrak departure or arrival time data-frame for each train at a given station including weather information data-frame if considered to build real-time PTDP models with or without external variables.

More detail discussion of a data-frame and model construction is provided in the Methods session.

3.4. Definition of Passenger Train Delay (PTD)

The difference between the scheduled and actual departure or arrival times is defined as *delay*. In other words, PTD can be calculated from the difference between the scheduled and actual departure or arrival times, defined as follow:

$$\text{PTD} = \Delta(SDT, ADT) \mid \Delta(SAT, AAT) \quad (1)$$

where SDT is the schedule departure time, ADT is the actual departure time, SAT is the scheduled arrival time, and AAT is the actual arrival time. The symbol ' | ' is the 'or' operator. It means that PTD can be derived from the difference between the two departure times ($\Delta(SDT, ADT)$) denoted as (PTDD) or the two arrival times ($\Delta(SAT, AAT)$) denoted as (PTDA). For example, train R364 is scheduled to arrive Port Huron station at 23:00. If the train arrives at the station at 23:10, it is considered that the train arrived at the station late and its arrival delay (PTDA) is 10 minutes.

In this article, we develop models to predict PTDA.

4. METHODS

We model real-time passenger train delay prediction by using linear regression (LR) as a baseline model for comparison, two common ensemble tree-based ML techniques: Random Forest (RF) and Gradient Boosting Machine (GBM), and Multi-layer Perceptron (MLP).

4.1. Linear Regression (LR)

LR is an approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables) [21]. A linear regression line has an equation of the form $Y = a + bX$, where X are the independent variables and Y is the dependent variable. The slope of the line is b and a is the intercept (the value of y when $x = 0$) [22]. A multiple linear regression model attempts to model the relationship between two or more explanatory variables as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i, i = 1,2, \ldots n \quad (2)$$

where i is number of observations, y_i is the dependent variable, x_i are the explanatory variables, β_0 is the intercept, β_p are the slope coefficients for each explanatory variable, and ε is the error term [23].

4.2. Random Forest (RF)

RF is a combination of multiple decision tree predictors, each of which depends on the values of a random vector sampled independently and with the same distribution for all decision trees in the collection of classifiers (or the so-called forests) [24], [25]. As an ensemble technique, RFs adopts a bagging approach, as in several imperfect models are trained and the result is the average of all models. The general algorithmic approach with random attribute selection is as follows [26]:

1) Repeat the following steps for iteration *i = 1, 2, ..., B*:
   a) Draw a bootstrap sample (Z^*) of size *N* from the training data set.
   b) Grow a random forest of tree (T_i) to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size (n_{min}) is reached.
      (i) Randomly select *m* variables/attributes from all *p* variables in the data set.
      (ii) Pick the best variable/split-point among the *m*.
      (iii) Split the node into two daughter nodes.
2) Output the ensemble of trees $\{T_i\}_1^B$.
   Thus, a prediction at a new point *x* is defined as:

$$\hat{f}_{RF}^{B}(x) = \frac{1}{B}\sum_{i=1}^{B} T_i(x) \quad (3)$$

   where *B* is the number of trees which is a free parameter depending on the size and characteristics or nature of the training data.

4.3. Gradient Boosting Machine (GBM)

GBM is also an ensemble approach like RF, but instead of using random attribute selection (known as bagging), it adopts a boosting approach to build decision trees. The concept of GBM begins by training a decision tree for which each observation is assigned an equal weight [27]. After the first evaluation, the weights of the observations, which are difficult to classify, are increased, whereas the weights for those that are easy to classify are decreased; in practice, residuals are used as weights since higher residuals translate to higher weights. The second tree is grown using this weighting concept, and therefore the new model is the combination of the first and second trees. After that, the error computed from the new ensemble tree will be used to grow a third tree to predict the revised residuals. These steps are then repeated over a given number of iterations. As a result, the final ensemble model is the weighted sum of the previous trees trained, evaluated, and grown using the steps mentioned above. The generic algorithm used in gradient tree-boosting for regression can be summarized as follows [28]:

1) Initialize the optimal constant model/approximation ($f_0(x)$) or a single terminal node tree denoted as:

$$f_0(x) = arg\ min_\gamma \sum_{i=1}^{N} L(y_i, \gamma) \quad (4)$$

   where γ is a pseudo-residual, *L* is a loss function on training data, y_i is the observed value, and *i* is index over training/sampling set of size *N*.
2) Update model/approximation by repeating the following steps for each iteration *m* = 1 to *M*:
   a) Compute pseudo-residuals (γ)
   b) Fit a regression tree (a weak base learner) to the targets.
   c) Minimize the loss function
   d) Update the model/approximation $f(x)$
3) Output $\hat{f}(x)$ is given as best approximation

4.4. Multi-layer Perceptron (MLP)

A neural network also called an artificial neural network (ANN) contains layers of interconnected nodes called neurons [29]. A basic ANN consists of three main components: input layer, hidden layer, and output layer. A more complex ANN is comprised of multiple hidden layers, which is commonly called multi-layer perceptron (MLP) [30].

MLP is one of the most popular feed forward classes of ANNs in real world applications [31]. In MLPs, the data flows in the forward direction from input to output layer. The input layer receives the input signal or raw data to be processed. The hidden layers performed computational tasks, while the output layer provides a prediction or classification [32]. Neurons in MLP are trained with a back propagation learning algorithm—another term of MLP is Back Propagation Neural Network (BPNN). The MLP process is summarized as follows.

1) Initialization: An input vector is fed into the input nodes or neurons in the hidden layer(s), and all weights are initialized to small random values (positive and negative).
2) Training: Train the model by repeating the following steps until the convergence is achieved:
   2.1) Forwards propagation.
      a) Compute the activation of each neuron in the hidden layer(s).
      b) Work through the network until the output layer neurons are determined.
   2.2) Backwards propagation.
      c) Compute the error at the output as the sum-of-squared difference between the network outputs and the targets.
      d) Compute the error in the hidden layers.
      e) Update the output layer weights.
      f) Update the hidden layer weights.
3) Prediction: Recall steps 2.1)

5. Modelling and Performance Evaluations

5.1. Modelling Real-time Passenger Train Delay Prediction (PTDP)

The main objective of the models is to predict passenger train delay (PTD) in minutes when the train arrives at a destination (D_{Si}^*) denoted as PTDA in section 3.4. This is a typical regression problem.

Firstly, a data-frame containing independent variables (D_{si}) needs to be constructed as an input for each PTDP model. Next, the constructed data-frame will be fed into a given prediction model developed by using different

approaches, namely LR, RF, GBM, and MLP. The PTDP model will then process the given data-frame and predict PTD at the destination in minutes as an output which is typically known as a dependent variable (D^*_{Si}).

5.2. Data-Frame Structures

This work proposes two different data-frame structures for building real-time PTDP models. The first data-frame structure consists of only real-time PTD data with or without external variables. The second data-frame structure is comprised of both real-time and historical PTD data with or without external variables. The historical data added up into the second data-frame structure is expected to help improve the performance of the proposed PTDP models. The concepts of building data-frame structures in this work are presented in general forms and with examples to illustrate how they can be applied. The performance comparison of these two proposed data-frame structures will be also provided in the result and discussion session.

5.3. Real-time based Data-frame Structure (RT-DFS)

RT-DFS prepares data input for PTDP models to predict delay at destination (D^*_{SP}) by using only known real-time delay (D^r_{Si}) of the current train (T_C) from the origin station (S_O) to the current stations (S_C) where the train is located as shown in figure 3.

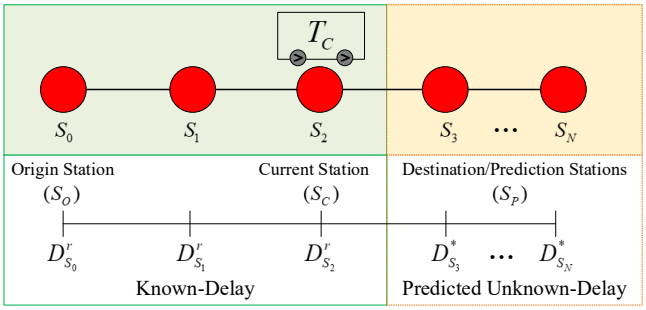


Fig 3. An illustration of variable definitions for the proposed PTDP models.

To estimate the PTDA throughout the entire route of $N\times(N-1)/2$ models are built where N is the number of stations of a given train route. For example, to predict PTDA of Amtrak train R370 at each station throughout its entire route from Chicago to Grand Rapids (MI), 10 prediction models for each of the four techniques, namely LR, RF, GMB, and MLP used are built, resulting in 40 models without external variables.

As for PTDP models without external variables, all known train delay of all stations from the origin station (S_O) to the current station (S_C) are used as independent variables ($D^r_{S_i}\ for\ i = 0,1,2,\dots C$), whereas the train delay at the destination/prediction station ($D^*_{SP}\ where\ C < P < N$) is the dependent variable. For example, table 2 (a) shows the proposed RT-DFS for predicting PTD of Amtrak route R370 without considering external variables.

As for the PTDP models with external variables, all exogenous features at each station as shown in Table 1 (2-6) (e.g., ridership, population, or weather information) are included into the proposed PTDP models so that the influence of each external variable on the models can be evaluated. Thus, the independent variables of the proposed PTDP with external variables does not only consist of known real-time delay at each station ($D^r_{si}\ for\ i = 0,1,2,\dots C$) but also external variables ($E^r_{si}\ for\ i = 0,1,2,\dots C$). Table 2 (b) shows an example of the proposed *RT-DFS* with external variables for predicting PTD of Amtrak route R370.

TABLE 1
AN EXAMPLE OF THE PROPOSED REAL-TIME BASED DATA-FRAME STRUCTURE (RT-DFS) (A) WITH AND (B) WITHOUT EXTERNAL VARIABLES FOR PREDICTING PTD OF AMTRAK TRAIN R370.

No.	S_C	S_P	Independent Variables			Dependent Variable
			(a) Without external variables	(b) With external variables		
			Real-time Delay ($D^r_{S_i}$)	*Real-time Delay* ($D^r_{S_i}$)	Exogenous variables ($E^r_{S_i}$)	Predicted Delay ($D^*_{S_P}$)
0	S_0	S_4	$D^r_{S_0}$	$D^r_{S_0}$	$E^r_{S_0}$	$D^*_{S_4}$
1	S_1	S_4	$D^r_{S_0}, D^r_{S_1}$	$D^r_{S_0}, D^r_{S_1}$	$E^r_{S_0}, E^r_{S_1}$	$D^*_{S_4}$
2	S_2	S_4	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}$	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}$	$E^r_{S_0}, E^r_{S_1}, E^r_{S_2}$	$D^*_{S_4}$
3	S_3	S_4	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}, D^r_{S_3}$	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}, D^r_{S_3}$	$E^r_{S_0}, E^r_{S_1}, E^r_{S_2}, E^r_{S_3}$	$D^*_{S_4}$
4	S_0	S_3	$D^r_{S_0}$	$D^r_{S_0}$	$E^r_{S_0}$	$D^*_{S_3}$
5	S_1	S_3	$D^r_{S_0}, D^r_{S_1}$	$D^r_{S_0}, D^r_{S_1}$	$E^r_{S_0}, E^r_{S_1}$	$D^*_{S_3}$
6	S_2	S_3	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}$	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}$	$E^r_{S_0}, E^r_{S_1}, E^r_{S_2}$	$D^*_{S_3}$
7	S_0	S_2	$D^r_{S_0}$	$D^r_{S_0}$	$E^r_{S_0}$	$D^*_{S_2}$
8	S_1	S_2	$D^r_{S_0}, D^r_{S_1}$	$D^r_{S_0}, D^r_{S_1}$	$E^r_{S_0}, E^r_{S_1}$	$D^*_{S_2}$
9	S_0	S_1	$D^r_{S_0}$	$D^r_{S_0}$	$E^r_{S_0}$	$D^*_{S_1}$

5.4. Real-time with Historical based Data-frame Structure (RWH-DFS)

RWH-DFS prepares data input for PTDP models to predict delay at destination stations (D^*_{SP}) like RT-DFS by using not only known real-time delay ($D^r_{S_i}\ for\ i = 0,1,2,\dots C$) of the current train ($T_C$) from the origin station (S_O) to current station (S_C) where the train is located, but also by including all historical delay after the current station (S_{C+1}) to the station before the destination (S_{P-1}) from the previous time the train ran (T_{C-1}), denoted as $D^h_{S_{C+j}}\ for\ j = 1,2,3,\dots(P-C)\ where\ \forall(C+j) < P$.

Table 3 shows an example of a data-frame structure for predicting PTD of Amtrak route R370 that is built by using both real-time of the current train (T_C) and historical data from the previous time the train ran (T_{C-1}) with (a) or without (b) external variables. For example, model no. 1 in Table 3 (a), a given train is currently at station S1, and this model is to predict PTD at the destination S_4 ($D^*_{S_4}$). In this case, real-time train delays of T_C at S_0 and S_1 ($D^r_{S_0}$ and $D^r_{S_1}$) are known, thus the historical passenger train delay of T_{C-1} at S2 and S3 ($D^h_{S_2}$ and $D^h_{S_3}$) are used to build RWH-DFS based data input for model no.1. In other words, the independent variables of the PTDP model using RWH-DFS without external variables for model no.1 consists of both real-time train delay of T_C at station S0 and S1 ($D^r_{S_0}$ and $D^r_{S_1}$) and historical train delay of of T_{C-1} at stations S2 and S3 (D^h_{s2} and D^h_{s3}), respectively.

As for the PTDP model using RWH-DFS with external variables does not only consist of known real-time delays of T_C from S_0 to S_C and historical train delays of T_{C-1} from S_{C+1} to S_{P-1}, but also real-time and historical external variables as well. For instance, model no.1 of Amtrak route R370 as in Table 3 (b), the independent variables does not only include known real-time delay of T_C at stations S0 and S1 ($D^r_{S_0}$ and $D^r_{S_1}$) and historical train delays of T_{C-1} at station S2 and S3 ($D^h_{S_2}$ and $D^h_{S_3}$), but also real-time external variables at stations S0 and S1 ($E^r_{S_0}$ and $E^r_{S_1}$) and historical external variables at stations S2 and S3 ($E^h_{S_2}$ and $E^h_{S_3}$), respectively.

TABLE 2
AN EXAMPLE OF THE PROPOSED REAL-TIME WITH HISTORICAL BASED DATA-FRAME STRUCTURE (RWH-DFS) (A) WITH AND (B) WITHOUT EXTERNAL VARIABLES FOR PREDICTING PTD OF AMTRAK TRAIN R370.

			Independent Variables			Dependent Variable
			(a) Without external variables	(b) With external variables		
No.	S_C	S_P	*Real-time ($D^r_{S_i}$) with Historical (D^h_{Si}) Delay*	*Real-time ($D^r_{S_i}$) with Historical ($D^h_{S_i}$) Delay*	*Real-time ($E^r_{S_i}$) and Historical ($E^h_{S_i}$) External variables*	Predicted Delay ($D^*_{S_P}$)
0	S_0	S_4	$D^r_{S_0}, D^h_{S_1}, D^h_{S_2}, D^h_{S_3}$	$D^r_{S_0}, D^h_{S_1}, D^h_{S_2}, D^h_{S_3}$	$E^r_{S_0}, E^h_{S_1}, E^h_{S_2}, E^h_{S_3}$	$D^*_{S_4}$
1	S_1	S_4	$D^r_{S_0}, D^r_{S_1}, D^h_{S_2}, D^h_{S_3}$	$D^r_{S_0}, D^r_{S_1}, D^h_{S_2}, D^h_{S_3}$	$E^r_{S_0}, E^r_{S_1}, E^h_{S_2}, E^h_{S_3}$	$D^*_{S_4}$
2	S_2	S_4	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}, D^h_{S_3}$	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}, D^h_{S_3}$	$E^r_{S_0}, E^r_{S_1}, E^r_{S_2}, E^h_{S_3}$	$D^*_{S_4}$
3	S_3	S_4	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}, D^r_{S_3}$	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}, D^r_{S_3}$	$E^r_{S_0}, E^r_{S_1}, E^r_{S_2}, E^r_{S_3}$	$D^*_{S_4}$
4	S_0	S_3	$D^r_{S_0}, D^h_{S_1}, D^h_{S_2}$	$D^r_{S_0}, D^h_{S_1}, D^h_{S_2}$	$E^r_{S_0}, E^h_{S_1}, E^h_{S_2}$	$D^*_{S_3}$
5	S_1	S_3	$D^r_{S_0}, D^r_{S_1}, D^h_{S_2}$	$D^r_{S_0}, D^r_{S_1}, D^h_{S_2}$	$E^r_{S_0}, E^r_{S_1}, E^h_{S_2}$	$D^*_{S_3}$
6	S_2	S_3	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}$	$D^r_{S_0}, D^r_{S_1}, D^r_{S_2}$	$E^r_{S_0}, E^r_{S_1}, E^r_{S_2}$	$D^*_{S_3}$
7	S_0	S_2	$D^r_{S_0}, D^h_{S_1}$	$D^r_{S_0}, D^h_{S_1}$	$E^r_{S_0}, E^h_{S_1}$	$D^*_{S_2}$
8	S_1	S_2	$D^r_{S_0}, D^r_{S_1}$	$D^r_{S_0}, D^r_{S_1}$	$E^r_{S_0}, E^r_{S_1}$	$D^*_{S_2}$
9	S_0	S_1	$D^r_{S_0}$	$D^r_{S_0}$	$E^r_{S_0}$	$D^*_{S_1}$

5.5. Hyper-parameter Configurations

To maximize model performance, fine-tuning the hyperparameters is necessary. The details of essential hyperparameter configurations, value ranges tested, and the optimized values in this work for each proposed PTDP modes using RF, GBM, and MLP methods are provided in the subsections below.

5.5.1. Tuning RF-based PTDP Models

Different numbers of estimators (*n_estimator*) and maximum depts (*max_depth*) of decision trees in the forest significantly affect the PTDP model performance. These two parameters are interchangeable. Based on our experiment with different values of *n_estimator* ranged from 10 to 10,000 and number of *max_depth* ranged from 10 to 3000, the optimized *n_estimator* and *max_depth* in this work are 1000 and 300 respectively.

5.5.2. Tuning GBM-based PTDP Models

In addition to *n_estimator* and *max_depth* like RF-based models, different loss function (*loss*) and learning rates (*learning_rate*) yield different performance for PTDP models as well. The most efficient loss functions examined in this work are least squares regression (*ls*), least absolute deviation (*lad*), a combination of the squared-error loss function and absolute-error loss function (*huber*), and quantile regression (*quantile*). '*ls*' and '*lad*' are more efficient and less-time consuming than '*huber*' and '*quantile*' and the optimized loss function for the proposed PTDP models in this work is '*ls*'. Learning rates ranging from 0.1 to 0.9 are also examined and the optimized learning rate in this work is 0.2.

5.5.3. Tuning MLP-based PTDP Models

The optimized number of hidden layers (n^h) and the number of neurons in each *i* hidden layer (n^n_i) are essential for building high-performance PTDP models which provide the optimal and consistent performance in terms of R^2, MAE, MSE (defined below). Different numbers of hidden layers ranged from 1 to 32 besides an output layer and various fixed numbers of neurons in each hidden layer ranged from 1 to 256 and flexible numbers of neurons in each hidden layer varied by the number of features in the input layer (n^f) defined as the below equation are also examined.

$$n^n_i = \begin{cases} (n^f) * 2^{i-1}, & if \quad i \le \dfrac{n^h}{2} \\ (n^f) * 2^{(n^h - i)}, & otherwise \end{cases} \tag{5}$$

where $i = 1, 2, 3, \dots, n^h$

In this work, n^h equal to 3 is the optimum for the proposed PTDP with the flexible number of neurons in each *i* hidden layer (n^n_i) varied by the total number of features (n^f) at the input layer. For example, if the number of features or elements at the input layer is 10, the optimized number of neurons at the first (1st), second (2nd), and third (3rd) layer is 10, 20, and 10 respectively.

5.6. Model Training, Evaluations, and Interpretations

5.6.1. Model Training and Outcome Performance Evaluations

In this work, each PTDP model is trained on 80% of the data and tested on the remaining 20% of the data. The training and testing data sets are randomly selected to ensure that the model is not susceptible to variations or changes in data over time. The performance of each model is assessed by using the standard goodness-of-fit measure R^2, Mean Absolute Error (*MAE*), and Root Mean Squared Error (*RMSE*). R^2 is bounded between minus infinity (poor) to one (perfect performance). MAE is one of the most common model evaluation metrics often used with regression models. It is literally the mean of the absolute values of the individual prediction errors on over all instances in the test set [33]. Each prediction error is the difference between the true value and the predicted value for the instance. *RMSE* is the Root of the Mean of the Square of Errors (*MSE*). *MSE* of a model is the mean of the squared prediction errors over all instances in the test set. The prediction error is the

difference between the true value and the predicted value for an instance [33].

In this work, all PTDP models without external variables based on LR, RF, GBM are constructed first. After that, the best prediction approach yielding the highest performance in terms of R^2, *MAE*, and *RMSE* is selected and used to construct PTDP models with external variables so as to examine how each exogenous variables affect the performance of the prediction models based on the most effective or suitable ML method for this work.

5.6.2. Variables' Significance Evaluations

For the linear regression models, the importance of independent variables can be determined by t-statistic values. t-statistic values are calculated using the common Python tool STATSMODELS.

For the ML models, the importance of independent variables can be determined with feature importance (FI) introduced by [24]. The concept of FI is to assign a score to each predictor based on its ability to improve predictions. So that the importance of each predictor or variable in the prediction model can be evaluated and ranked based on its relative predictive power [34]. In this article, the FI values of ensemble-based models either RF or GBM are estimated after fitting the model by calling "feature_importances_" feature through Scikit-learn (Sklearn), a robust tool for ML and data analysis in Python. As for MLP, FI is estimated through the ELI5 Python library which provides a method named "PermutationImportance" to compute FI for any model by measuring how scores decrease when a feature is not available [24], [35].

6. Results and Discussion

6.1. Performances of real-time PTDP models based on the proposed RT-DFS without external variables

Fig. 4 (a-c) shows the average performances in terms of R^2 (a), *MAE* (b), and *RMSE* (c) of PTDP models built by using the proposed RT-DFS for all Amtrak routes R364, R370 and R350 without external variables. The x-axis in Fig. 4 refers to the number of stations between the current station where the train is located (S_C) and the destination station (S_P) where the unknow PTD is predicted.

The average R^2 of the proposed RT-DFS based PTDP models using RF and MLP are about 0.60 which outperformed the others using GBM and LR yielding R^2 of 0.55 and 0.47. Besides, Fig4 (a) indicates that the limitation of RT-DFS based models using MLP and RF to predict PTD in term of the number of stations between S_C and S_P is 11 stations, whereas the limitation of the model using GBM is 10 and that of LR is only 8 stations, respectively.

The average *MAE* of the proposed RT-DFS based models built using MLP are 9.9 minutes, which is better than the models built using GBM, RF, and LR having *MAE* of 10.6, 11.1, and 11.84 minutes, respectively. As for *RMSE*, the proposed PTDP models based on MLP provides better performance than RF, GBM, and LR up to 2 minutes.

Overall, the real-time RT-DFS based models using MLP technique excels the models using RF, GBM, and LR, respectively.

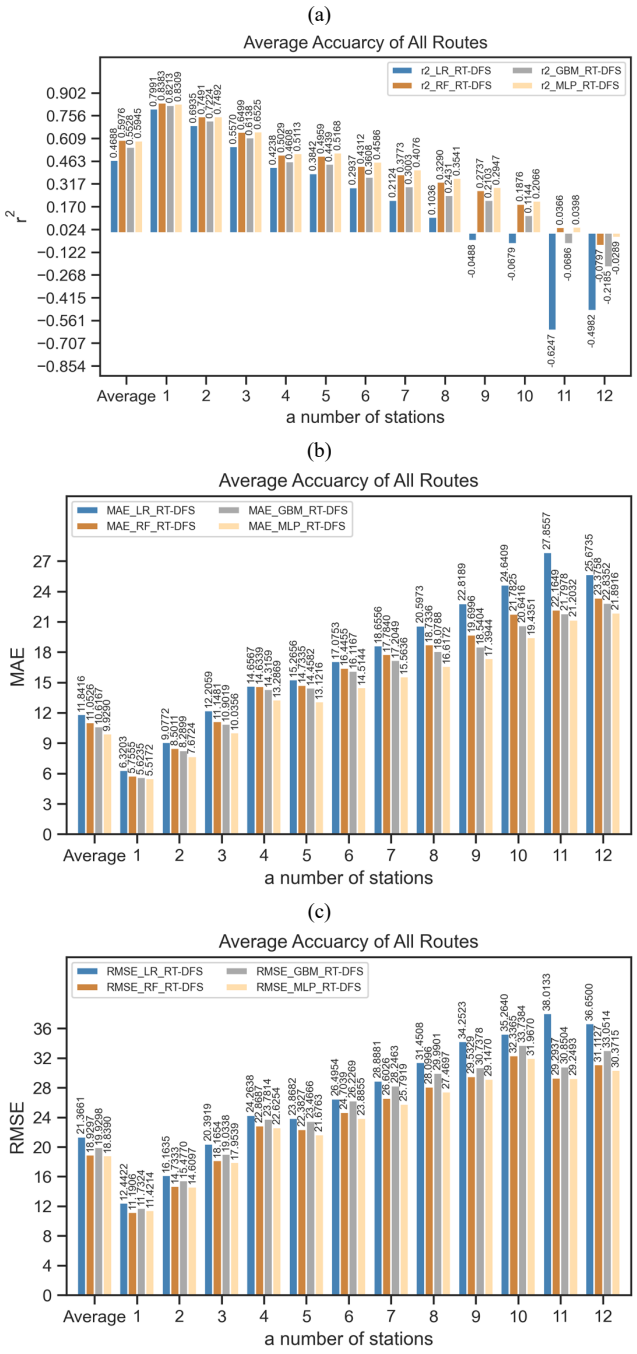


Fig 4. Average performance of the proposed real-time PTDP models based on RT-DFS without external variables

6.2. *Performances of real-time PTDP models based on the proposed RWH-DFS without external variables*

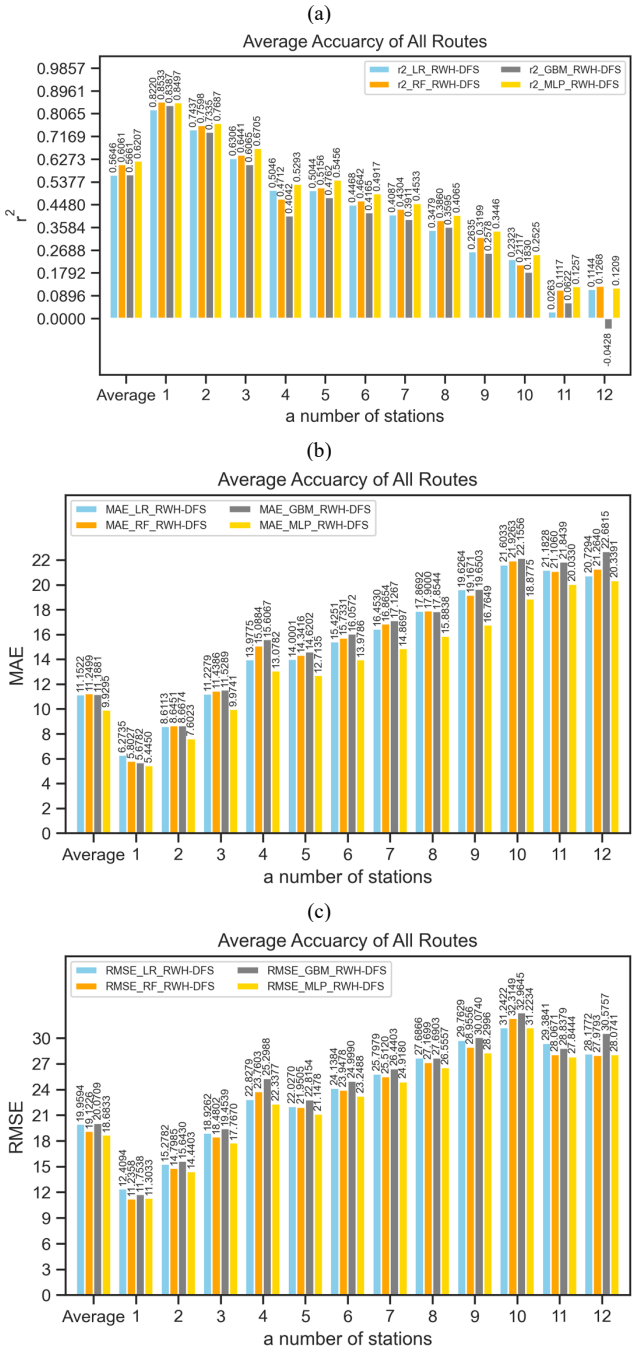


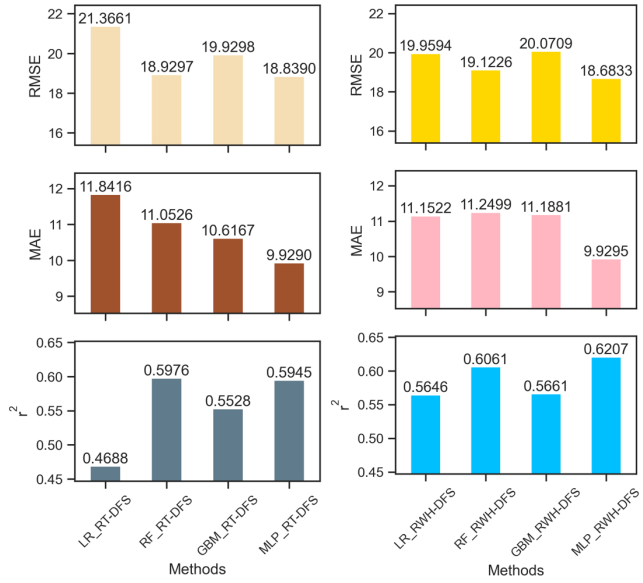
Fig 5. Average performance of the proposed real-time PTDP models based on RT-DFS without external variables

Fig. 5 shows the average performances in terms of R^2 (a), *MAE* (b), and *RMSE* (c) of PTDP models built by using the proposed RWH-DFS for all Amtrak routes R364, R370 and R350 without external variables.

The average R^2 of the proposed RWH-DFS based PTDP models using MLP technique is about 0.62 which outperformed the others using RF, GBM and LR yielding R^2 of 0.61, 0.57, and 0.54, respectively. Additionally, Fig4 (a) indicates that the proposed RWH-DFS eliminates the limitation of the PTDP models using all ML techniques to predict PTD in term of the number of stations between S_C and S_P, whereas the limitation of the RWH-DFS based model using LR is enhanced to be 11 stations (compared to the RT-DFS based model using LR).

Fig 5 (b) also shows that the average MAE of the proposed RWH-DFS based PTDP models using MLP is about 9.9 minutes, which is better than all other models yielding MAE of more than 11 minutes. In addition, the RMSE of RWH-DFS based models using MLP is about 18.7, which is also better than the other RWH-DRS based models using RF, GBM, and LR up to 1.3 minutes.

To sum up, the real-time RWH-DFS based PTDP models using MLP technique performs better than the models using RF, GBM, and LR approaches, respectively.



(a) RT_DFS (b) RWH-DFS

Fig 6. Average performance of real-time PTDP models using RT-DFS (a) and RWH-DFS (b) without external variables

Considering the average performances of real-time PTDP models built using the proposed RT-DFS and RWH-DFS as shown in tables Fig 6 (a) and 6 (b), we can see that the proposed RWH-DFS helps significantly improve the performance of PTDP models in terms of R^2 regardless of ML technique used.

Overall, the results in section 6.1 and 6.2 indicate that the proposed RWH-DFS based PTDP models using MLP yields the highest performance. Thus, it is used to evaluate the how each external variable influence the performance of the most effective PTDP models of this work in the following section.

6.3. *Performance of the proposed real-time PTDP models based on RWH-DFS with external variables*

Fig. 7 shows average performances of the real-time PTDP models using MLP and RWH-DFS with external variables listed in Table 1 for all Amtrak routes. Overall, the average performance of all Amtrak routes as shown in Fig. 7 indicates that the passenger-related external variables, which are

ridership (rt) and population (pt), can help improve the performance of the proposed models in terms of R^2, *MAE*, and *RMSE* up to 2.1%, 2.1%, and 1.2%. Historical delay profiles at destination- (HDPD) related variables, especially the median of HDPD of the past 14 days, can also improve R^2, *MAE*, and *RMSE* of the models up to 5.2%, 5.9%, and 3.6%, respectively. Weather related external variables, namely snow hourly (snow_hrly), precipitation hourly (precip_hrly), feels-like temperature, wind chill (WC), pressure tend, dew point (dewPt), wind speed (WSPD) and UV index can also help increase the performance of the proposed models in terms of *R2, MAE, and RMSE* up to 1%.

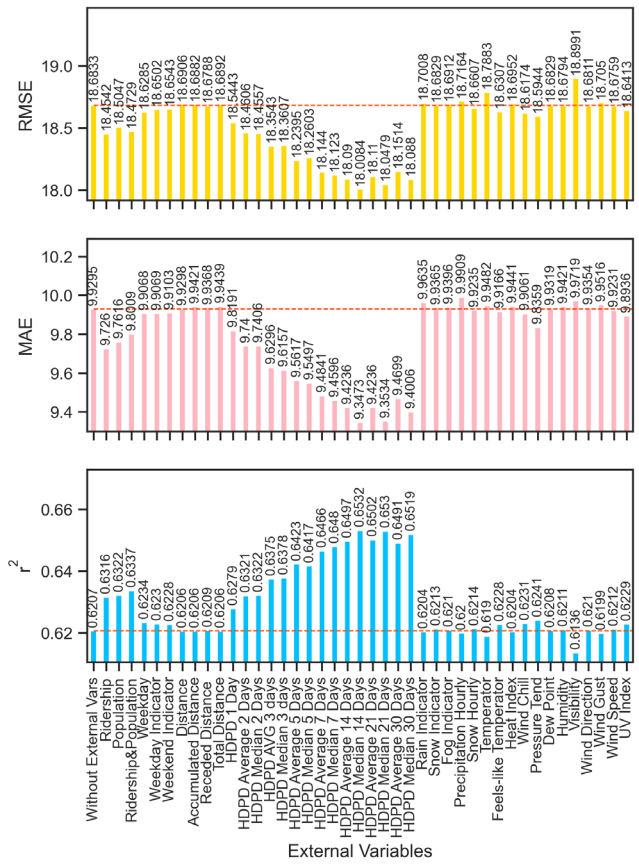
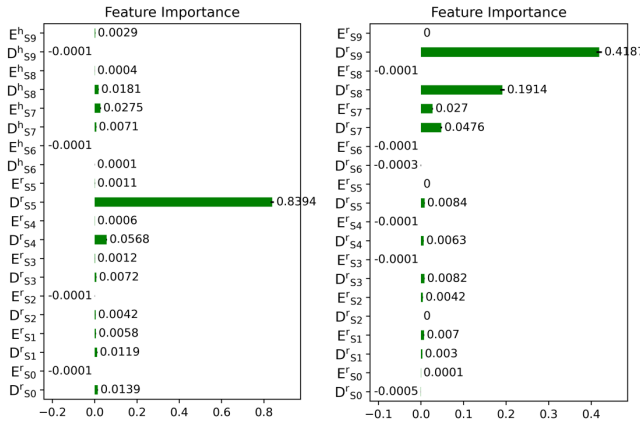


Fig 7. Average performance of the proposed real-time PTDP models using RWH-DFS with external variables

6.4. *Discussion and Interpretations with Further Analysis*

The average performances as shown in Figs. 5 and 6 show that the PTDP models based on MLP technique outperformed the others using RF, GBM, and LR. Besides, applying the proposed RWH-DFS making use of historical data of the previous train ($T_{C\text{-}1}$) helps improve the performance of the PTDP models, when compared to the models using RT-DFS which uses only real-time data of the current train (T_C) to predict PTD at destinations.

Moreover, the proposed RWH-DFS helps lower the limitation of PTDP models using ML techniques (RF, GBM, and MLP) to be able to predict unknow train delay at destination since the train arrived the first station (the number of stations between S_C and S_P is 12), whereas the models using RT-DFS and ML techniques can be used to predict PTD after the second station (the number of stations between S_C and S_P is 11) and after the 4th stations for the model using RT-DFS with LR technique. Furthermore, the performance of PTDP models using the proposed RWH-DFS steadily increased when the number of stations between S_C and S_P decreased as the current train is approaching the destination. For example, the performance in term of R^2 of PTDP models using MLP and the proposed RWH-DFS increased approximately by 20% from 0.12 to 0.82 when the number of stations between S_C and S_P reduced from 12 to 1.

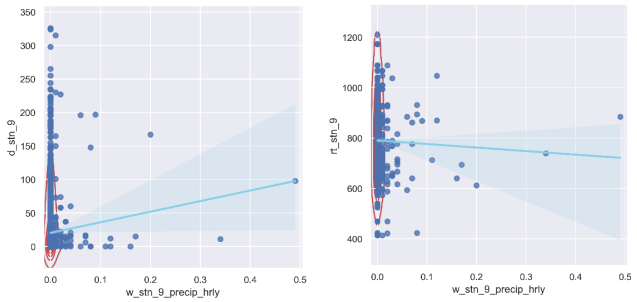


(a) S_C = S5, S_P = S10 (b) S_C = S9, S_P = S10

Fig 8. Feature importance (FI) of endogenous ($D^r_{S_i}$ *or* $D^h_{S_i}$) and exogenous ($E^r_{S_i}$ or $E^h_{S_i}$) variables at each station of the MLP-based PTDP models using the proposed RWH-DFS where the current train (T_C) is at (a) S5 and (b) S9 to predict PTD at S10 for Amtrak R364.

As mentioned in section 5, an analysis of feature importance (FI) can determine the influence of each independent variable influence on the proposed PTPD models. Fig. 8 (a) shows an example of the future importance (FI) of both PTD at each station ($D^r_{S_i}$ *or* $D^h_{S_i}$) and external variables ($E^r_{S_i}$ or $E^h_{S_i}$) which is precipitation hourly in this case of the most effective PTDP model using RT-DFS and MLP to predict PTD of Amtrak route R364 at the last station S10 (S_P = S10) when the current train (T_C) is at station S5. The most influential variables in this scenario are PTD of T_C at station S5 ($D^r_{S_5}$) where the train is located, PTD at the previous station S4 ($D^r_{S_4}$), and followed by some other variables such as external variable at S8 ($E^h_{S_7}$) and PTD of the previous train ($T_{C\text{-}1}$) at S8 ($D^h_{S_8}$), respectively. Similarly, Fig. 8 (b) shows FIs of the PTDP model using RT-DFS and MLP to predict PTD of Amtrak route R364 at station S10 when the current train (T_C) is at station S9. In this scenario, the most influential variables are also PTD of T_C at S9 ($D^r_{S_9}$) where the train is currently located and followed by PTD at the previous station S8 ($D^r_{S_8}$) and S7 ($D^r_{S_7}$), and external variable at S7 ($E^r_{S_7}$), respectively. This indicated that the PTD at the destination (D^*_P) is significantly influenced by the latest real-time train delay at the current station ($D^r_{S_C}$) and followed by a few prior stations if available, and then by some other external variables, and historical PTD of the previous train ($T_{C\text{-}1}$). This also implied that train behave differently each time resulting in different PTDs. Thus, using the proposed

PTDP based on RWH-DFS and ML techniques, especially MLP is more effective than relying on average delay to estimate or predict PTD at destination even it is the same station and train route.



(a) (b)

Figure 9. Relationships between precipitation hourly and passenger train delay (a), and between precipitation hourly and ridership (b) at station 9 of Amtrak train route R364

In addition to performance interpretations above, further analysis on some external variables can also help provide more insights of how each of them influences PTD. For example, Fig. 9(a) shows the relationship between precipitation hourly (x-axis) and passenger train delay (y-axis) of Amtrak train R364 at station S9. It shows that an increased precipitation hourly results in higher train delay. Fig. 9(b) shows the relationship between precipitation hourly (x-axis) and ridership (y-axis). The figure suggests that an increased precipitation leads to a reduction of ridership as it may not be convenient for riders to commute while it is raining, but PTD at destination increased. Thus, these can be interpreted that the reduction of ridership due to a higher precipitation does not lead to the reduction of PTD, but the increase in PTD instead because the train has to spend more time at each station resulting in higher dwell-time and eventually causing more delay at the next stations.

CONCLUSION

This article proposed real-time PTDP models using LR as benchmark and three ML techniques, namely RF, GBM, and MLP, with and without several external variables from various data sources. In addition, we also proposed RT-DFS using only real-time data of the current train (T_C) and RWH-DFS making use of historical data of the previous train (T_{C-1}) to construct a data-frame panel as an input for the PTDP models. The results show that the performance (in terms of R2, MAE, RMSE) of the PTDP models using MLP approach outperformed the models using other techniques. Furthermore, applying RWH-DFS to MLP-based models yield even better performance.

In addition, we also showed that considering and including exogenous data, especially ridership, population, HDPD, and weather information as additional independent variables can improve the performance of the models in terms of R^2, MAE, and RMSE by 5.2%, 5.9%, and 3.6%, respectively. Looking at the impact of individual variables, we found that the FI of PTD at the latest current station (S_C) where the train is located is the most influential variable on PTD and it also dominates all other independent variables. In addition, we found that precipitation hourly can affect PTD during rainy conditions.

The proposed PTDP models in this work are designed for predicting PTD at a specific destination of a given train route. Thus, many station-to-station PTDP models are required to construct a PTDP system. Future work should focus on developing more versatile models to predict PTD at any station of interest with less time. Moreover, other related endogenous or exogenous variables, especially freight-train related information could be further examined and studied in the future work as they may have significant impact on PTD.

REFERENCES

[1] S. Derrible, *Urban Engineering for Sustainability*. Cambridge, MA, USA: MIT Press, 2019.

[2] R. Nilsson and K. Henning, *Predictions of train delays using machine learning*. 2018. Accessed: Jul. 27, 2019. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-230224

[3] Amtrak, "Amtrak Five Year Service Line Plans FY20-24," 2019.

[4] N. O. E. Olsson and H. Haugland, "Influencing factors on train punctuality—results from some Norwegian studies," *Transport Policy*, vol. 11, no. 4, pp. 387–397, Oct. 2004, doi: 10.1016/j.tranpol.2004.07.001.

[5] W. Peetawan and K. Suthiwartnarueput, "Identifying factors affecting the success of rail infrastructure development projects contributing to a logistics platform: A Thailand case study," *Kasetsart Journal of Social Sciences*, vol. 39, no. 2, pp. 320–327, May 2018, doi: 10.1016/j.kjss.2018.05.002.

[6] P. Wang and Q. Zhang, "Train delay analysis and prediction based on big data fusion," *Transportation Safety and Environment*, vol. 1, no. 1, pp. 79–88, Jul. 2019, doi: 10.1093/tse/tdy001.

[7] L. Oneto *et al.*, "Train Delay Prediction Systems: A Big Data Analytics Perspective," *Big Data Research*, vol. 11, pp. 54–64, Mar. 2018, doi: 10.1016/j.bdr.2017.05.002.

[8] Z. Alwadood, A. Shuib, and N. Abd. Hamid, "Rail passenger service delays: An overview," in *2012 IEEE Business, Engineering Industrial Applications Colloquium (BEIAC)*, Apr. 2012, pp. 449–454. doi: 10.1109/BEIAC.2012.6226102.

[9] S. Milinković, M. Marković, S. Vesković, M. Ivić, and N. Pavlović, "A fuzzy Petri net model to estimate train delays," *Simulation Modelling Practice and Theory*, vol. 33, pp. 144–157, Apr. 2013.

[10] B. W. Schlake, C. P. L. Barkan, and J. R. Edwards, "Train Delay and Economic Impact of In-Service Failures of Railroad Rolling Stock," *Transportation Research Record*, vol. 2261, no. 1, pp. 124–133, Jan. 2011, doi: 10.3141/2261-14.

[11] R. Wang and D. B. Work, "Data Driven Approaches for Passenger Train Delay Estimation," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 535–540. doi: 10.1109/ITSC.2015.94.

[12] L. Oneto *et al.*, "Advanced Analytics for Train Delay Prediction Systems by Including Exogenous Weather Data," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2016, pp. 458–467. doi: 10.1109/DSAA.2016.57.

[13] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich, "Traveling time prediction in scheduled transportation with journey segments," *Information Systems*, vol. 64, pp. 266–280, Mar. 2017, doi: 10.1016/j.is.2015.12.001.

[14] A. Estes, M. O. Ball, and D. Lovell, "Predicting performance of ground delay programs," presented at the 12th USA/Europe air traffic management R&D seminar, Seattle, WA, 2017.

[15] R. Gaurav and B. Srivastava, "Estimating Train Delays in a Large Rail Network Using a Zero Shot Markov Model," *arXiv:1806.02825 [cs, stat]*, Jun. 2018, Accessed: Apr. 15, 2019. [Online]. Available: http://arxiv.org/abs/1806.02825

[16] R. Nair *et al.*, "An ensemble prediction model for train delays," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 196–209, Jul. 2019, doi: 10.1016/j.trc.2019.04.026.

[17] P. Taleongpong, S. Hu, Z. Jiang, C. Wu, S. Popo-Ola, and K. Han, "Machine learning techniques to predict reactionary delays and other associated key performance indicators on British railway network,"

Journal of Intelligent Transportation Systems, vol. 0, no. 0, pp. 1–28, Dec. 2020, doi: 10.1080/15472450.2020.1858822.

[18] C. M. Zappi, "Amtrak Host Railroad Report Card 2019," Jan. 2020.

[19] Amtrak, "Amtrak Host Railroad Report Card 2020," Apr. 2021. Accessed: Aug. 17, 2021. [Online]. Available: https://www.amtrak.com/content/dam/projects/dotcom/english/public/documents/corporate/HostRailroadReports/Amtrak-2020-Host-Railroad-Report-Card-FAQs.pdf

[20] P. Lapamonpinyo, S. Derrible, and F. Corman, "A Python tool and database of Amtrak departure and arrival times with weather information." Oct. 19, 2021. doi: 10.31224/osf.io/rhxcj.

[21] Y. Ding, "Predicting flight delay based on multiple linear regression," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 81, p. 012198, Aug. 2017, doi: 10.1088/1755-1315/81/1/012198.

[22] Yale, "Linear Regression," 98 1997. http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm (accessed Jul. 31, 2020).

[23] K. M. Ramachandran, *Mathematical statistics with applications in R*, Second edition. Amsterdam: Elsevier, 2015.

[24] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[25] J. Han, M. Kamber, and J. Pei, *Data Mining*. Elsevier, 2012. doi: 10.1016/C2009-0-61819-5.

[26] T. Hastie, R. Tibshirani, and J. Friedman, "Random Forests," in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, T. Hastie, R. Tibshirani, and J. Friedman, Eds. New York, NY: Springer, 2009, pp. 587–604. doi: 10.1007/978-0-387-84858-7_15.

[27] H. Singh, "Understanding Gradient Boosting Machines," *Medium*, Nov. 04, 2018. https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab (accessed Mar. 15, 2021).

[28] T. Hastie, R. Tibshirani, and J. Friedman, "Additive Models, Trees, and Related Methods," in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, T. Hastie, R. Tibshirani, and J. Friedman, Eds. New York, NY: Springer, 2009, pp. 295–336. doi: 10.1007/978-0-387-84858-7_9.

[29] V. Silaparasetty, "Neural Networks," in *Deep Learning Projects Using TensorFlow 2: Neural Network Development with Python and Keras*, V. Silaparasetty, Ed. Berkeley, CA: Apress, 2020, pp. 71–86. doi: 10.1007/978-1-4842-5802-6_3.

[30] J. Moolayil, "An Introduction to Deep Learning and Keras," in *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*, J. Moolayil, Ed. Berkeley, CA: Apress, 2019, pp. 1–16. doi: 10.1007/978-1-4842-4240-7_1.

[31] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*, Second edition. Piscataway, New Jersey: IEEE Press, 2011. doi: 10.1002/9781118029145.

[32] S. Abirami and P. Chitra, "Chapter Fourteen - Energy-efficient edge based real-time healthcare support system," in *Advances in Computers*, vol. 117, 1 vols., P. Raj and P. Evangeline, Eds. Elsevier, 2020, pp. 339–368. doi: 10.1016/bs.adcom.2019.09.007.

[33] C. Sammut and G. I. Webb, Eds., "Mean Absolute Error," in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2010, pp. 652–652. doi: 10.1007/978-0-387-30164-8_525.

[34] S. Billiau, "From Scratch: Permutation Feature Importance for ML Interpretability," *Medium*, Jun. 17, 2021. https://towardsdatascience.com/from-scratch-permutation-feature-importance-for-ml-interpretability-b60f7d5d1fe9 (accessed Sep. 21, 2021).

[35] "Permutation Importance — ELI5 0.11.0 documentation." https://eli5.readthedocs.io/en/latest/blackbox/permutation_importance.html (accessed Feb. 12, 2022).