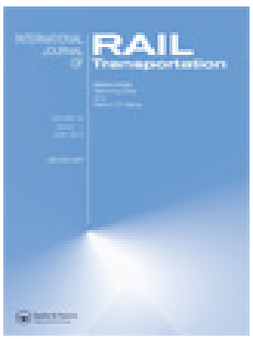


International Journal of Rail Transportation

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjrt20

Near-term train delay prediction in the Dutch railways network

ZhongCan Li , Chao Wen , Rui Hu , Chuanlin Xu , Ping Huang & Xi Jiang

To cite this article: ZhongCan Li , Chao Wen , Rui Hu , Chuanlin Xu , Ping Huang & Xi Jiang (2020): Near-term train delay prediction in the Dutch railways network, International Journal of Rail Transportation, DOI: 10.1080/23248378.2020.1843194

To link to this article: https://doi.org/10.1080/23248378.2020.1843194

INTERNATIONAL JOURNAL OF RAIL TRANSPORTATION
https://doi.org/10.1080/23248378.2020.1843194

Near-term train delay prediction in the Dutch railways network

ZhongCan Li[a], Chao Wen[a], Rui Hu[a], Chuanlin Xu[a], Ping Huang[a] and Xi Jiang[b]

[a]School of Transportation & Logistics, Southwest Jiaotong University, Chengdu, China; [b]State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China

ABSTRACT

Due to the unsuitable train delay prediction methods currently used in the Netherlands, a more accurate delay prediction method is needed. In this work, based on the data provided by the *2018 RAS Problem Solving Competition: Train Delay Forecasting*, a data-driven model is established to predict the delay 20 min later. By combining the current delay with the operating conditions, the influencing factors that may influence delay propagation are extracted after analysing the delay propagation mechanisms and train movement data structure. These factors are considered as model input features for random forest regression, via which a prediction model is established. It is found that the random forest model exhibits high prediction accuracy and fast callback in terms of the training model, and ANN, XGBOOST, GBDT, and statistical algorithms are applied as benchmark algorithms. Finally, to complete the study, the importances of different delay influencers are investigated, calculated, and discussed.

ARTICLE HISTORY
Received 6 March 2020
Revised 20 October 2020
Accepted 24 October 2020

KEYWORDS
Train operation; Delay prediction; Data-driven; Random forest

1. Introduction

A railway transportation system is an extremely complex multi-subsystem and multi-departmental cooperative system. During train operation, this system will be subjected to various operating circumstances, facilities utilization, and organizational management issues, which may lead to trains being influenced by the weather, system mechanical failure, or organization strategies. This could result in resource conflicts, and, ultimately, train delays. Train punctuality is an important factor considered by railway companies when evaluating transport service quality. However, train delays are inevitable. For instance, in Norway, the punctuality ratio of the best-operating railway has reportedly reached 94%, while the worst railway has achieved just 80%, with a wide range of factors causing delays [1]. Even in Japan, which is reputed to have the best high-speed railway (HSR) operating quality, a 0.9-min delay per train has been reported [2]. Delays disrupt normal operations and reduce railway system reliability, which in turn decreases the attractiveness of railways as a transport option to both passengers and companies, which could motivate them to choose other transportation modes.

CONTACT Ping Huang huangping129@my.swjtu.edu.cn School of Transportation & Logistics, Southwest Jiaotong University, Chengdu China

Delay propagation results from the influences of multiple factors, of which the timetable structure, operating environment, and infrastructure conditions will all affect the dispatching decision-making of operators. Accurate train delay prediction in terms of the train operating conditions and real-time timetable structure is vital for rescheduling, and can help train dispatchers to make more informed dispatching decisions. To facilitate intelligent railway dispatching and improve the decision-making quality of train operation control, highly efficient and accurate models that can predict train delay propagation are critically needed.

At present, however, most delays are predicted empirically, which leads the delay prediction accuracy to depend primarily on the experience and skills of the dispatcher. In the Netherlands, for example, the most commonly used estimation method for future delays is to assume that the current delay will remain unchanged [3]. This method is highly uncertain and is not supported by scientific theory. Fortunately, with the development of railway storage devices and computer science, real-time train operation data can be collected and stored, making it possible to uncover delay propagation processes using a new, data-driven perspective. Data-driven methods require neither a large amount of detail nor explicit knowledge of the physical behaviour of the system; instead, they are established using actual historical data. This data is the consequence of all pertinent influencing factors, and may reveal delay propagation influencing factors, periods, or regularity better than the current non-science-based methods.

The *2018 RAS Problem Solving Competition: Train Delay Forecasting* provided a real train operation dataset in which the timetable structure and train operating conditions were given. Concerning the known *2018 RAS Competition* results, in a previous study [4], a delay prediction model was established based on the competition dataset. However, the study only randomly employed 18,000 data samples provided by the committee, and did not consider the influences that various current delay times may have on the dispatching strategy. Another study [5] used a neural network to complete the competition requirements; however, the details are unavailable.

On the basis of this dataset, the purpose of the present study is the establishment of an accurate prediction model that can predict train delay 20 min later (because some trains do not arrive at the recording point exactly 20 min later, the delay of these trains is predicted at the nearest 20-min recording point) according to the current delay, the related timetable structure, and the train operating conditions. Thus, the dispatchers can master the delay earlier, which provides scientific support to create a rescheduling plan. In this study, the influencing factors which could be considered as model inputs are first extracted after analysing the delay prediction propagation process and dataset structure. Then, the dataset is divided into two categories in terms of the current delay. The random forest (RF) regression algorithm, for which hyperparameter optimization is conducted after parameter selection, is applied to separately establish prediction models for these two categories. Some widely-applied algorithms are then used as benchmarks for comparison with the RF algorithm, which allows for the comparision of the prediction precision of the RF model.

The contributions of this study include the following. (1) In accordance with disposal strategies for different delay times, the delays are split into two datasets and respective delay prediction models are established. (2) These established delay prediction models consider the influences of time and space on the train operation itinerary. Moreover, the

importance of each influence factor is calculated; this can ascertain which factors have priority when a delay occurs, and may provide some inspiration for planners to schedule the timetable. (3) Both prediction models are found to have high predictive accuracy, and could contribute to dispatchers conducting scientific and reasonable dispatching in real-time.

The remainder of this paper is organized as follows. In Section 2, other work related to delay prediction is briefly reviewed. In Section 3, the problem addressed in this study is defined, and the data and data management processes used in this study are introduced. The details of the approach used in the delay prediction model are reviewed in Section 4, and a comparison with the precision of other models is presented. Finally, the conclusions are presented in Section 5.

2. Literature review

Before machine learning methods prevailed, intelligent computing methods, such as fuzzy networks, the Bayesian method, and graph models, were considered as efficient predictors of train delays. One such method was the fuzzy Petri net (FPN) model, in which expert knowledge was used to define the fuzzy sets and rules, and expertise was transformed into a model to calculate train delays and conflicts in a section of the Belgrade railway [6]. Wen studied triangular fuzzy number workflow nets in an HSR running state prediction model, and the fuzzy times for train activities were generated using data over 21–24 June 2012 at five stations between Beijing-South and Dezhou-East on the Beijing-Shanghai HSR [7]. Several studies have shown that graph models can be employed to predict train status (e.g., running times, dwelling times, arrival/departure times, train delays, etc.). By combining the work of dispatching decisions, station work plans (e.g., the train path plan, waiting policies, track conditions, etc.), and interactions among trains, the data of the Rotterdam C-The Hague HS railway line [8], the German train schedule [9], the Leiden–Dordrecht corridor in the Netherlands [10], and the Hague Central to Venlo line on the basic Dutch railway [11] have been employed to evaluate the applicability of graph-based models.

Train delay propagation has been considered as a stochastic process, and many related data-driven methods have been applied to model delay propagation [12]. For instance, in some studies, the Markov chain has been proven as an efficient way to predict train delays. Barta [13] applied the data provided by a Swiss company to evaluate train delay evolution when a train visits successive terminals. Additionally, according to data from Turkish State Railways, Şahin [14] classified the delay times into six states, and then exerted the Markov chain to calculate the probability of arrival and departure delay of each state after passing a particular number of section runs or experiencing a particular number of conflicts. Assuming that the probability of a state change relies on the moment of transition, train delay predictions have been modelled using a non-stationary Markov chain, and this model has been validated on a portion of the HSR between Beijing and Shanghai in China [15]. In addition, Bayesian network models are widely used. Zilko [16] proposed a Copula Bayesian Network that considered the time, location, weather, and presence of an overlapping disruption as influencing factors to predict the disruption length, and this model was verified to be sound based on the incident data of the Dutch railway network between 1 January 2011 and 30 June 2013. In another study, a hybrid

Bayesian network model was used to predict HSR delays via the use of the Wuhan-Guangzhou HSR train operating records; the proposed model achieved an average accuracy of over 80% for predictions within a 60-min horizon [17]. Corman [18] validated the applicability of Bayesian networks via a realistic case study from a busy railway between Stockholm and Norrköping in Sweden. Furthermore, Huang et al. [19] investigated the interactions between the three influencing indicators of delay, namely the primary delay, the number of affected trains, and the total delay times, and confirmed that the hybrid Bayesian network revealed the evolutions of the three indicators at subsequent stations. This model used data from the Wuhan-Guangzhou and Xiamen-Shenzhen HSR lines in China.

In recent years, machine learning methods have become widely used due to their better fit and straightforward interpretation. Statistical and machine learning models have been used for the prediction of the lengths of running and dwelling times with high accuracy via the use of delay history data from Rotterdam and The Hague in the Netherlands between March and May 2010 [20]. In addition, while the neural network method was developed in the 20th century, it has recently become more widely used as a method for delay estimation. Neural networks have been applied to estimate passenger train delays using data in Germany [21] and Iran [22]. Oneto [23,24] proposed a neural network method called Deep Extreme Learning Machines to solve the train delay prediction problem, and this method was compared with the original extreme learning machines and the prediction system of the data provider Rete Ferroviaria Italiana. The results demonstrated that the Deep Extreme Learning Machines outperformed the other two methods when using more than one year of data from two main areas in Italy. Deep learning approaches, such as long short-term memory recurrent neural networks, have been applied to train delay prediction via the use of data from the Netherlands and China [25,26]. However, their use is limited to a single railway line due to their sequence input format.

Marković et al. [27] first presented a support vector regression (SVR) approach to address train delay estimation; when using data for Rakovica Station on the Belgrade Railway in Serbia, it was found that the SVR technique outperformed the ANN algorithm. Later, by taking into account the train properties and railway network structure, Barboura et al. [28] used SVR to predict freight train arrival times on a US railway. Additionally, a hybrid model that is integrated with SVR and a Kalman filter has been proposed to improve the accuracy of SVR when predicting the running times of high-speed trains on a section of the Wuhan-Guangzhou HSR [29]. The *k*-nearest neighbours algorithm has also shown good prediction accuracy as compared with traditional statistical methods when applied to historical data from Thai passenger trains [30]. Li et al. [31] also applied the *k*-nearest neighbours algorithm to estimate the dwell time using data from selected Dutch railway stations.

As an ensemble decision tree model that has shown good precision and rapid operation, the random decision forest has been widely used in many fields. After analysing the historical railway operation data from the Taiwan Railways Administration, Lee [32] employed a decision tree to propose a knock-on delay root cause discovery model, which can be applied to ascertain the root cause and propagation process of a delay. Based on data from the Wuhan-Guangzhou HSR, Jiang et al. [33] compared the prediction accuracy of delay recovery by several machine learning methods, including the RF,

multiple linear regression, SVR, and ANN algorithms. The results indicated that the RF algorithm outperformed the other baselines, and had the highest accuracy of up to 80.4% when the required error was <1 min. Lulli et al. [34] presented a hybrid approach that combined the decision tree and RF regression to predict the running time, dwell time, train delay, and penalty costs. This hybrid model represented a merger of data-driven and experience-based model approaches, and performed better than either. Gaurav and Srivastava [35] considered delay propagation to be an *n*-order Markov model. The prediction performances achieved by the RF and ridge regressions were compared using the delay of the previous station as the model input. The results showed that most trains in the target (Indian) railway followed a first-order Markov process, and that the RF regression had a better effect than the ridge regression process. Moreover, the interactions between the previous station and the target station were explored.

A summary of the preceding literature review is presented in Table 1.

Table 1. A summary of the review of delay prediction research.

Method	Citation number	Data source	Contributions
Fuzzy Petri net	[6]	A portion of the Belgrade railway	Delay and conflict prediction
	[7]	Five stations on the Beijing-Shanghai HSR	Running state prediction
Graph model	[8]	The Rotterdam C-The Hague HSR line	Running and dwelling time prediction
	[9]	The German train schedule	Departure and arrival time prediction
	[11]	The Leiden–Dordrecht corridor in the Netherlands	Event times (running times, conflicts, and blocking times)
	[12]	Hague Central to Venlo line on the Dutch railway	Delay prediction
Markov chain	[13]	A Swiss company	Delay prediction
	[14]	Turkish State Railways	Arrival and departure delay prediction
	[15]	A portion of the HSR between Beijing and Shanghai	Arrival and departure delay prediction
Bayesian network	[16]	Dutch railway network	Disruption length prediction
	[17]	Wuhan-Guangzhou HSR	Arrival and departure delay prediction
	[18]	A railway between Stockholm and Norrköping in Sweden	Passenger and freight train delay prediction
	[19]	Wuhan-Guangzhou and Xiamen-Shenzhen HSRs	Delay influence prediction
Statistical learning	[20]	Rotterdam and The Hague in the Netherlands	Running and dwell time prediction
Neural network	[21]	Deutsche Bahn German	Delay prediction
	[22]	Iran	Delay prediction
	[23]	Two main areas in Italy	Delay prediction
	[24]	Two main areas in Italy	Delay prediction
	[25]	Railway network in the Netherlands and China	Delay prediction
	[26]	The Netherlands	Delay prediction
SVR	[27]	Rakovica Station (Belgrade Railway, Serbia)	Arrival delay prediction
	[28]	A US railway	Train arrival time prediction
	[29]	Wuhan-Guangzhou HSR	Running time prediction
KNN	[30]	Thai passenger trains	Arrival time prediction
	[31]	Dutch railway stations	Dwell time prediction
Ensemble Decision tree model	[32]	Taiwan Railways	Ascertainment of the root cause of delay
	[33]	Wuhan-Guangzhou HSR	Delay recovery prediction
	[34]	One large Italian region (Liguria)	Running time, dwell time, train delay, and penalty cost prediction
	[35]	Indian railway	Arrival delay prediction

From the summarized literature, it is clear that there is no definitive evidence that some specific algorithms always outperform others. The optimal algorithms vary due to differences in the train operating environment, equipment, organization, timetable structure, etc. Therefore, algorithm selection is necessary, and the algorithm with the best accuracy should be chosen to establish the delay prediction model. Therefore, in the present work, the optimal prediction model is ultimately determined via algorithm comparison.

3. Problem statement and data description

3.1. Problem statement

In the Netherlands, more than one million passengers travel by train daily, and the largest passenger train operating company, Netherlands Railways, operates almost 6000 trains each day. During a train journey, there can be many unpredictable disturbances that can lead to inevitable delays. According to the data provided by the *2018 RAS Problem Solving Competition: Train Delay Forecasting*, the percentage of delayed trains (delayed more than 0 min) in the Netherlands railways network is about 40%. Train delay has become a crucial problem, and accurately predicting future delays ahead of time can benefit both the passengers and the railway operation dispatching system. However, the most common future delay estimation method has been to assume that the current delay will remain unchanged, i.e., that the train delay will neither increase nor decrease downstream. This is clearly unreasonable, and the establishment of a reliable prediction model has therefore become very important. This not only could provide more reasonable dispatch strategies, but could also lay the foundation for the introduction of intelligent transport.

Recently, real operating data regarding delays has been collected across the entire network of Netherlands Railways and has been made available to the operators. This data can be used for predicting train delay by combining the latest machine learning and data mining algorithms. The *2018 RAS Problem Solving Competition: Train Delay Forecasting* provided a real train operation dataset of the entire network of Netherlands Railways. The data provided by the competition committee covered the period from 4 September to 9 December 2017, and the last four Tuesdays of this period (2017–11-14, 2017–11-21, 2017–11-28, and 2017–12-05) were excluded and used as the test dataset. Based on the dataset, the competition committee first required participants to train their models with the data from 4 September to 9 December 2017 without the data from the last four Tuesdays. Then, given the state of the network on those last four Tuesdays at 08:00, 12:00, and 16:00, participants were instructed to predict the delays for the state of the network at 08:20, 12:20, and 16:20.

Using the dataset provided by the competition committee, the research reported in this paper aimed to predict the train delay up to approximately 20 min later after a given current delay for all trains at any given moment in time. However, to judge the prediction precision of different participants, only the states at 08:00, 12:00, and 16:00 on the last four Tuesdays of the given period were provided in the competition, while the real delays of 20 min later (i.e., 08:20, 12:20, and 16:20, respectively) were not provided to the participants; thus, this data was inaccessible to calculate the prediction accuracy.

Therefore, in this study, the training dataset contained all days excluding the last four Mondays, and the data from the last four Mondays were used as test data to validate the prediction accuracy. Different from the original competition, the delays 20 min later of all trains in the test dataset were predicted. In addition, because some trains did not arrive at the recording point exactly 20 min later, the delay of these trains was predicted at the nearest 20-min recording point. In summary, the purpose of this study was to extract and analyse the influencing factors of delay propagation, and then to establish a prediction model based on the historical data to minimize the errors between the actual and predicted delays 20 min later. After determining the current state of a train, the future state can be predicted earlier. A schematic of the problem is presented in Figure 1.

3.2. Data description

The data used in this study covered the period from 4 September to 9 December 2017, excluding the data from the last four Tuesdays, and was sourced from the *2018 RAS Problem Solving Competition: Train Delay Forecasting*. Only weekday data was used, because although the weekends generally had the same train schedules, operations were often affected by maintenance work. Also, the timetables were identical for Mondays, Tuesdays, Thursdays, and Fridays; because there were extra trains operated on a busy part of the network between Eindhoven and Amsterdam on Wednesdays, a different timetable was applied on that day [3]. The raw data mainly included the following information.

(1) (1) **Planned timetable – the planned train running times**

The planned timetable consisted of the scheduled arrival times for each train number. The buffer time, which is the resource required to make up for a train delay, could be calculated based on the planned timetable.

(1) (2) **Actual historical train performance – actual train running times**

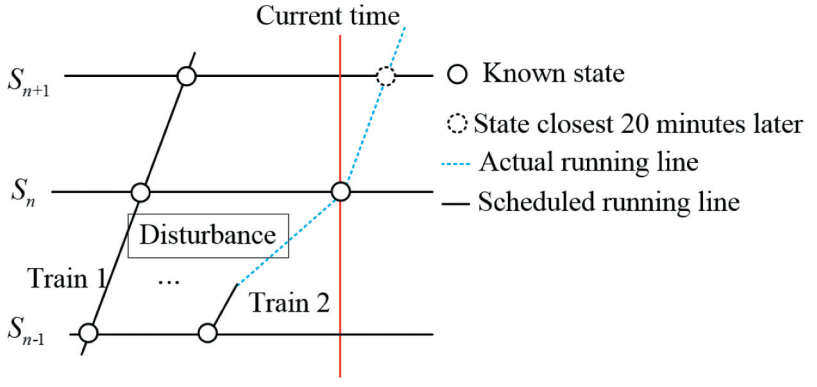


Figure 1. Problem statement.

The raw data provided the actual historical train performance, with actual and planned times for each train recorded in units of seconds. However, in the real dispatching process, dispatchers are more concerned with the delay time in terms of minutes. For instance, if a train delay is 2 minutes and 40 seconds, dispatchers will deem this delay as 2 minutes. Therefore, in this work, delays were calculated and predicted in terms of minutes.

(1) (3) **Crew schedules – the plan for crew changes on a train**

Ordinarily, drivers stay on the same train during stops. Occasionally, however, the drivers are changed. The data contained all driver changes, including the driver changes at the terminus of a train number.

(1) (4) **Rolling stock circulation – the plan for rail cars being assigned to trains**

This dataset included the plans for rail cars being assigned to trains. Generally, a train operates with the same rolling stock composition after a stop; however, it was also possible for the composition to change, and any such changes were recorded.

(1) (5) **Infrastructure data – describes the node and link structure of the railway**

This data included an overview of travel distances between locations.

(1) (6) **Overview of weather conditions – daily weather conditions**

An average weather condition summary for each day was obtained from the Dutch weather agency.

Train delays can be caused by timetable and infrastructure shortcomings, as well as bad weather. Because weather conditions were recorded daily, the weather conditions for each train operation were unknown; thus, the weather condition dataset was regarded as negligible. To obtain a better understanding of the influencing factors, the train operating data, including scheduled and historical timetables, as well as the infrastructure data, such as section lengths, were obtained from the Netherlands railways operations dataset. Considering the train operating process and the timetable structure that contains the itinerary each train has passed and will go through, for this study, eight parameters that were likely to have impacts on future train delays were selected as the feature space (F):

(1) Delays at the starting station (X_1);
(2) Current delays (X_2);
(3) Planned time in the travelled section (X_3);
(4) Actual travel time (X_4);
(5) Planned time from now until 20 min later (X_5);
(6) Distance travelled (X_6);
(7) Distance from now to 20 min later (X_7);
(8) Period of delay occurrence (X_8), which is the time of day classified by the hour.

Furthermore, delays 20 min later were regarded as the dependent variable y. All variables related to time were calculated in minutes.

Based on these 8 factors, a total of 5,624,764 train operation data records were extracted from the metadata. Several preprocessing steps were applied before modelling, including the following: (1) filling in missing data using the weighted means of adjacent records; (2) deleting abnormal delays (longer than 20 min) in which train cancellations could have occurred; in addition, delays longer than 20 min only accounted for 0.02% of all data; (3) deleting abnormal delay recoveries longer than the scheduled sectional buffer times. The data from five sample days is presented in Table 2.

Starting station represents the originating station of the train, *Current station* is the station at which the train has just arrived, and *Future station* represents the target station to be predicted. *Activaty* is the activity of this train. Where a *V* stands for a departure, a *D* stands for a passage, and an *A* stands for an arrival. A *K* can be added in front to indicate a short stop, which is an arrival and departure planned within the same minute. X_1–X_8 and y represent the independent and dependent variables, respectively.

In railway delay prediction, different attention is given to different types of delay severity in real-time dispatching. For example, as found in a previous study, dispatchers tended to focus mainly on delayed trains (>3 min in Netherlands Railways) to re-schedule the timetable, and delays of ≤3 min were usually not considered as delays [36]. Thus, the samples were divided into two groups in terms of the current delay (X_2), namely delays >3 min (*Longer than 3 min*) and delays ≤3 min (*Shorter than 3 min*), to establish their respective predictive models. It should be noted that delays >3 min indicates that the delay longer than 3 min ($(3, +\infty)$), while delays ≤3 min indicates that the delay is equal to or less than 3 min ($(-\infty, 3]$)

Moreover, the construction of separate models allows for the reduction of the training time and model prediction time for delayed trains, which requires instant training and prediction times.

In the original *RAS Competition*, the committee used the last four Tuesdays as the test dataset and required the participants to predict the delays of trains in the Netherlands Railways network at 08:20, 12:20, and 16:20 according to the respective states at 08:00, 12:00, and 16:00. However, the real delays at 08:20, 12:20, and 16:20 were not provided, and this data was not accessible to calculate the prediction accuracy of the test dataset in

Table 2. Dataset extracted for modelling.

Traffic date	Train number	Starting station	Current station	Future station	Activity	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	y
2017/9/28	4033	Utg	Mda	Rtd	*A*	0	1	86	20	87	121.5	24	8:00–9:00	0
2017/11/7	5171	Gvc	Sdm	Rlb	*K_V*	1	0	25	17	23	24.3	24.7	12:00–13:00	0
2017/10/16	1406	Rtd	Dvnk	Shl	*V*	0	2	40	22	42	39.4	19.7	15:00–16:00	2
2017/10/18	7668	Zp	Ahpr	Nml	*K_V*	0	4	23	24	27	9.9	16.3	7:00–8:00	1
2017/11/24	300,869	Amr	Ashd	Htnc	*D*	0	1	52	21	53	53	38.4	6:00–7:00	1

the present study. Therefore, the processed data was separated into training and test datasets; the training dataset contained days other than the last four Mondays, which were used as test data to validate the prediction accuracy. The sizes of the training and test datasets containing delays >3 min were 401,082 and 32,320, respectively, while the sizes of the training and test datasets containing delays ≤3 min were 516,6345 and 325,017, respectively. In both training datasets, 30% of the data was randomly selected as the validating dataset to train the hyperparameters of different algorithms.

Figure 2 presents the flowchart of the delay prediction process applied in the present study.

The original competition committee evaluated the final prediction accuracy by the following indicators: (1) whether there was a delay jump; (2) whether the delay decreased,

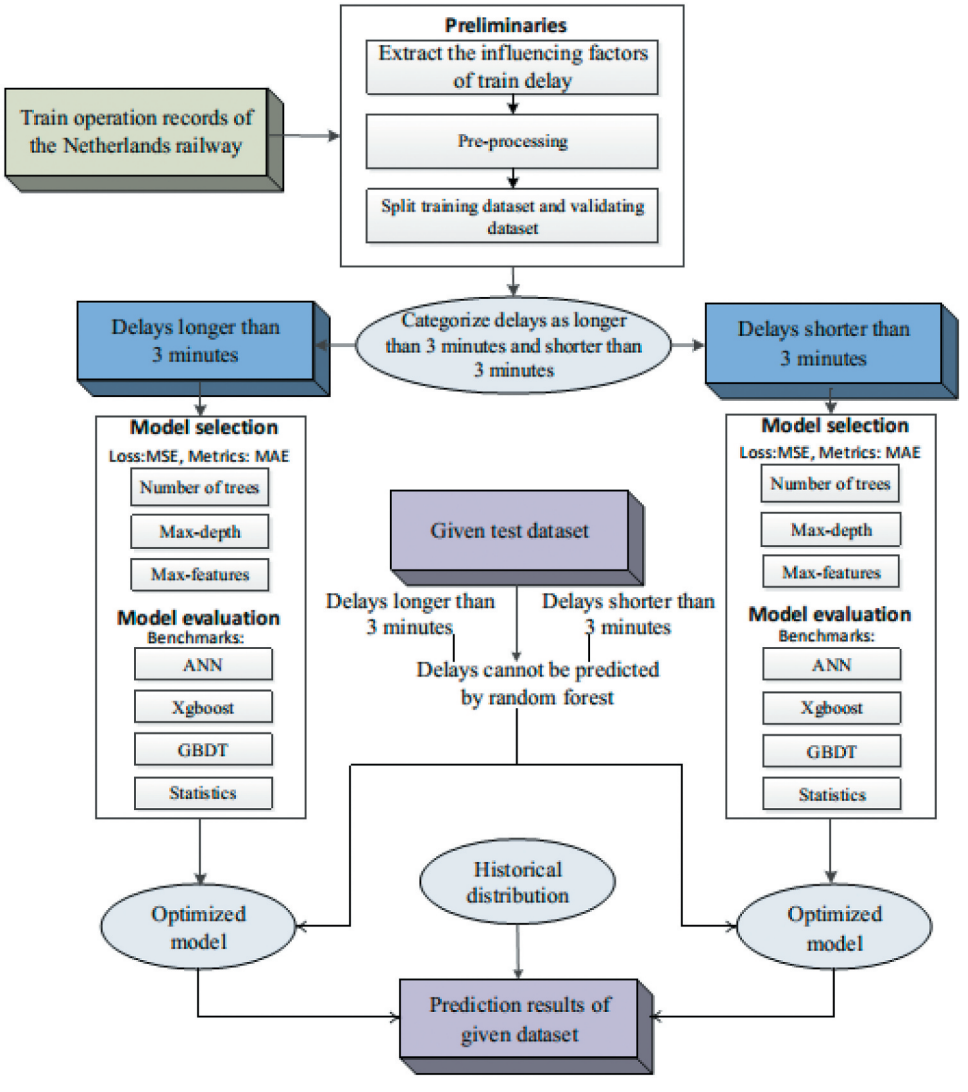


Figure 2. Delay prediction process flowchart.

increased, or stayed equal (delays within 1 min counted as equal; any delay change of greater than or equal to 2 min was considered a jump); (3) the delay prediction, rounded to the nearest minute. The final score was obtained by assigning different weights to these three indicators and then adding them together; the higher the score, the more precise the model.

In the present study, the scoring criteria were simplified and the precision of the prediction models was evaluated by calculating the mean absolute error (MAE) and root-mean-square error (RMSE) of different algorithms. The MAE and RMSE are respectively defined as Equations (1) and (2):

$$MAE = \frac{1}{N}\sum_{N}^{k=1}\left|\hat{y}_k - y_k\right|a \tag{1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{N}^{k=1}\left(\hat{y}_k - y_k\right)^2} \tag{2}$$

where y_k and $\hat{y}_k$ respectively represent the actual delay and predicted delay 20 min later, and were recorded in minutes.

The indicator *LESSTHAN i* was also used to evaluate the performance of the delay prediction accuracy, and is defined as Equation (3):

$$LESSTHAN\, i = \frac{N_d}{N_a} * 100\% \tag{3}$$

where N_d represents the sample size for which the absolute value of the difference between the actual and predicted values is less than or equal to i min, and N_a represents the total sample size.

The error between the actual and predicted delay was recorded in minutes, because the delays were recorded in minutes in the raw data.

4. Establishing the prediction model

4.1. Random forest regression model

RF is a machine learning method based on the bagging technique [37]. This modelling approach uses an ensemble of decision trees $\{h(X, \beta_k), k = 1, \ldots\}$ to map the relationship between vectors of the predictor and dependent variables. For each tree, X is the input vector and β_k is an independent stochastic variable that decides the growth of every tree. Each tree is a decision tree without pruning, and it is established according to the classification and regression tree (CART) principles [38] and the bootstrap sampling technique [39]. Given training dataset D_N (N is the sample size), according to the bootstrap sampling technique, n sub-samples are obtained by repetitive and independent sampling. Each sub-sample, which contains different features of the dataset, is used to establish a decision tree. Finally, the output of the model is the modes of the classes (classification) or the mean prediction (regression) of the individual trees, as shown in Figure 3.

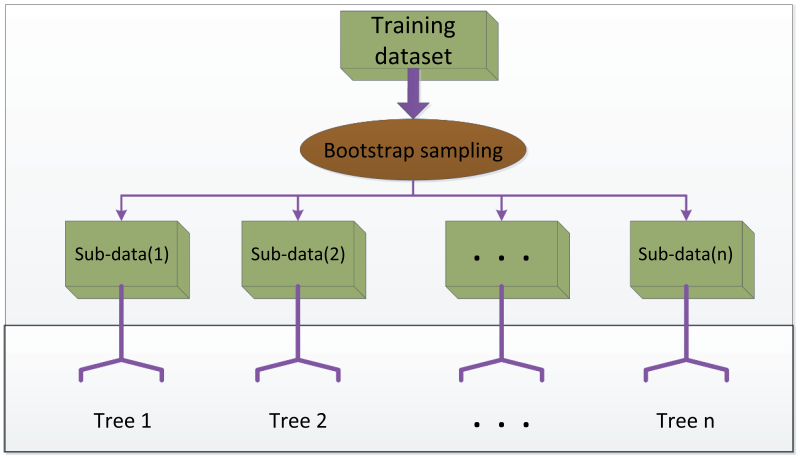


Figure 3. RF structure.

The accuracy of RF is mainly determined by the number of trees and the ability of each tree, which is in turn influenced by its complexity and variety. In this research, to obtain reasonable models, three key parameters in the Python Scikit-learn package were optimized, namely 'n_estimators,' 'max_depth,' and 'max_features,' which are defined as follows:

- n_estimators: the number of trees, which decides the scale of the forest. In general, more trees indicates stronger fitting/classifying abilities of the forest; however, the calculation time (cost) also notably increases;
- max_depth: the maximum number of generated nodes, which decides the complexity of each tree. More nodes in a tree indicates stronger fitting/classifying abilities. The training time increases with the number of nodes, and the model is more likely to be over-fitted;
- max_features: the maximum number of features chosen to split at the node. If the values for this parameter are too large or too small, the variety of the tree could be reduced.

In the present study, prediction models were established for both delays >3 min and ≤3 min, and the three key hyperparameters were optimized for the two different models. The mean squared error (MSE), as given by Equation (4), was selected as an RF loss function. The model performances on the validating dataset and the model training time are presented in Figures 4–6.

$$loss = \frac{1}{N}\sum_{N}^{k=1}\left(\hat{y}_k - y_k\right)^2 \tag{4}$$

Due to the large time requirement for model training, when optimizing n_estimators, the value was selected from five alternative values (100, 200, 300, 400, or 500). The RF models included eight input variables as a result of constructing each node of a decision tree. Thus, eight different scenarios were considered, each with a different number of

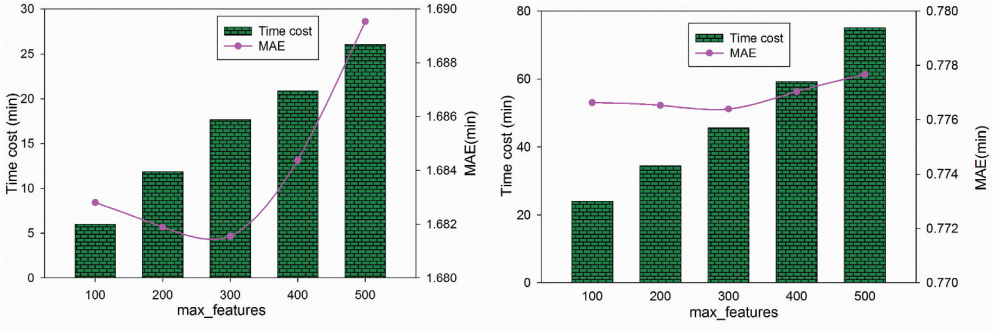


Figure 4. Optimal RF n_estimators parameter selection results: (a) >3 min and (b) ≤3 min.

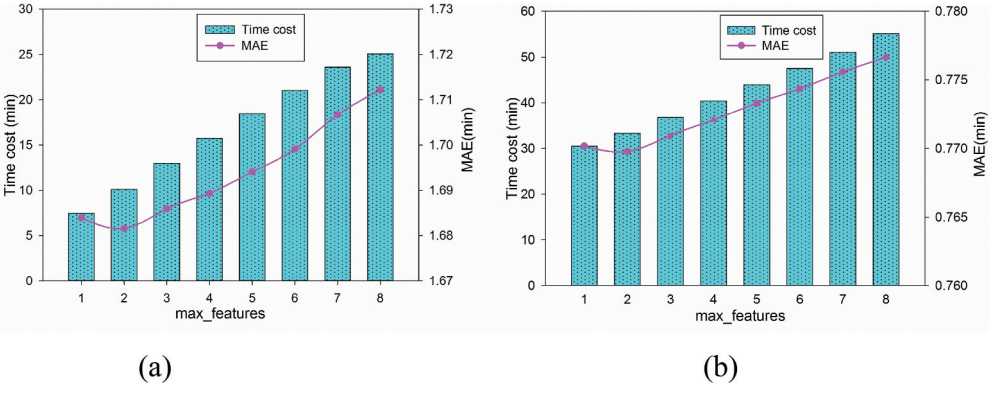


Figure 5. Optimal RF max_features parameter selection results: (a) >3 min and (b) ≤3 min.

variables. Theoretically, the max_depth parameter of the decision tree can be unlimited,

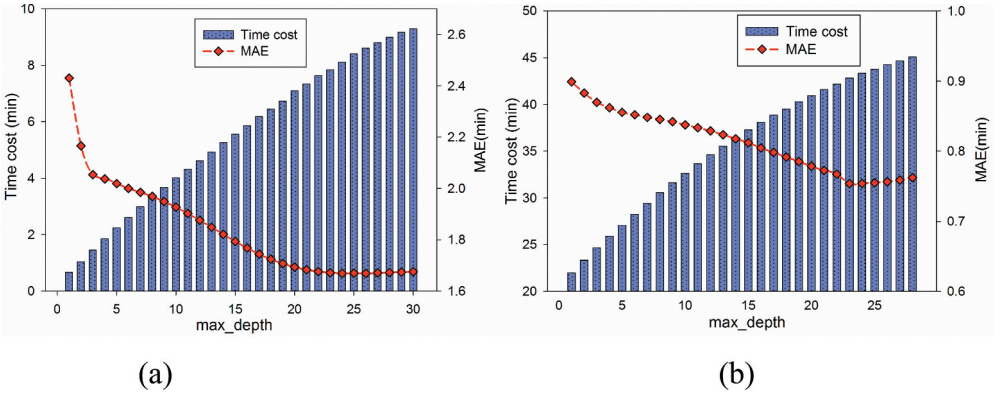


Figure 6. Optimal RF max_depth parameter selection results: (a) >3 min and (b) ≤3 min.

but a very large value for this parameter can lead to inefficiency and over-fitting. Therefore, the models were run with different max_depth values until the MAE stopped decreasing.

As can be seen in Figures 4–6, the model performances followed a tendency in which the error first decreased with the increase of the three key parameters, and then slightly increased with the further increase of the parameters. This phenomenon can be explained as follows. When the parameters were small, the forest structure and tree variety were simple; consequently, the data was under-fitted. When the parameters were too large, the forest structure and tree variety became too complicated; consequently, the data was over-fitted. Moreover, the time cost increased dramatically with the increase of model complexity. For these reasons, to construct reasonable prediction models and obtain a balance between the accuracy and computational speed, (n_estimators = 300, max_depth = 25, max_features = 2) and (n_estimators = 300, max_depth = 23, and max_features = 2) were ultimately chosen as the standard model architectures for delays >3 min and ≤3 min, respectively. In this research, the models were run on a device with a seventh-generation Intel Core i7 processor (four cores and eight threads).

4.2. Model evaluation

To evaluate the performance of the proposed model, four benchmark models were selected for comparison; these models included the artificial neural network (ANN), which is one of the most commonly used models in train delay prediction, and XGBOOST and GBDT, which are algorithms commonly used in the Kaggle data science competition platform, and have been widely used by data scientists to achieve state-of-the-art results in many machine learning challenges. The statistical method based on the average delay recoveries of each station was also selected. The respective operational principles of these baseline models are given as follows.

(1) In the ANN, the neurons are fully connected between adjacent layers, and information flows are transferred from the input layer to the output layer [40]. The loss function is obtained from a comparison between the fitted and observed values, and errors are back-propagated from the output to the input layers to optimize the weights and biases of each neuron [41].
(2) XGBOOST is an efficient learning algorithm based on the gradient boosting (GB) algorithm and CART. XGBOOST iteratively generates base learners according to first- and second-order derivatives, and then updates the learner by adding the base learners. The loss function of XGBOOST is regularized according to the CART leaf nodes [42].
(3) The gradient boosting decision tree (GBDT) [43,44] is an ensemble algorithm for data classification or regression that uses an addition model (the linear combination of the weak classifier) and the reduction of the residual generated by the training process. Through multiple iterations, GBDT produces a weak classifier in each iteration, and each classifier is trained based on the residual of the previous classifier.

(4) In the actual process of dispatching, that dispatchers predict the short-term delay according to historical experience is the easiest way, i.e., the average delay increase or future delay recovery when the train arrives at the station. Therefore, the average delay increases or delay recoveries after 20 min for each station were investigated. The ideal delay was then calculated according to these average statistics.

The MAE of the test dataset, the model training time, and the callback time are reported in Table 3, from which it is evident that the RF model outperformed the other benchmark algorithms on the test dataset; moreover, the RF time cost was reasonable. Although the training time for delays ≤3 min reached 47.95 min, the model callback cost was only 52.81 s. Prior to comparing the model performance, some baseline model parameters were also optimized, namely the optimizers, hidden layers, neurons in each layer, and ANN activation function.

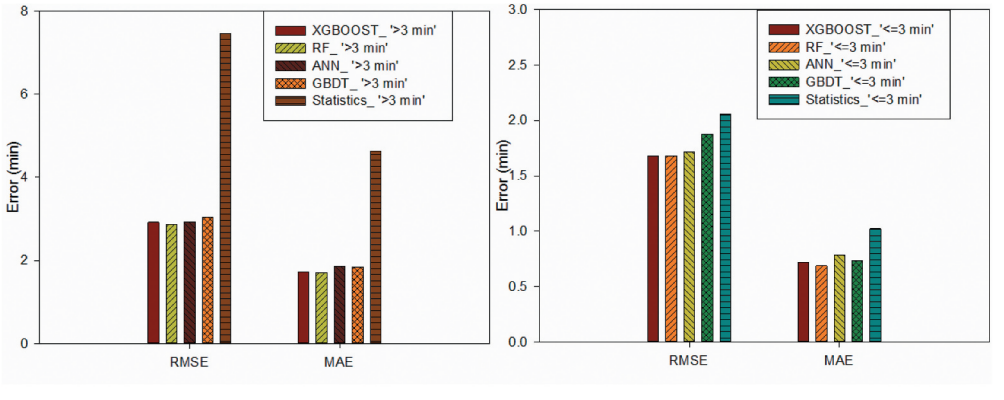
The MAE and RMSE of the models for different datasets are presented in Figure 7.

In addition, the evaluation indexes *LESSTHAN 1, LESSTHAN 2, LESSTHAN 3, LESSTHAN 4*, and *LESSTHAN 5* were used as validation criteria, as exhibited in Figure 8.

It can be seen from Figures 7 and 8 that the RF algorithm achieved the highest accuracy of all compared algorithms. The *LESSTHAN i* values reached high levels of

Table 3. Model performance comparison.

>3 min				≤3 min			
Model	MAE	Total time cost for model training (min)	Callback time (s)	Model	MAE	Total time cost for model loading (min)	Callback time (s)
RF	1.708	8.78	15.322	RF	0.687	47.95	52.81
ANN	1.857	46.75	3.687	ANN	0.788	331.75	6.351
XGBOOST	1.724	7.15	4.274	XGBOOST	0.723	19.58	14.185
GBDT	1.842	391.27	6.376	GBDT	0.732	647.75.	13.219
Statistics	4.631	<0.1	<0.1	Statistics	1.023	<0.1	<0.1



(a) (b)

Figure 7. MAE and RMSE results of all algorithms: (a) >3 min and (b) ≤3 min.

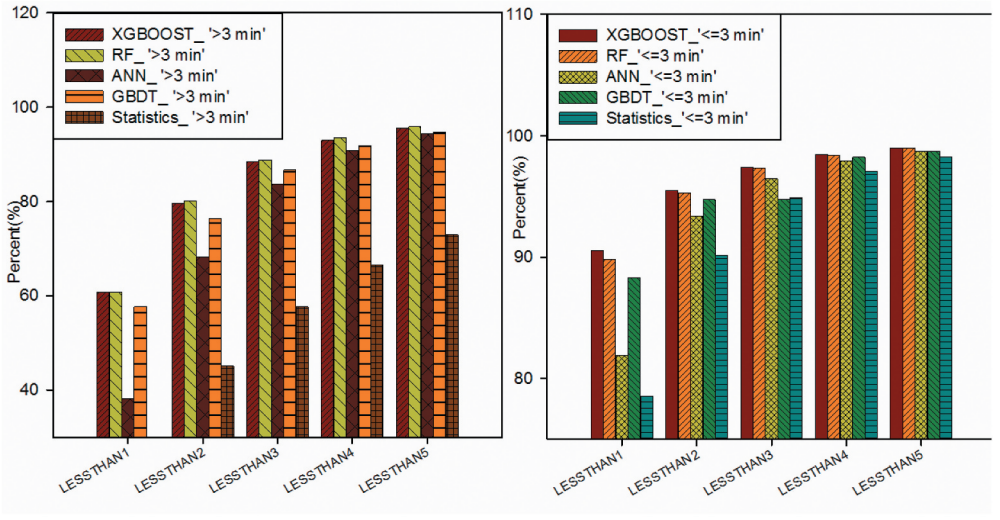


Figure 8. *LESSTHAN i* precision of all algorithms: (a) >3 min and (b) ≤3 min.

precision on both the >3 min and ≤3 min datasets, which indicates that the models had a good prediction effect. In addition, the importances of different features were also significant for the RF model, and were investigated using the Python Scikit-learn machine learning package [45]. The proportional importance was determined for each feature, and the results are reported in Table 4.

Table 3 reveals that there were differences among the relative feature importances for the >3 min and ≤3 min datasets. The X_2 feature had the highest importance for the >3 min dataset, and the next most important features were X_5, X_7, and X_8, i.e., the >3 min dataset was the most susceptible to the current delay severity and future operating conditions. The reason for this phenomenon is that when the delays are longer than 3 min, dispatchers may intend to take measures to recover from these delays, lest they influence more trains. The current delays are the basics, and the planned time and distance from now to 20 min later are the resources of delay recovery, which determine the maximum amount of time available for delay recovery. The period of delay

Table 4. The importances of features of the RF prediction model.

>3 min dataset		≤3 min dataset	
Feature name	Importance	Feature name	Importance
X_2 (Current delays)	0.395	X_8 (Period of delay occurrence)	0.352
X_5 (Planned time from now until 20 min later)	0.169	X_7 (Distance from now to 20 min later)	0.188
X_7 (Distance from now to 20 min later)	0.100	X_5 (Planned time from now until 20 min later)	0.098
X_8 (Period of delay occurrence)	0.098	X_6 (Distance travelled)	0.089
X_1 (Delays at the starting station)	0.078	X_4 (Actual travel time)	0.088
X_6 (Distance travelled)	0.061	X_2 (Current delays)	0.078
X_4 (Actual travel time)	0.055	X_3 (Planned time in the travelled section)	0.059
X_3 (Planned time in the travelled section)	0.045	X_1 (Delays at the starting station)	0.048

occurrence contains some hidden information, such as the number of trains running in this period and the interval of adjacent trains, etc., which may cause dispatchers to generate different adjustment strategies in different periods. This dataset was found to be not as sensitive to the past running status, because the past status may have little impact on the dispatcher's adjustment strategies. The hidden information that is supposed to be contained in a black box is represented by the delay value, as dispatchers are usually concerned more with the delay value than too much detailed information.

However, the ≤3 min dataset was found to be most vulnerable to the period of delay occurrence (X_8), followed by features X_7 and X_5. Because the ≤3 min dataset includes data with current delays equal to 0, this data may generate new delays 20 min in the future, while the new delays are dramatically influenced by the different periods within a single day. The data with a current delay of 0 while the delay 20 min later was not equal to 0 represented 21% of the total data, which may be the reason why the period of delay occurrence was the most important feature in the ≤3 min dataset. Additionally, like for the >3 min dataset, the future operation conditions (X_7, X_5) were also important for the ≤3 min dataset, while the variables related to past states were less crucial. It is worth noting that the current delay was found to slightly impact the delay 20 min later, which may be because the delay was so small that the current delays could be easily recovered from the future journey.

From the preceding analysis of the importances of the influencing factors of different datasets, the significant and not-so-vital factors were determined. Thus, only the first four crucial factors were considered to simplify the models to some extent and make them easier to train, and it was investigated whether the accuracies of the models were acceptable. The results are reported in Table 5.

Table 5 reveals that there were significant increases in all indicators for the >3 min dataset when all influencing factors were used as model inputs, while the increases in the indicators for the ≤3 min dataset were inconspicuous. *Thus, it is acceptable to use the first four critical factors to establish the delay prediction model for the ≤3 min dataset, whereas all influencing factors must be used for >3 min dataset.*

5. Conclusions

In relation to a current delay condition, the prediction of near-term delay using only actual records is a new challenge for railway operators. The accurate prediction of train delay propagation can provide useful information to train dispatchers, which can help them make more informed dispatching decisions. For this purpose, in the present study,

Table 5. Performance comparison of the simplified and entire models.

		LESSTHAN i (%)						
		1	2	3	4	5	MAE (min)	RMSE (min)
>3 min	Simplified	57.14	77.24	86.86	91.85	94.82	1.860	3.051
	Entire	60.80	80.12	88.82	93.50	95.81	1.708	2.863
	Change ratio	3.67	2.88	1.96	1.64	0.99	8.152%	6.150%
≤3 min	Simplified	89.24	95.12	97.32	98.36	98.98	0.720	1.707
	Entire	89.80	95.29	97.35	98.38	98.98	0.687	1.681
	Change ratio	0.57	0.18	0.03	0.02	0.00	4.603%	1.539%

the train delay 20 min later was predicted using the current circumstances based on real data provided by the *2018 RAS Problem Solving Competition: Train Delay Forecasting*. In this study, after analysing the delay propagation mechanisms and the data structure, various influencing factors were extracted and used as model inputs. Then, via optimized hyper-parameter selection, the RF algorithm was applied to establish the delay prediction model. The XGBOOST, ANN, GBDT, and statistical algorithms were used as benchmarks for comparison, and the results demonstrated that the RF technique had a good prediction effect. Finally, the importance of each input feature was calculated and analysed, and the results revealed that if the delay is longer than 3 min, the current delays are the most significant factors; in contrast, if the delay is shorter than 3 min, the period of delay occurrence is the greatest influencer of the delay 20 min later. Moreover, the future train operation conditions were found to play a vital role in both datasets, while the past running statuses were found to be less important. For a real-time train itinerary, because the input features can be known in real-time, the features of a delay can be input into this model once the delay occurs, and the delay propagation 20 min later can be output.

Acknowledgments

The authors are also grateful for the contributions made by our project partners.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Nature Science Foundation of China [Grant Nos. 71871188 and U1834209], the Science & Technology Department of Sichuan Province [Grant No. 2018JY0567], and the State Key Laboratory of Rail Traffic Control [Grant No. RCS2019K007].

ORCID

ZhongCan Li http://orcid.org/0000-0001-7123-5198

References

[1] Harris NG, Mjøsund CS, Haugland H. Improving railway performance in Norway. J Rail Transp Plann Manage. 2013;3(4):172–180.
[2] Wen C, Li Z, Lessan J, et al. Statistical investigation on train primary delay based on real records: evidence from Wuhan–Guangzhou HSR. Int J Rail Trans. 2017;5(3):170–189.
[3] Sciences TIfORatM. RAS problem solving competition: train delay forecasting. 2018. Available from: http://connect.informs.org/railway-applications/awards/problem-solving-competition/new-item2
[4] Nabian MA, Alemazkoor N, Meidani H. Predicting near-term train schedule performance and delay using bi-level random forests. Transp Res Rec. 2019;2673(5): 564–573.

[5] Haahr JT, Hellsten EO, van der Hurk E. Train delay prediction in the netherlands through neural networks. 2019.
[6] Milinković S, Marković M, Vesković S, et al. A fuzzy Petri net model to estimate train delays. Simul Modell Pract Theory. 2013;33:144–157.
[7] Wen C, Peng Q, Chen Y, et al. Modelling the running states of high-speed trains using triangular fuzzy number workflow nets. Proc Inst Mech Eng F J Rail Rapid Transit. 2014;228 (4):422–430.
[8] Hansen IA, Goverde RM, van der Meer DJ. Online train delay recognition and running time prediction. In: 13th International IEEE Conference on Intelligent Transportation Systems; 19–22 Sept 2010 Funchal, Portugal.
[9] Berger A, Gebhardt A, Müller-Hannemann M, et al.. Stochastic delay prediction in large train networks. In: 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany; 2011.
[10] Kecman P, Goverde RMP. Online data-driven adaptive prediction of train event times. IEEE Trans Intell Transp Syst. 2015;16(1):465–474.
[11] Goverde RMP. A delay propagation algorithm for large-scale railway traffic networks. Transp Res Part C Emerging Technol. 2010;18(3):269–287.
[12] Wen C, Huang P, Li Z, et al. Train dispatching management with data- driven approaches: a comprehensive review and appraisal. IEEE Access. 2019;7:114547–114571.
[13] Barta J, Rizzoli AE, Salani M, et al. Statistical modelling of delays in a rail freight transportation network. In: Proceedings of the 2012 Winter Simulation Conference (WSC); 9–12 Dec 2012; Berlin, Germany.
[14] Şahin İ. Markov chain model for delay distribution in train schedules: assessing the effectiveness of time allowances. J Rail Transp Plann Manage. 2017;7(3):101–113.
[15] Kecman P, Corman F, Meng L. Train delay evolution as a stochastic process. In: Proceedings of the 6th International Conference on Railway Operations Modelling and Analysis: RailTokyo2015; Tokyo, Japan; 2015.
[16] Zilko AA, Kurowicka D, Goverde RMP. Modeling railway disruption lengths with Copula Bayesian Networks. Transp Res Part C Emerging Technol. 2016;68:350–368.
[17] Lessan J, Fu L, Wen C. A hybrid Bayesian network model for predicting delays in train operations. Comput Ind Eng. 2019;127:1214–1222.
[18] Corman F, Kecman P. Stochastic prediction of train delays in real-time using Bayesian networks. Transp Res Part C Emerging Technol. 2018;95:599–615.
[19] Huang P, Lessan J, Wen C, et al. A Bayesian network model to predict the effects of interruptions on train operations. Transp Res Part C Emerging Technol. 2020;114:338–358.
[20] Kecman P, Goverde RMP. Predictive modelling of running and dwell times in railway traffic. Public Transp. 2015;7(3):295–319.
[21] Peters J, Emig B, Jung M, et al. Prediction of delays in public transportation using neural networks. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06); 28–30 Nov 2005; Vienna, Austria.
[22] Yaghini M, Khoshraftar MM, Seyedabadi M. Railway passenger train delay prediction via neural network model. J Adv Transp. 2013;47(3):355–368.
[23] Oneto L, Fumeo E, Clerico G, et al. Train delay prediction systems: a big data analytics perspective. Big Data Res. 2018;11:54–64.
[24] Oneto L, Fumeo E, Clerico G, et al. Dynamic delay predictions for large-scale railway networks: deep and shallow extreme learning machines tuned via thresholdout. IEEE Trans Syst Man Cybern Syst. 2017;47(10):2754–2767.
[25] Huang P, Wen C, Fu L, et al. A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems. Inf Sci. 2020;516:234–253.
[26] Wen C, Mou W, Huang P, et al. A predictive model of train delays on a railway line. J Forecasting. 2020;39(3):470–488.

[27] Marković N, Milinković S, Tikhonov KS, et al. Analyzing passenger train arrival delays with support vector regression. Transp Res Part C Emerging Technol. 2015;56:251–262.
[28] Barbour W, Mori JCM, Kuppa S, et al. Prediction of arrival times of freight traffic on US railroads using support vector regression. Transp Res Part C Emerging Technol. 2018;93:211–227.
[29] Huang P, Wen C, Fu L, et al. A hybrid model to improve the train running time prediction ability during high-speed railway disruptions. Saf Sci. 2020;122:104510.
[30] Pongnumkul S, Pechprasarn T, Kunaseth N, et al. Improving arrival time prediction of Thailand's passenger trains using historical travel times. In:. 2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE); 14–16 May 2014; Chon Buri, Thailand
[31] Li D, Daamen W, Goverde RMP. Estimation of train dwell time at short stops based on track occupation event data: A study at a Dutch railway station. J Adv Transp. 2016;50 (5):877–896.
[32] Lee W-H, Yen L-H, Chou C-M. A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services. Transp Res Part C Emerging Technol. 2016;73:49–64.
[33] Jiang C, Huang P, Lessan J, et al. Forecasting primary delay recovery of high-speed railway using multiple linear regression, supporting vector machine, artificial neural network, and random forest regression. Can J Civil Eng. 2019;46(5):353–363.
[34] Lulli A, Oneto L, Canepa R, et al. Large-scale railway networks train movements: a dynamic, interpretable, and robust hybrid data analytics system. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA) Turin; 1–3 Oct 2018; Italy, Italy.
[35] Gaurav R, Srivastava B. Estimating train delays in a large rail network using a Zero Shot Markov Model. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC); 4–7 Nov 2018; Maui, HI, USA.
[36] Yuan J, Goverde R, Hansen I. Propagation of train delays in stations. WIT Trans Built Environ. 2002;61(10):975–984.
[37] Breiman L. Bagging predictors. Mach Learn. 1996;24(2):123–140.
[38] Breiman L, Friedman J, Olshen R, et al. Classification and regression trees. 1984.
[39] Felsenstein J. Confidence limits on phylogenies: an approach using the bootstrap. Evolution. 1985;39(4):783–791.
[40] Svozil D, Kvasnicka V, Pospichal J. Introduction to multi-layer feed-forward neural networks. Chemometr Intell Lab Syst. 1997;39(1):43–62.
[41] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323(6088):533.
[42] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug; San Francisco California, USA; p. 785–794.
[43] Mason L, Baxter J, Bartlett PL, et al. Boosting algorithms as gradient descent. In: Advances in neural information processing systems; 2000; Cambridge, England; p. 512–518.
[44] Li C. A gentle introduction to gradient boosting. 2016. Available from: http://wwwccsneuedu/home/vip/teach/MLcourse/4_boosting/slides/gradient_boostingpdf
[45] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–2830.