Transportation research part E: logistics and transportation review, 2020, 141: 102022.
https://doi.org/10.1016/j.tre.2020.102022

Modeling train operation as sequences: A study of delay prediction with operation and weather data

Ping Huang [a,b], Chao Wen [a,b,c*], Liping Fu [b], Javad Lessan [b], Chaozhe Jiang [a *], Qiyuan Peng [a], and Xinyue Xu [c]

[a] National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Southwest Jiaotong University, Chengdu, 610031, China;

[b] High-speed Railway Research Centre, University of Waterloo Waterloo, N2L3G1, Canada;

[c] State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China, 100044.

***Corresponding at: wenchao@swjtu.cn; jiangchaozhe@163.com; ping.huang@swjtu.edu.cn**

Abstract

This paper presents a carefully designed train delay prediction model, called FCLL-Net, which combines a fully-connected neural network (FCNN) and two long short-term memory (LSTM) components, to capture operational interactions. The performance of FCLL-Net is tested using data from two high speed railway lines in China. The results show that FCLL-Net has significantly improved prediction performance, over 9.4% on both lines, in terms of the selected absolute and relative metrics compared to the commonly used state-of-the-art models. Additionally, the sensitivity analysis demonstrates that interactions of train operations and weather-related features are of great significance to consider in delay prediction models.

Keywords: Train operation; sequences; delay prediction; deep learning; interactions

1. Introduction

Despite the availability of advanced communication and control technologies, delays in train operations are inevitable due to unexpected disruptions, such as poor weather, power outages, and facility failures (Khadilkar, 2016). Once such disruptions occur, train dispatchers must assess the severity in the form of predicting the expected delays to the affected trains, and reduce the impacts by adjusting the timetable to minimize the overall service disturbances. Therefore, the accurate prediction of train delays is necessary for effective train dispatching and service management, and for improved passenger information provision.

The problem of train delay prediction has been studied extensively in the literature (Cacchiani et al., 2014; Wen et al., 2019). A handful number of models have been proposed, with approaches ranging from statistical methods, e.g., linear regression and probability density models (Lessan et al., 2018; Yuan et al., 2002), to machine learning approaches, e.g., support vector machines and neural networks (Malavasi and Ricci, 2001; Marković et al., 2015; Yaghini et al., 2013). However, traditional train delay prediction models usually have shortcomings in extracting the knowledge hidden in train operation data for the following main issues.

1) Diversity in the factors that contribute to train delay: these influencing factors have multiple attributes, including operational factors (i.e., timetable-related factors) and non-operational factors (i.e., equipment- and weather-related factors).

2) Lack of a feedback characteristic in the conventional delay prediction model: Existing delay prediction models do not take into account sequence characteristics. Instead, all input is treated as static features; as a result, important information hidden in the operational data is neglected.

3) Interdependency: train events (i.e., arrival, passage, and departure) are interdependent due to using common interlocking and dispatching resources. The interdependency factor plays a major role in train delay prediction, however, existing studies fail to take into consideration the collective

information regarding interactions of train events.

Recent advancements in artificial intelligence (AI) have provided new opportunities to address many limitations of the traditional models. For example, deep learning techniques can be applied to process and utilize high-dimensional, non-linear sequence/time-series data (Najafabadi et al., 2015). These techniques have already been successfully applied to address numerous challenging problems, such as the use of long short-term memory (LSTM) in sequence learning, e.g., speech recognition and language modeling. For example, consider a language model trying to predict the next word based on the previous words using LSTM (Olah, 2015). To predict the last word in "I grew up in France… I speak fluent French", recent information, i.e., the word "fluent", suggests that the next word is probably a noun, and, more accurately, the name of a language. However, to narrow down which language, the context of "France" from an earlier stage in the process is required. The LSTM recognizes the relationships of the to-be-predicted word with the recent and earlier stage information to predict the language name. This is precisely why LSTM has achieved great success in language modeling (Graves et al., 2013), and it has motivated the authors of the present study to utilize the ability of LSTM to capture important information regarding interaction among train events in delay prediction. In a typical railway system, the interlocking of equipment, train overtaking, and other resource limitations cause the operation of a train to be necessarily affected by one or more preceding operations. Moreover, the state of a train at a specific station is dependent on its states at previous stations. In this paper, train operations are treated as sequences, and the ability of LSTM in sequence learning is used to capture the interactions between trains and adjacent stations to more accurately predict upcoming train delays. Considering the interactions between trains and stations, and influencing operational and non-operational factors, the proposed model integrates two LSTM components with a fully-connected neural network (FCNN) component to separately process the operational and non-operational features. Therefore, proposed hybrid architecture (FCLL-Net) possess two unique features. First, in contrast to the traditional train delay predictors that treat all factors as static features, which results in the loss of information embedded in the operational data, the proposed model combines two different types of neural network models: one for operational features and the other for non-operational features. Second, it represents interactions and delay propagation patterns between trains and stations, as the LSTM units in the proposed model have a feedback mechanism to capture the subtle correlations between the operational features of a train group. The main contribution of this study lies in proposing a delay prediction model that can capture the interactions between trains and stations, which are two critical factors for explaining and describing the train delay propagation between trains and between stations. To the best of the authors' knowledge, this is the first study that simultaneously considers these dependencies in train operation for delay prediction purpose.

The remainder of this paper is organized as follows. In Section 2, the related work on train delay prediction is reviewed, and the real-time delay prediction problem is described. In Section 3, the architecture of FCLL-Net and the mechanism underlying FCNN and LSTM are introduced. In Section 4, the train operation data and the model selection experiments are described. In section 5, the prediction capability of the FCLL-Net model and the generalization of its application, as well as its sensitivity to model components, are reported. Finally, the work is concluded in Section 6.

2. Literature review and problem statement

2.1. Literature review

Train delay prediction has been an active area of research in train dispatching for decades. Most recently, the topic of the INFORMS 2018 Railroad Problem Solving Competition was set as

“Predicting Near-Term Train Schedule Performance and Delay” based on operational records.

In general, there are two types of approaches to train delay propagation modeling: event-driven methods and data-driven methods. Event-driven methods estimate train delays by fitting models to train events (arrival, passage, and departure) using interlocking rules, such as track and signal use sequences, whereas data-driven methods reveal the distributional characteristics of data or extract principles from input data, and then use these for prediction via the application of tools such as probability density models, regression models, and artificial intelligence (AI) models.

As train operations are composed of arrival, passage, and departure events, graph models whose nodes can be interpreted by the train events are widely-used methods. The proposed graph-based models for train delay prediction or propagation include time event graphs (Goverde, 2010; Hansen et al., 2010; Kecman and Goverde, 2015a), activity graphs (Büker and Seybold, 2012), and alternative graphs (Corman et al., 2014). (Berger et al., 2011) proposed a stochastic graph model for forecasting train arrival and departure times while considering the passenger waiting policies, train running time, train dwelling times, and buffer times. This model was implemented on a German timetable, and the results showed that the proposed model was sufficient to be applied in practice, but that the accuracy of the model required further improvement. (Carey and Kwieciński, 1994) proposed simple stochastic approximations to derive knock-on train delays and simulate the interactions between trains, from which they developed and assessed a simulation model to predict the probability distributions of knock-on delays at stations (Carey and Carville, 2000). (Yuan and Hansen, 2007) proposed an analytical stochastic model of train delay propagation between stations that could realistically estimate the knock-on delays of trains caused by route conflicts and late transfer connections. Another type of event-driven model for train operation is network-based models, such as Bayesian network (BN) models and the Markov model (MM). Recently, there have been two types of BN structures proposed for train delay prediction. One used the train events of every single train as the nodes of the BN model to predict HSR delays using the operational records of the Chinese HSR, and obtained 80% accuracy in prediction over a 60-min horizon (Lessan et al., 2019). The other used the events of two consecutive trains as the nodes of the BN model to predict the train delay in the Sweden network (Corman and Kecman, 2018). These two studies presented improvements over conventional prediction approaches based solely on the fixed values obtained offline from historical data. In addition to these two BN-based studies, other BN studies used in railway system include a non-parametric BN model for the prediction of disruption length to assess factors influencing the lengths of disruptions (Zilko et al., 2016), and a carefully designed BN structure to predict the effects of interruptions on train operation (Huang et al., 2020a). MM is also widely used for train delay prediction (Al-Ibrahim, 2010; Barta et al., 2012; Gaurav and Srivastava, 2018; Kecman et al., 2015), as the train operation at stations can be treated as typically discrete processes. However, the performance of MM is usually worse than that of the BN model because the MM treats train operation as only a chain, rather than a network. (Şahin, 2017) generated a transition matrix of state-to-state transition probabilities in an MM from actual records of train movements; however, the data used for modeling, which comprised 6-h, 18-station train graphs covering seven days (July 14-20, 2002), employed only six delay classes for distinguishing delay states. A fuzzy Petri net (FPN) model for estimating train delays, in which experts’ knowledge was used to define fuzzy sets and rules to transform expertise into a model to calculate train delays, was proposed by (Wang and Ma, 2013). An adaptive network fuzzy inference system (ANFIS) model based on historical data on train delays in the Serbian Railways system was set up by (Milinković et al., 2013). An HSR running state model based on triangular fuzzy number workflow nets of fuzzy train activity times was generated using data taken from five stations between Beijing South and

Dezhou East in the Beijing-Shanghai HSR system from June 21-24, 2012 (Wen et al., 2014). However, all the graph- and network-based methods are built on the Markov property assumption, which means that the train operation is only related to its latest state. Although this simplification can enable reasoning and computations that would otherwise be intractable, it oversimplifies train operations and neglects many important influencing factors of delays, such as timetable-, weather-, and infrastructure-related features.

Data-driven methods use unsupervised learning (e.g., probability distribution and clustering) for train operation knowledge discovery, and supervised learning (e.g., regression models and machine learning models) for train event times prediction. Due to unexpected disturbances, complex systems such as rail systems seem to exhibit heavy-tailed distributional forms (Lessan et al., 2018). Common heavy-tailed probability distribution models, such as exponential distribution (Briggs and Beck, 2007; Harris, 2006; Huisman and Boucherie, 2001), Weibull distribution (Goverde et al., 2013; Goverde et al., 2001), and log-normal distribution (Huang et al., 2019; Yang et al., 2019), are three of the widely used probability density models for railway systems. Other unsupervised learning methods used for train operation data include association analysis (Wallander and Mäkitalo, 2012) and the K-means clustering model (Cerreto et al., 2018) for delay pattern discovery. (Van der Meer et al., 2009) focused on the statistical analysis of train running times among stations to improve the prediction accuracy of delay propagation in railway systems. The analyses revealed a strong correlation between arrival delays and dwell times, whereas the correlation between running times and departure delays was found to be much weaker. (Meester and Muns, 2007) used a phase-type distribution model to evaluate secondary delay distribution according to the primary delay distribution. (Cerreto et al., 2016) investigated the distributions of the actual running times of trains in sections. (Lessan et al., 2018) developed a statistical method to estimate the arrival delay distribution according to the previous departure delay and running time distributions.

Supervised learning models are usually used for prediction tasks. Such methods include regression models and AI methods. Regression models have been used in train operation modeling to predict train delays (Murali et al., 2010), train dwelling times (Li et al., 2016), and the probability of the number of delayed trains (Wen et al., 2017). Further, (Gorman, 2009) estimated a linear regression model to predict the total running time of freight trains in the US. (Kecman and Goverde, 2015b) compared the predictive accuracy of the least-trimmed squares (LTS) robust linear regression model and the decision tree model in train running time and dwelling time prediction, and demonstrated that the local LTS model outperformed others in terms of computational time and prediction accuracy. AI models learn information from input data and map it to outputs to make predictions. Such models can be regarded as a black box; they are more powerful in fitting nonlinear and high-dimensional data, but are less straightforwardly interpretable than statistical models. As noted by (Wallander and Mäkitalo, 2012), data mining approaches can be used to obtain a better understanding of train delay concatenation, and are useful tools in developing more robust timetables and more powerful support systems for real-time dispatching. The most widely used conventional AI approaches in train delay prediction are fully-connected neural networks, or called multilayer perceptron (MLP) in the previous studies (Haahr et al., 2019; Malavasi and Ricci, 2001; Yaghini et al., 2013), although such networks have been outperformed in train delay prediction by support vector regression (SVR) systems (Marković et al., 2015). The authors compared the performances of SVR and MLP on all the testing datasets (including normal trains and delayed trains), but the main purpose of a predictive model is to predict the status of delayed trains. Recently, an SVR was applied to the arrival times prediction of freight trains in the US Railway system (Barbour et al., 2018), and the authors attentively investigated the impact of model parameters and input data on model performance. Other AI applications for train operations include the use of deep extreme learning machines (DELMs) (Oneto et al., 2017, 2018) and decision trees (Jiang et al., 2018; Nabian et al., 2019; Nair et al., 2019) for real-time delay prediction, the SVR for train position (Chen et al., 2015), the k-nearest neighbor (KNN) for dwell times estimation (Li et al., 2016), and a combination of SVR and a filtering technique for running time prediction during disruptions (Huang et al., 2020)b. A recent study applied a hybrid deep learning model for train delay prediction based on operation data (Huang et al., 2020c), in which the proposed model showed satisfactory performance comparing against conventional AI models. AI approaches are superior to statistical methods in handling train operation data due to their reliance on fewer internal and mathematical assumptions regarding railway operation. However, although all AI applications in train delay estimation of which the authors are aware require data on all individual trains as input, interactions in train groups are never taken into account. Furthermore, as no existing AI model uses a sequence-based approach, none is capable of addressing heterogeneous influencing factors. These shortcomings are the drawbacks of train delay prediction models, as delays are influenced by both operational and non-operational features, and it is common for interactions to occur as a result of the interlocking and continuity of train operations. It is therefore important to take into account the interactions in train groups via the application of suitable techniques in conjunction with historical train operation data.

Although delay prediction is an active research area and is of particular importance concerning issues of delay recovery and propagation, a review of the literature reveals that it is difficult to find specific railway traffic control system models that can handle dynamics in real time (Cacchiani et al., 2014; Ghofrani et al., 2018; Wen et al., 2019). Motivated by the successful applications of LSTM in language modeling and the drawbacks of existing train delay prediction models, the authors of the present study decided to establish a train delay prediction model based on LSTM that can treat input factors as sequences for the potential consideration of interactions of train operations. The primary advantages of the present work include: 1) the ability of the proposed model to feed operational and non-operational factors into corresponding units to efficiently recognize their respective influences, and 2) the use of adjacent trains as a group to predict individual train delays, which can be regarded as sequences, to uncover cumulative interactions within the train operation data.

2.2. Problem statement

In railway systems, consider the train operation on a general high-speed railway (HSR) line including N stations, as shown in the time-space diagram in Fig. 1 (a) and 1(b). The stations are sequentially labeled as S_1, S_2, …, S_N, with S_1 representing the originating station and S_N the terminal station. A timetable is designated for the HSR line, which specifies the scheduled arrival and departure times at individual stations. Consider a particular train that has just arrived at station S_P, $S_P \in \{S_1, S_2, .., S_N\}$, at the prediction time (now); the problem is to predict the expected arrival time of this train of interest at the downstream stations, i.e., S_{P+1}, S_{P+2}, ..., S_N. Due to the interlock of train routes, train overtaking, and dispatching operations (e.g., changing train order, recovering from delays), the trains that precede the current train may have an effect on the operation of the current train, and the train operations at the previous stations may have an effect on the delays at the to-be-predicted station. Therefore, such information should be taken into account in a delay prediction model with the support of sophisticated AI techniques and computational ability. Consider H consecutive trains operating from S_1 to S_N, labelled sequentially as I-H+1, I-H+2,…, I according to their actual operating order, where train I is the current train of interest. The prediction target is the expected delay of train I at station S_{P+1}, denoted by $y_{I,\ P+1}$ for one-step prediction, or the delay of train I at station S_{P+Q}, denoted by $y_{I,P+Q}$ for Q-step prediction ($Q > 1$). The information of H trains from the previous Z stations and sections are used as the inputs of the proposed model. The inputs of the prediction process include the infrastructure-related information, the weather-related information, and the timetable-related (operational) information of individual trains, which are defined as follows.

1). Infrastructure-related variables:

- L is a vector representing the length of the individual sections between stations. $L=\{l_{P-Z+1},...,l_{P-1},l_P\}$, where l_n is the section length between S_n and S_{n+1}.
- M is a vector representing the number of tracks at individual stations. $M=\{m_{P-Z+2},...,m_P,m_{P+1}\}$, where m_n is the number of tracks at station S_n.

2). Weather-related variables:

- C_i is a vector representing the temperature of train i suffered in the individual sections (between stations). $C_i=\{c_{i,P\text{-}Z+1},...,c_{i,P-1},c_{i,P}\}$, where $c_{i,n}$ is the temperature of train i suffered between S_n and S_{n+1} (i=I-H+1, I-H+2, …, I).
- V_i is a vector representing the wind speed of train i suffered in the individual sections (between stations). $V_i=\{v_{i,P\text{-}Z+1},...,v_{i,P-1},v_{i,P}\}$, where $v_{i,n}$ is the wind speed of train i suffered between S_n and S_{n+1} (i=I-H+1, I-H+2, …, I).
- A_i is a vector representing the rainfall of the nearest hour of train i suffered in the individual sections (between stations). $A_i=\{a_{i,P\text{-}Z+1},...,a_{i,P-1},a_{i,P}\}$, where $a_{i,n}$ is the rainfall of train i suffered between S_n and S_{n+1} (i=I-H+1, I-H+2, …, I).

3). Operational or timetable-related variables:

- T_i is a vector representing the actual running times of train i at the individual sections between stations. $T_i=\{t_{i,P\text{-}Z+1},...,t_{i,P-1},t_{i,P}\}$, where $t_{i,n}$ is the actual running time of train i between S_{n-1} and S_n (i=I-H+1, I-H+2, …, I).
- W_i is a vector representing the actual dwell times of train i at individual stations. $W_i=\{w_{i,P\text{-}Z+1},...,w_{i,P-1},w_{i,P}\}$, where $w_{i,n}$ is the scheduled dwell time of train i at station S_n (i=I-H+1, I-H+2, …, I).
- R_i is a vector representing the actual time interval of train i and train i-1 at individual stations. $R_i=\{r_{i,P\text{-}Z+1},...,r_{i,P\text{-}1},r_{i,P}\}$, where $r_{i,n}$ is the actual time interval between train i and i-1 at station S_n (i=I-H+1, I-H+2, …, I).
- D_i is a vector representing the departure delay that train i incurred at individual stations. $D_i=\{d_{i,P\text{-}Z+1},...,d_{i,P-1},d_{i,P}\}$, where $d_{i,n}$ is the delay of train i at station S_n (i=I-H+1, I-H+2, …, I).
- Y_i is a vector representing the arrival delay that train i incurred at individual stations. $Y_i=\{y_{i,P\text{-}Z+1},...,y_{i,P-1},y_{i,P}\}$, where $y_{i,n}$ is the delay of train i at station S_n (i=I-H+1, I-H+2, …, I).
- T'_I is a vector representing the scheduled running times of train i at the individual sections between stations. $T'_i=\{t'_{i,P-Z+1},...,t'_{i,P-1},t'_{i,P}\}$, where $t'_{i,n}$ is the scheduled running time

of train *i* between S_n and S_{n+1} (*i*=*I*-*H*+1, *I*-*H*+2, …, *I*).

- *W'$_I$* is a vector representing the scheduled dwell times of train *i* at individual stations. $W_i^{'}=\left\{w_{i,P-Z+1}^{'},\ldots,w_{i,P-1}^{'},w_{i,P}^{'}\right\}$, where $w'_{i,n}$ is the scheduled dwell time of train *i* at station S_n (*i*=*I*-*H*+1, *I*-*H*+2, …, *I*).
- *R'$_I$* is a vector representing the scheduled time interval of train *i* and train *i*+1 at individual stations. $R_i^{'}=\left\{r_{i,P-Z+1}^{'},\ldots,r_{i,P-1}^{'},r_{i,P}^{'}\right\}$, where $r'_{i,n}$ is the time interval between train *i* and *i*+1 at station S_n (*i*=*I*-*H*+1, *I*-*H*+2, …, *I*).
- *S'$_I$* is a vector representing the scheduled stops of train *i* at each pair of adjacent stations. $S_i^{'}=\left\{s_{i,P\text{-}Z+1}^{'},\ldots,s_{i,P-1}^{'},s_{i,P}^{'}\right\}$, where $s'_{i,n}$ is the number of stops at S_n and S_{n+1} (*i*=*I*-*H*+1, *I*-*H*+2, …, *I*).

Because train running times are different between sections, and because stopping schemes are different between trains, in *Q*-step prediction, the scheduled running times, scheduled dwelling times, and stop information from the current station to the target station should be considered in the model to indicate the supplemental time volume of each train in the section. For example, Fig. 1(c) shows a two-step prediction problem in which the red lines represent the current time, station S_P is the current station, and the prediction of the delays at S_{P+2} is of interest. Train 1 arrives at station S_{P+2} from S_P without stopping at S_{P+1}, whereas Train 2 arrives at station S_{P+2} from S_P having stopped at S_{P+1}. According to the definitions of the above factors, vectors *T'$_I$, W'$_i$*, and *S'$_I$* only include the train operation information from station $S_{P\text{-}Z+1}$ to S_P. If these two trains are allowed to have the same maximum speed, and if they have the same scheduled travel time from S_P to S_{P+2}, ignoring the information from S_P to S_{P+2} will lead to the incapability of the model to recognize the greater potential delay recovery of Train 1, thus impeding the performance of the model. Therefore, in *Q*-step prediction, these three vectors are modified to include the scheduled travel time, scheduled dwelling times, and the number of stops information from the current station to the target station. In detail, the first *Z*-1 elements of *T'$_I$*, *W'$_I$*, and *S'$_i$* are the same as those in one-step prediction, but the last element of these vectors is the scheduled running time from S_P to S_{P+2}, the scheduled dwelling time at S_P and S_{P+1}, and the number of stops at S_P, S_{P+1}, and S_{P+2}, respectively, which means that the last element of these two vectors in Q-step prediction always includes the train operation information in the section S_P-S_{P+Q}. In addition, for vector *M*, the last element is the number of tracks at S_{P+Q}.

Finally, to simplify notation, two new vector variables are introduced to combine the individual vectors presented previously: $OF=\left\{T_i,W_i,R_i,D_i,Y_i,T_i^{'},W_i^{'},R_i^{'},S_i^{'}\right\}$ for operational (timetable-related) factors, and $NF=\left\{L,M,C_i,V_i,F_i\right\}$ for non-operational (infrastructure- and weather-related) factors.

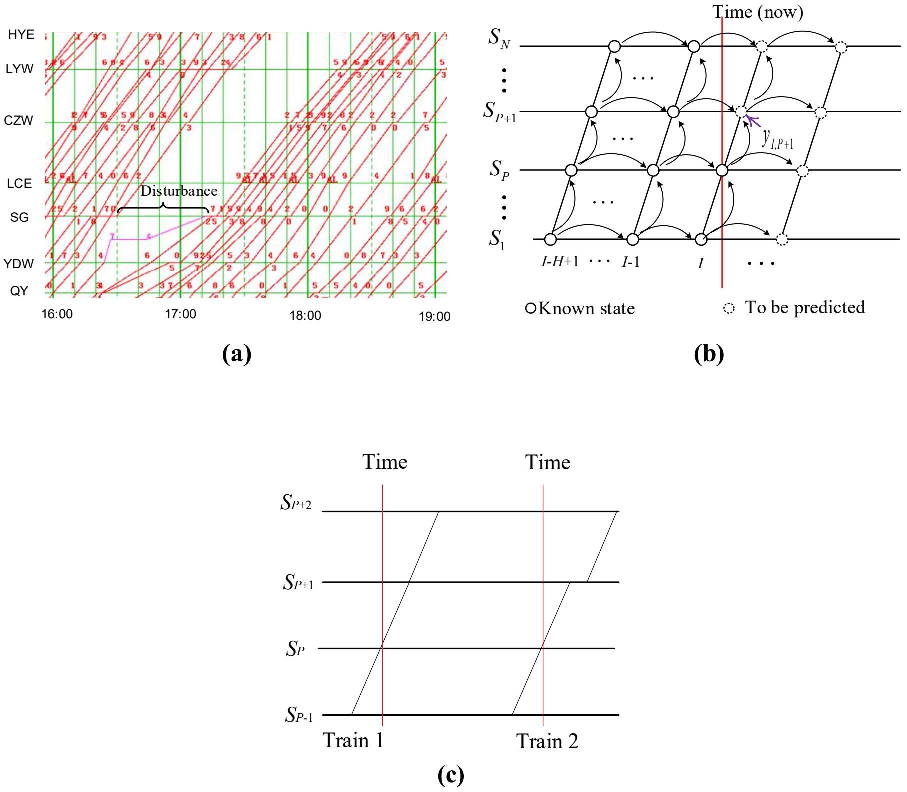


Fig. 1. Time-space diagram: (a) a train operation example with a service disturbance; (b) a general description of the real-time train delay prediction problem; (c) two train operation cases in two-step prediction.

3. The proposed deep learning model (FCLL-Net)

The proposed delay prediction model (FCLL-Net) is a hybrid of two types of popular neural network models, namely fully-connected neural network (FCNN) and long short-term memory (LSTM), as briefly introduced in the following section.

3.1. Architecture of FCLL-Net

As described previously, train delays are affected by both operational and non-operational factors. It is inefficient to feed non-operational factors into LSTM because of the following: 1) the infrastructure-related factors are static features that do not convey any interactions between trains or stations; 2) though the weather-related factors are time-series factors, they are unrelated to train operations, and fail to convey train operation information; 3) feeding these factors into LSTM will increase the model training time; for the same data, training LSTM usually costs more time than training the simpler FCNN model, even though LSTM and FCNN have the same layers and neurons. To effectively handle these two types of factors, the authors propose a hybrid of two neural network models: long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997), which has found widespread success in predictive tasks using sequence data, and fully-connected neural networks (FCNN), which are effective in modelling cross-sectional data. To simultaneously capture the interactions between trains and between stations, two LSTM components are used in the proposed model: one for capturing the interactions between trains, and the other for capturing the interactions between stations. Fig. 2 presents the architecture of the proposed model, in which the FCNN component could have multiple hidden layers with a varying number of neurons, while both

LSTM components could include multiple layers of basic LSTM units stacked onto each other for improved learning power.

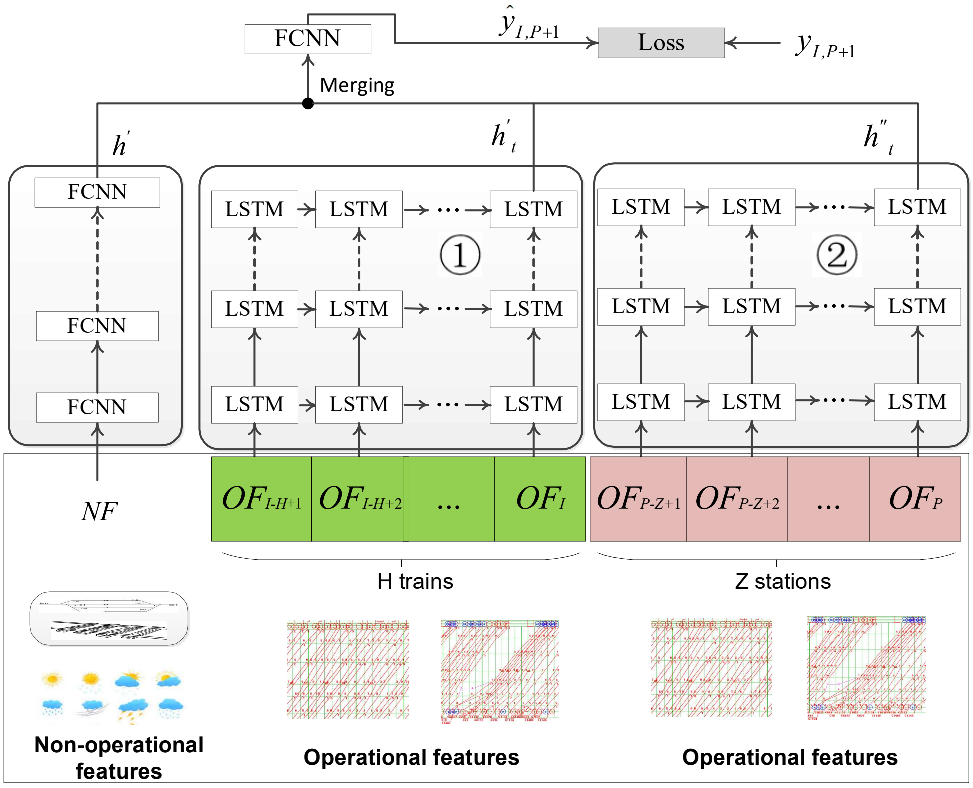


Fig. 2. The structure of FCLL-Net; LSTM component 1 is used for interactions between trains, and LSTM component 2 is used for interactions between stations.

The inputs of both LSTM components are sequences, which are transformed using the operational factors (*OF*) of *H* consecutive trains. To capture the interactions between trains, the operational factors of each train are combined side-by-side, as shown in Eq. (1), and OF_i is then treated as a word of a sentence of the language model to capture the interactions between trains. To capture the interactions between stations, the operational features of *H* trains at/in the same station/section are combined side-by-side, as shown in Eq. (2), and OF_p is then treated as a word of a sentence of the language model to capture the interactions between stations.

$$OF_i = \left[t_{i,P\text{-}Z+1}, t_{i,P\text{-}Z+2}, \ldots, t_{i,P}, w_{i,P\text{-}Z+1}, w_{i,P\text{-}Z+2}, \ldots, w_{i,P}, \ldots, s'_{i,P\text{-}Z+1}, s'_{i,P\text{-}Z+2}, \ldots, s'_{i,P} \right]^{\mathrm{T}}, \quad (1)$$

$$OF_p = \left[t_{I-H+1,p}, t_{I-H+2,p}, \ldots, t_{I,p}, w_{I-H+1,p}, w_{I-H+2,p}, \ldots, w_{I,p}, \ldots, s'_{I-H+1,p}, s'_{I-H+2,p}, \ldots, s'_{I,p} \right]^{\mathrm{T}}, \quad (2)$$

where i is the train indices, $i \in \{I-H+1, I-H+2, \ldots, I\}$, p is the station indices, $p \in \{P-Z+1, P-Z+2, \ldots, P\}$, *I* is the train of interest, *P* is the station at which train *I* just arrives, *H* is the number of trains considered in each prediction, *Z* is the number of stations or sections that the input data are from, and superscript T represents matrix transposition.

3.2. FCNN and LSTM

In an FCNN, the neurons of adjacent layers are fully connected and, thus, information flows are transferred from the input layer to the output layer through intermediary hidden layers (Svozil et al., 1997), as shown in Fig. 3(a). The inputs are treated as static features and fed into the neurons without any given order, which causes the FCNN neurons to only recognize the relationships between the independent and dependent variables. However, an LSTM unit has a “sequence” characteristic, which enables processing sequential inputs, such as time-series data (Grossberg, 2013), and predicting the subsequent patterns (Hochreiter and Schmidhuber, 1997). For example, consider a sequence $X = (x_0, x_1, \cdots, x_t)$. In an LSTM unit, the sequence X is fed into the LSTM unit. The hidden state (h_t) of the LSTM at timestep t is obtained based on the information from the current input x_t and the previously hidden state $h_{t\text{-}1}$, as shown in Fig. 3(b). This enables the LSTM model to memorize the effects of previous elements on the latter elements.

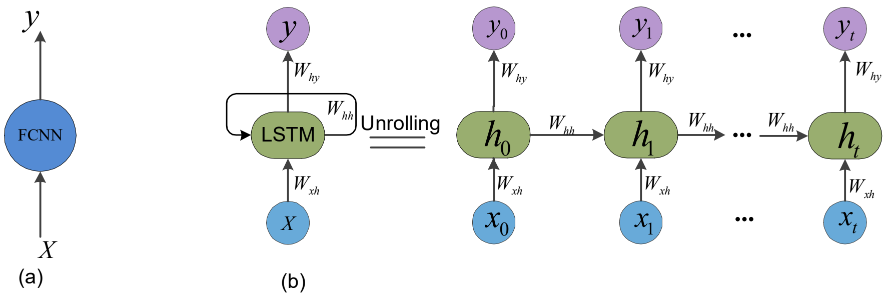


Fig. 3. (a) An FCNN neuron and (b) an LSTM unit with its unrolling form.

This ability of LSTM to capture the cumulative dependencies in sequences lies in its self-controlled gate mechanism (Hochreiter and Schmidhuber, 1997). The most important component in an LSTM unit is its memory cell state c_t, which is constantly present from the first step to the last step, serving as an accumulator to store useful information. The cell state is written, utilized, and cleared through the self-controlled gate mechanism. At each time step, a new input is added to the model, and learned information is written into the cell state if the input gate u_t is activated. Furthermore, the past cell state $c_{t\text{-}1}$ is forgotten if the forget gate f_t is on. The final decision as to whether cell state c_t is used as an input to the output h_t (y_t, if it is the output of the last step) is made by the output gate O_t. The underlying process is expressed by Eqs. (3)-(7):

$$u_t = \sigma\left(W_{xu}x_t + W_{hu}h_{t-1} + W_{cu} \otimes c_{t-1} + b_u\right), \tag{3}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \otimes c_{t-1} + b_f\right), \tag{4}$$

$$c_t = f_t \otimes c_{t-1} + u_t \otimes \varphi\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right), \tag{5}$$

$$O_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \otimes c_t + b_O\right), \tag{6}$$

$$h_t = o_t \otimes \varphi\left(c_t\right), \tag{7}$$

where $\otimes$ indicates the element-wise multiplication of two vectors, $\sigma(x)$ is a threshold function (sigmoid function) of the LSTM unit, the value of which ranges from 0 to 1, and indicates how much information is stored, utilized, and discarded, and $\varphi(x)$ is the tanh function that pushes the output values of LSTM to be between -1 and 1. In the proposed model, the activation function ReLU is used in the FCNN layers to map the non-linear relationships between train delays and the non-operational factors.

$$\sigma(x) = \text{sigmoid}(x) = \frac{1}{1+e^{-x}}, \quad \text{x} \in (-\infty, \infty) \tag{8}$$

$$\varphi(x) = \tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}, \quad \text{x} \in (-\infty, \infty) \tag{9}$$

$$\mathrm{ReLU}(x) = \begin{cases} x, & \mathrm{x} \geq 0 \\ 0, & \mathrm{x}<0 \end{cases} \tag{10}$$

3.3. *Model compilation*

As presented in Fig. 2, the LSTM units are used to capture interactions between trains and stations through the operational factors, whereas the FCNN layers are used to account for the influence of the non-operational factors. In the LSTM components, the initial layers utilize a many-to-many structure to store past information, while the last layer has a many-to-one structure to map all learned information as a multi-dimensional vector.

To train the FCLL-Net, a fusion technique is employed to concatenate the output vectors of the two LSTM components and the FCNN layers. This means that, if the output dimensions of the FCNN layers and two LSTM components are respectively N_{FCNN}, N_{LSTM1}, and N_{LSTM2} as determined by the number of neurons/units in the last layer, the merged vector will have a dimension of $N_{FCNN} + N_{LSTM1} + N_{LSTM2}$. The merged inputs are then passed through another FCNN layer to produce the final output: the predicted delay.

The FCLL-Net model is trained by passing a sample of training data through the network from the input to the output, and then back-propagating the resulting errors from the output to the input (Rumelhart et al., 1988). The quality of the proposed model is quantified by using an objective function that compares the estimated delay to the observed delay to calculate a so-called error term "loss". In other words, the loss is to measure the errors of the model and it needs to be reduced in the training process. The loss of FCLL-Net is given by the mean absolute error (MAE), as shown in Eq. (11). The underlying training algorithm in the Keras package (Chollet, 2015) was implemented on the backend of TensorFlow (Abadi et al., 2016).

, (11)

$$\mathrm{MAE} = \frac{1}{K}\sum_{n=1}^{K}\left|\hat{y}_n - y_n\right|$$

where K is the data volume, $\hat{y}_n$ y_n is the observed delay, and is the predicted delay.

4. Model training

4.1. *Data description and analysis*

The FCLL-Net model described in the previous section was trained and tested using real train operation data from two Chinese HSR lines, namely the Wuhan-Guangzhou HSR (W-G) and Xiamen-Shenzhen (X-S) HSR, as illustrated in Fig. 4. The W-G HSR, which has a length of 1,096 km, is one of the busiest passenger railway lines in China. It intersects with the Guangzhou-Shenzhen, Hengyang-Liuzhou, and Shanghai-Kunming HSRs at the GZS, HYE, and CSS stations, respectively. Trains operating on this line are all equipped with the CTCS (Chinese train control system) with a maximum operating speed of 350 km/h, and an automatic train supervision system (ATS) which keeps track of the movements of all trains. The X-S HSR has a length of 514 km and a maximum train speed of 250 km/h. The focus of this study was the operation of trains in the northbound direction on both lines, i.e., from GZS to CSS in the southern part of the W-G HSR line, and from SZ to CS in the southern part of the X-S HSR line. As shown in Fig. 4, the GZS, HYE, CSS, and SZ stations are junction stations serving as the origins, destinations, and turn-overs of different routes.

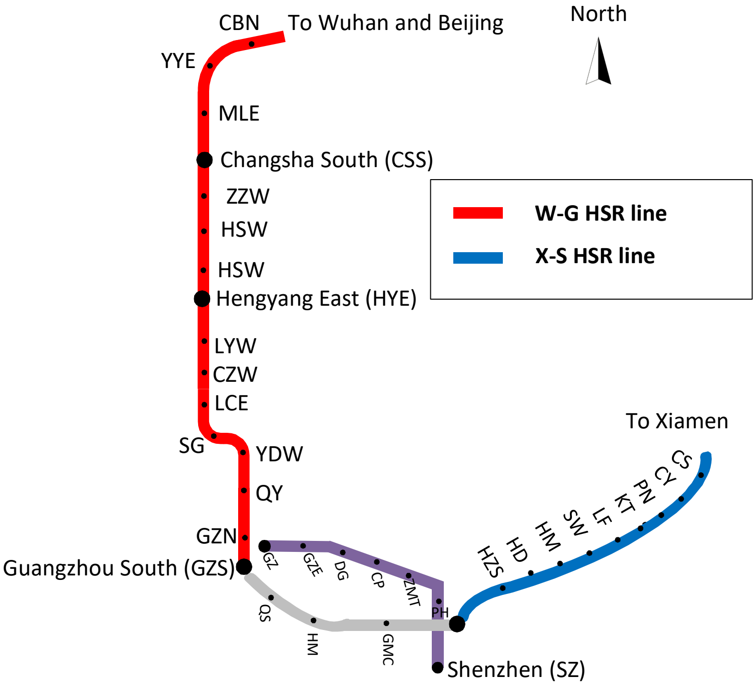


Fig. 4. Layout of the W-G and X-S HSR lines.

The operation data collected from the W-G line included a total of 57,796 train trips along the GZS–HYE segment and 64,547 trips along the HYE–CSS segment, while the X-S HSR data included a total of 41,186 train trips. The data set covered the period from March 24, 2015 to November 10, 2016, and consisted of scheduled/actual arrival/departure times for each train at each station, train numbers, dates, occupied tracks, and section lengths. Table 1 lists four records of the train operation data in the database.

Table 1 An example of the train operation data.

Train	Station	Date	Actual arrival	Actual departure	Scheduled arrival	Scheduled departure	Occupied track
G1002	HYE	2016/10/18	9:37	9:39	9:26	9:28	10
G1016	HSW	2015/3/24	18:01	18:01	17:57	17:57	II
G280	LYW	2015/3/28	8:24	8:24	8:24	8:24	II
G1112	CSS	2015/7/3	14:41	14:45	13:36	13:40	4

The passage tracks at the station are labeled with Roman characters, while the dwelling tracks are labeled with numbers.

Train delays can be caused by both environmental factors, such as severe weather and animals entering the track, and system-related factors, such as track failures and power outage. According to existing records of the causes of delays, the delay causes in the Chinese HSR network can be classified into seven categories: automatic train protection system faults (ATPF), turnout and track faults (TTF), pantograph, catenary, and signal faults (PCSF), rolling stock faults (RSF), foreign body invasions (FI), severe weather (SEW), and other reasons that are not recorded in the systems (OR). Fig. 5 presents the statistical results of the proportions of the causes in the dataset, which indicate that RSF and PCSF composed the largest proportion, both accounting for over 20%, whereas TTF had the smallest proportion, only accounting for 6.68%. In this study, the delays caused by all these reasons were considered to complete the prediction tasks.

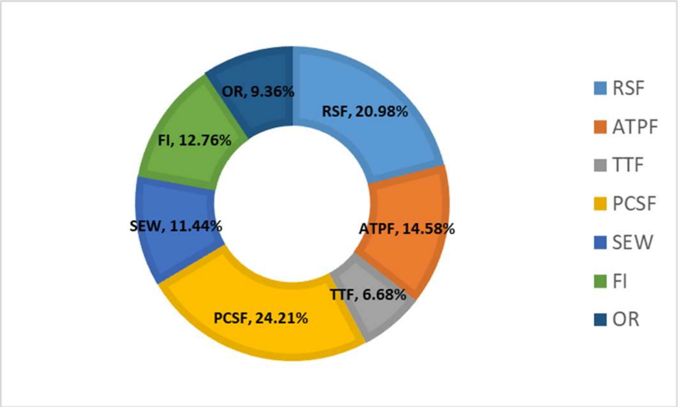
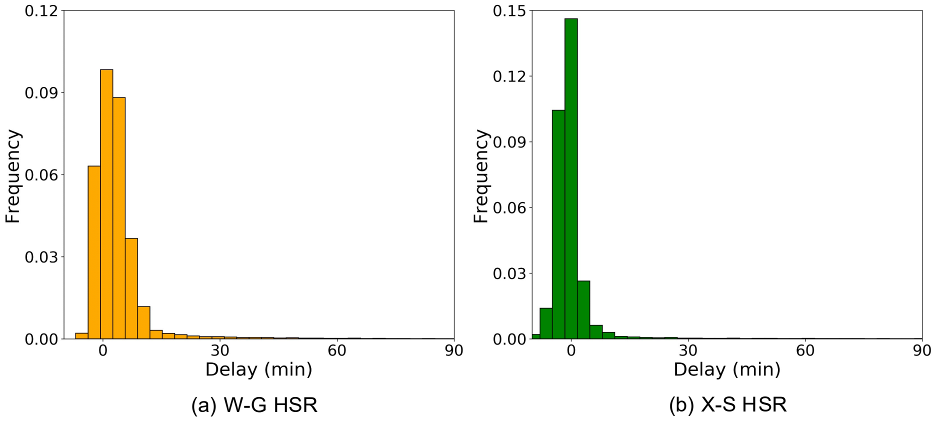


Fig. 5. The proportions of delay causes.

To better understand the patterns of train delays, their distributions on these two lines were analyzed. The histograms in Fig. 6 show that train delays on both HSR lines follow a heavy-tailed distribution, and those located in the tailed area can be identified as long delays. These distributional characteristics have also been ascertained in many other studies that have utilized data from other railway systems (Khadilkar, 2016; Lessan et al., 2019; Marković et al., 2015; Nair et al., 2019). Because longer delays have lower frequencies, the effective prediction of long delays requires much more operational data to train the forecasting model; this partially explains why train delay prediction is a difficult challenge to address (Nair et al., 2019). Statistics show that the average train delay on the W-G HSR is 3.12 minutes, and that on X-S HSR is 1.10 minutes. In addition, according to the standard set by the Chinese Railway Company, trains delayed longer than 4 minutes are labeled as delays; therefore, the average proportion of delays on the W-G HSR is 20.6%, and that on the X-S HSR is 6.6%.



(a) W-G HSR

(b) X-S HSR

Fig. 6. Delay distributions on the W-G and X-S HSRs.

Tables 2 and 3 exhibit the operation features of these two HSR lines, including the arrival/departure punctuality, average arrival/departure delays, percentage of dwelling trains, average actual/scheduled dwell times at each station, and average actual/scheduled running times in each section. Table 2 shows that the punctuality at the CZW and LYW stations on the W-G HSR line is the lowest, and trains at these stations experience the longest average delays. The delays of the northbound trains on the W-G HSR line present an increasing trend during their operations. Trains have a higher dwelling frequency at the CSS, HYE, CZW, and SG stations, and have a longer average running time in the LCE-CZW and ZZW-CSS sections. Table 3 shows that the punctuality at each station on the X-S HSR line is around 94%, and the train delays on this line do not show any specific trend. Trains have a higher dwelling frequency at the SW, PN, CY, and CS stations, and have longer running times in the SW-LF, PN-CY, and CY-CS sections.

Table 2 Operational features of W-G HSR line.

Station	AP	DP	AD (min)	DD (min)	PD	ADT (min)	SDT (min)	ART (min)	SRT (min)
GZN	92.70%	92.64%	1.90	1.93	10.91%	3.36	3.18	9.27	7.60
QY	80.18%	80.42%	3.61	3.59	13.25%	2.26	2.48	12.04	12.63
YDW	85.08%	85.07%	3.01	3.03	11.53%	2.10	2.00	16.86	17.17
SG	85.32%	86.33%	2.72	2.51	64.18%	2.30	2.64	10.99	10.46
LCE	82.57%	82.55%	3.05	3.07	0.00%	12.85	0.00	20.25	19.10
CZW	69.32%	71.29%	4.22	4.09	68.01%	2.05	2.27	18.03	17.95
LYW	70.46%	70.85%	4.19	4.17	17.96%	1.89	2.05	14.55	14.29
HYE	76.80%	76.77%	3.17	3.52	66.41%	3.10	2.61	10.74	9.87
HSW	70.96%	70.51%	4.20	4.17	13.77%	2.41	2.63	15.30	15.45
ZZW	71.31%	72.37%	4.25	4.17	36.53%	2.33	2.56	13.80	18.08
CSS	91.46%	89.70%	-0.06	0.84	79.60%	5.18	3.95		

Arrival punctuality (AP), departure punctuality (DP), average arrival delay (AD), average departure delay (DD), percentage of trains with dwell (PD), average actual dwell time (ADT), average scheduled dwell time (SDT), average actual running time (ART), average scheduled running time (SRT).

Table 3 Operational features of X-S HSR line.

Station	AP	DP	AD (min)	DD (min)	PD	ADT (min)	SDT (min)	ART (min)	SRT (min)
HD	93.54%	93.61%	1.25	1.21	28.32%	2.12	2.35	11.43	11.89
HM	94.00%	94.06%	0.78	0.81	16.61%	2.22	2.18	12.59	11.95
SW	92.91%	92.81%	1.47	1.69	78.48%	2.30	2.03	13.69	14.37
LF	93.40%	93.39%	1.03	1.09	40.42%	2.29	2.17	10.36	10.21
KT	93.13%	93.14%	1.25	1.27	18.17%	2.28	2.21	12.10	12.42
PN	93.05%	92.79%	0.96	1.26	52.57%	2.59	2.03	14.01	13.96
CY	92.72%	92.68%	1.31	1.45	44.02%	2.29	2.00	14.08	15.11
CS	95.01%	94.31%	0.34	0.68	59.47%	3.19	2.34		

Arrival punctuality (AP), departure punctuality (DP), average arrival delay (AD), average departure delay (DD), percentage of trains with dwell (PD), average actual dwell time (ADT), average scheduled dwell time (SDT), average actual running time (ART), average scheduled running time (SRT).

The values of the 14 variables ascertained in Section 2.2 were extracted, and some data pre-processing techniques were applied before training the model. These included: 1) filling in missing data using the mean value of adjacent records, 2) replacing abnormal observations (e.g., the minus running time and minus dwelling time) using the mode value of the respective variable, and 3) standardizing the input to eliminate the dimensions of data. Because the trains were ordered according to their actual departure time at the original station, the variable R'_i (scheduled intervals) is likely to include minus values, as train overtaking may happen during disturbances. We thus kept the minus values of variable R'_i on both lines. In addition, because the train operation data included empty-ride trains, and because it is possible that the empty-ride trains arrive or depart much earlier than their scheduled times, all the minus delays on both lines were therefore kept. For the selected factors in Section 2.2, *M* and *R'* are integers, where $m_n \in \{2,4,6,7,10,12\}$ on the W-G HSR line, and $m_n \in \{2,4,6,10\}$ on the X-S HSR line; additionally, $r'_{i,n} \in \{0,1,2\}$ for one-step prediction, and $r'_{i,n} \in \{0,1,...,\mathrm{Q}+1\}$ for Q-step prediction on both lines. The statistical analysis of the continuous variables (before standardization) is presented in Table 4.

Table 4 Statistics of continuous variables of W-G and X-S HSR lines.

Variable	W-G				X-S			
	Min	Mean	Max	SD	Min	Mean	Max	SD
C (℃)	-12.20	22.24	40.20	8.31	1.30	25.27	37.60	5.81
V (m/s)	0.00	2.69	21.40	2.03	0.00	2.61	19.00	1.35
F (mm)	0.00	0.27	71.60	1.68	0.00	0.25	60.00	1.67
L (min)	37.00	57.21	84.00	5.50	25.00	31.21	35.00	3.20
T (min)	6.00	14.71	157.00	4.21	6.00	12.35	173.00	2.23
R (min)	3.00	10.48	203.00	7.70	3.00	15.43	324.00	18.95
W (min)	0.00	0.87	165.00	2.11	0.00	1.13	201.00	1.98
T' (min)	7.00	14.26	45.00	3.81	8.00	12.76	35.00	1.97
R' (min)	-92.00	10.39	46.00	6.24	-92.00	12.43	38.00	6.78
W' (min)	0.00	0.87	60.00	1.60	0.00	0.93	13.00	1.13
D (min)	-48.00	3.19	191.00	9.52	-41.00	1.20	321.00	12.63
Y' (min)	-56.00	3.19	191.00	9.41	-41.00	0.99	320.00	12.58

SD stands for standard deviation.

To enable the interactions to be captured from the inputs, the data of five trains from five stations and sections (*H* and Z equal 5 in Section 2.2) were used to predict future delays. For the delay prediction at stations where the northbound trains have fewer than five past stations and sections, e.g., GSN, QY, YDW, and SG on the W-G HSR, and HD, HM, SW, and LF on the X-S HSR, the influencing factor vectors were filled with the mode value of the respective variable so they would all have the same vector length. The delay prediction data (input and output of the model) of each station on these two HSR lines were stacked, and thus 651,264 and 323,584 cases for modeling on the W-G and X-S HSR lines were respectively obtained. After transforming every *H* consecutive trains as train groups, the data were then randomly split into three separate data sets: 1) data from 55% of all operating days for model training, which included 358,400 trains on the W-G HSR line and 178,176 trains on the X-S HSR line; 2) data from 20% of all operating days for model parameter tuning (validation), which included 129,024 trains on the W-G HSR line and 63,488 trains on the X-S HSR line; 3) the remaining 25% of data for model testing, which included 163,840 trains on the W-G HSR line and 81,920 trains on the X-S HSR line.

4.2. Model calibration and design

The proposed FCLL-Net described in the previous section entails a large number of variations in terms of structural settings and model parameters, which can be determined through a calibration and evaluation process for achieving optimal performance for a specific domain of problems. In this section, the results of the experimental process are presented, aiming at determining the best model for the train delay prediction problem.

The first set of experiments focused on the effect of the number of hidden layers and neurons on the performance of the model. Because these two parameters are mutually dependent (the increasing of one can lead to the change of the other), researchers usually manually set one parameter considering the data size and dimension while investigating the other parameter to obtain a well-fitted model, such as in existing successful studies about LSTM (Graves et al., 2013; Sak et al., 2014). Considering the number of operational variables (nine) and non-operational variables (five) ascertained in Section 2.2, 128 units were manually set in each LSTM layer and 64 neurons were manually set in each FCNN layer to investigate the number of hidden layers in the LSTM and FCNN components, respectively. The model was first trained with only one hidden layer in each component (LSTM components 1 and 2, and the FCNN component), and layers were gradually added until the validation loss of the model stopped decreasing. In addition, because the two LSTM components have the same input data (only the data shapes are different), it was assumed that they also have the same number of hidden layers. Thus, these two components were first optimized together, and the FCNN components were then optimized. Fig. 7 presents the relationship between the model performance (validation losses) and the number of hidden layers. The results indicate that there is an optimal number of hidden layers in terms of model performance for the problem. These results are intuitive, as a neural network structure with too few layers or neurons would not be able to capture the full operational dependency of the trains and the influencing factors of system, whereas one with too many layers or neurons could lead to over-fitting. Based on these results, it was decided to use three hidden layers for both LSTM components, each with 128 units, and four hidden layers for the FCNN component, each with 64 neurons, in the proposed FCLL-Net model. In addition, Fig. 7 also presents the effect of the model structure on its computational efficiency. The results indicate that the training time of the model was sensitive to the LSTM layers, which is mainly because the LSTM architecture needs to process the sequence inputs from the first element to the last.

The performance of neural network models can also be influenced by other parameters. However, to avoid over-optimization that requires too much computation cost, parameters were borrowed from other successful studies, such as the Adam optimizer (Kingma and Ba, 2014), which updates the local learning rate according to the training process, the initial learning rate (0.001) that is used by default in the Adam optimizer (Kingma and Ba, 2014), the "ReduceLROnPlateau" technique in the Keras package to reduce the global learning rate in cases where the loss does not decrease after five steps to avoid over-fitting (Chollet, 2015), and the ReLU activation function, which is used in neural network models by default in the Scikit-learn package (Pedregosa et al., 2011). All the hyper-parameters used in the FCLL-Net model are summarized in Table 5.

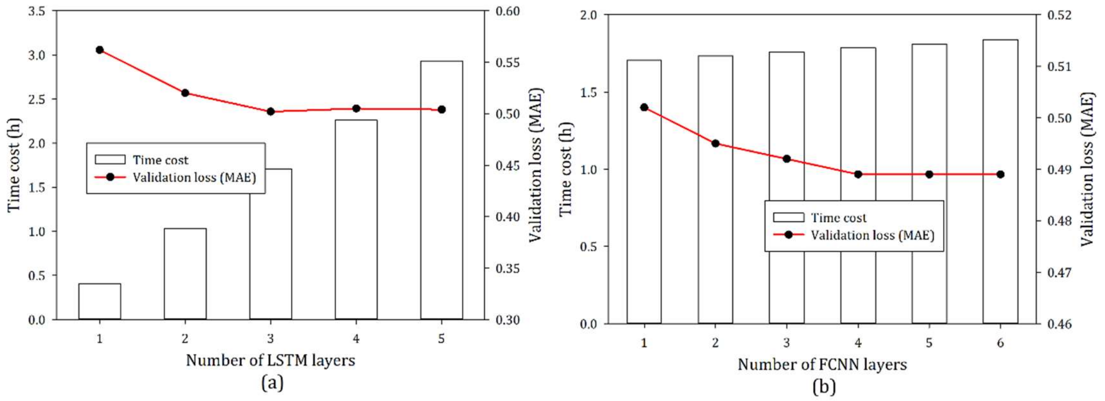


Fig. 7. Model performance (validation loss/MAE) vs. number of hidden layers.

Table 5 Structure and parameters of FCLL-Net.

Model architecture	FCNN component: 4 hidden layers, each with 64 neurons; LSTM component 1: 3 hidden layers, each with 128 units; LSTM component 2: 3 hidden layers, each with 128 units;
Optimizer	Adam

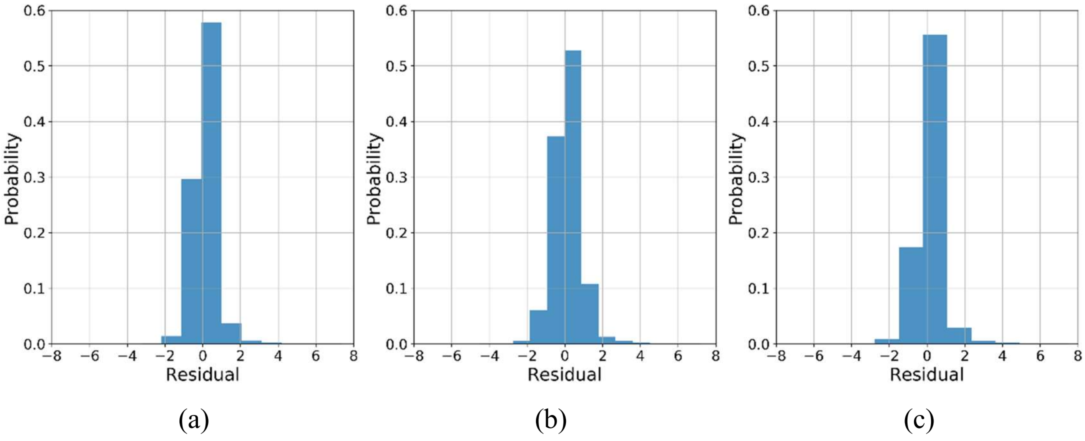
Initial learning rate	1×10^{-3}
Activation function	ReLU
Techniques to avoid over-fitting	Cross-validation and ReduceLROnPlateau
Learning rate reduction	50%
Training steps/Epochs	150
Mini-batch	2048

5. Results

5.1. Performance analysis

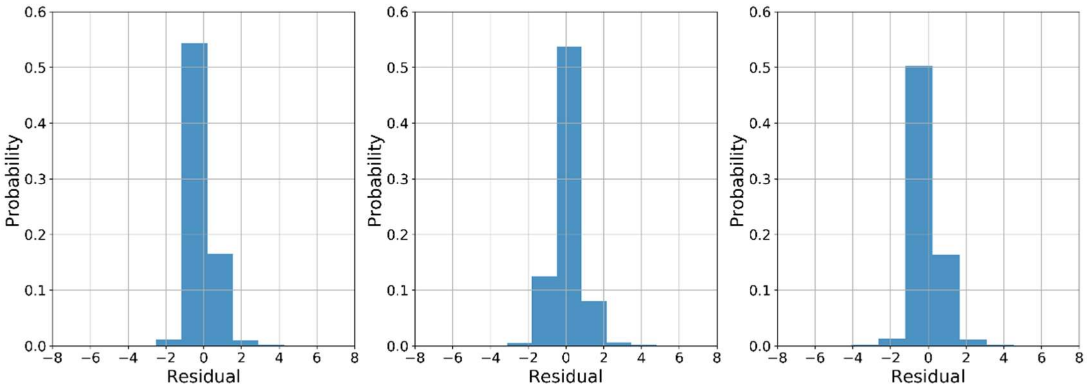
Residuals are the differences between fitted or predicted values and observed values. The residuals of training data can indicate the goodness-of-fit of the regression model, whereas the residuals of the validation and testing data can show the prediction errors of the predictive model. The probability distributions of residuals of the datasets were first analyzed, as shown in Figs. 8 and 9. These figures indicate that the residuals of the training, validation, and testing datasets all have the highest probability at zero, and all the residuals follow a normal distribution. This indicates that the residuals of the proposed model meet both the zero-mean and normal distribution assumptions. In addition, the cumulative distribution function (CDF) of the residuals are shown in Fig. 10, which demonstrates that the CDF of the training dataset can encircle the CDF curves of the validation and testing datasets, which indicates that the model has smaller errors on the training dataset. This is understandable, as the model was formulated using the training dataset. This is a common conclusion that can also be found in other studies based on the regression technique (Ma et al., 2017). In addition, as the model parameters were tuned using the validation data, and because the CDF curves of validation data and testing data are well coincident on both railway lines, which indicates that the proposed model may have good generalizability, and can be well applied to prediction tasks.

The influence of the data horizon on model accuracy was then analyzed. Generally, the predictive errors will increase when the data horizon grows, as the longer delays are more random, thus leading to greater difficulty in prediction. Fig. 11 shows that when the considered horizon grows, the errors on the training, validation, and testing data all increase. The accuracy of the model on data with a 10-minute horizon is almost double that on data with a 120-minute horizon. However, the figure shows that the larger the horizon is, the lower the increasing rate of errors will be. This is because the train delays follow a heavy-tailed distribution, and longer delays have lower frequencies. Finally, Fig. 11 also shows that the errors of the training dataset are smaller than those of the validation and testing datasets on all horizons, which corresponds to Fig. 10.



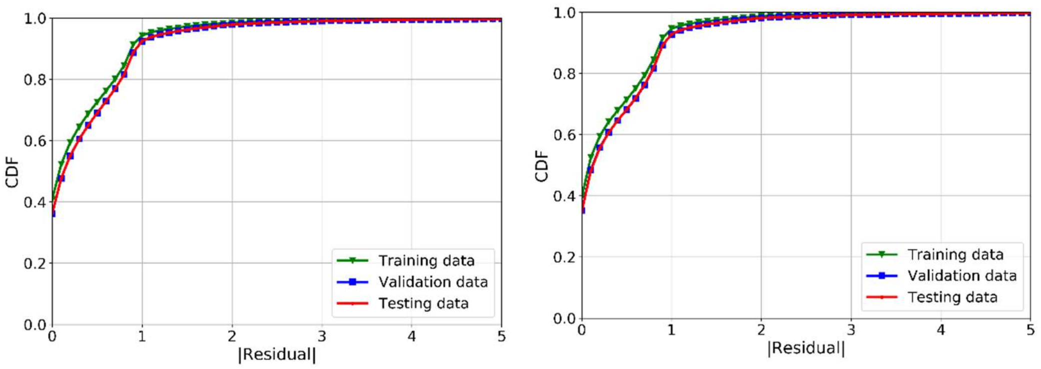
(a) (b) (c)

Fig. 8. Residual distribution of (a) training, (b) validation, and (c) testing dataset of W-G HSR.



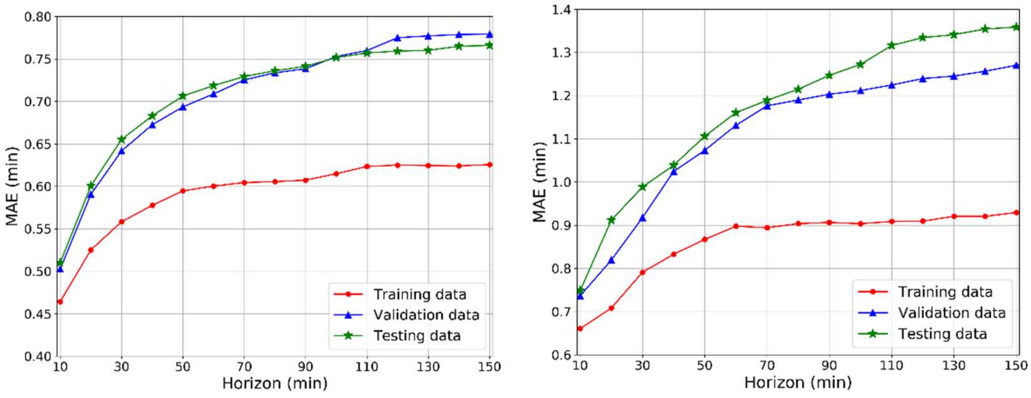
(b) (b) (c)

Fig. 9. Residual distribution of (a) training, (b) validation, and (c) testing dataset of X-S HSR.



(a) (b)

Fig. 10. Cumulative distribution of residuals of (a) W-G and (b) X-S HSR.



(a) (b)

Fig. 11. MAE for data horizons on (a) W-G and (b) X-S HSRs.

5.2. Comparative analysis with other models

5.2.1. Benchmarks

To compare the performance of the proposed model, four widely used train delay prediction models, namely the support vector regression, deep extreme learning machine, random forest, and multilayer perceptron are chosen as benchmarks. Moreover, four state-of-the-art deep learning models widely used in traffic prediction (Wang et al., 2019), including the deep brief networks, hybrid model composed of stacked autoencoders and FCNNs, and two hybrid models containing LSTMs and FCNNs for capturing interactions between trains and stations, respectively, were selected as baseline models. The basic principles and parameters of these baseline models are given as follows.

1) A support vector regression (SVR) is a supervised machine learning technique that constructs a hyperplane or set of hyperplanes in a multi-dimensional space, which can be used for classification and regression (Smola and Schölkopf, 2004). From an input *X*, the SVR calculates a predicted value and sets a threshold ε to assess the difference between the predicted value and the true value. Only when their difference exceeds the threshold value is the loss counted. Such a model was used in previous studies (Barbour et al., 2018; Marković et al., 2015) to predict train delays. The SVR was established using the Scikit-learn package, and the kernel function, penalty function coefficient, and regularization of the loss function were optimized (Pedregosa et al., 2011). The kernel function was chosen from *linear*, *rbf*, *sigmoid*, and *poly*, the penalty function coefficient was chosen from 0.01, 0.1, 0.5, 1, 2, 5, 10, and 100, and the loss function was chosen from the *epsilon-insensitive loss* and *squared-epsilon-insensitive loss* functions, the details of which are provided in the work by (Pedregosa et al., 2011). Finally, the kernel function of the SVR used in this study was a *linear* kernel function, the penalty function coefficient was 2, and the loss function was *epsilon-insensitive loss*.
2) The deep extreme learning machine (DELM) is a shallow extreme learning machine (SELM) with multiple hidden layers. The SELM was originally developed based on the single hidden layer perceptron (Huang et al., 2006). In SELM, weights and biases are set randomly instead of being updated using the back-propagation algorithm. The most significant strongpoint of DELM is that it requires less training time due to its randomized parameters (Huang et al., 2006). The DELM proposed by (Oneto et al., 2017, 2018) has been proven to outperform the timed event graph used by (Kecman and Goverde, 2015a). The DELM in this study was established using the Helm package (Akusok et al., 2015), and its hidden layers, the number of neurons in each layer, and the type of neurons were optimized. The hidden layers and number of neurons were optimized with the method used to optimize FCLL-Net; the type of neuron was chosen from *linear, sigmoid, tanh, rbf_l1, rbf_l2, rbf_linf*, details of which are provided in the work by (Akusok et al., 2015). The DELM used in this study was optimized with three layers, each of which had 512 neurons, and the type of neuron was *rbf_l2*.
3) Random forest (RF) is composed of decision trees. The final result of a regression RF model is the average of the results of the decision trees (Liaw and Wiener, 2002). Recently, the RF model has been widely used in train delay prediction (Jiang et al., 2018; Nabian et al., 2019; Nair et al., 2019). The RF was established using the Scikit-learn package (Pedregosa et al., 2011), and the number of decision trees and the depth (number of splits) of each tree were optimized. The number of decision trees was chosen from 50, 100, 150, 200, 250, 500, and 1000, and the depth of each tree was chosen from 3, 6, 9, 12, 15, 18, 21, and 24. Finally, 150 decision trees were used, each with 12 splits.
4) Multilayer perceptrons (MLPs), also known as fully-connected neural networks (FCNN), are a type of feedforward neural network that consists of at least one hidden layer that is fully connected between adjacent layers (Svozil et al., 1997). The model is trained by back-propagating the prediction errors from the output layer to the hidden layer(s), and finally to the input layer with each step, including a process to revise the weights of the connectors (Rumelhart et al., 1988). A MLP used in (Yaghini et al., 2013) was selected as a baseline model,

but its parameters were optimized based on the HSR operation records. The MLP was established using the Scikit-learn package (Pedregosa et al., 2011). The numbers of hidden layers and neurons were optimized, as was the activation function. The hidden layers and neurons were optimized with the method used to optimize the layers and neurons of FCLL-Net. The number of neurons of the MLP in each layer was set to 128, and hidden layers were added to obtain the well-fitted model. Finally, five hidden layers were used, each with 128 neurons. The activation function was chosen from *identity, logistic, tanh, and relu*, and the *relu* activation function was used.

5) Hybrid FCNN and LSTM model used for capturing interactions between trains (FCL-IT). This model contains an FCNN component and an LSTM component. In the FCL-IT model, the FCNN component is fed with non-operational features, whereas the LSTM component is fed with operational features to capture the interactions between trains. Other parameters in FCL-IT are the same as those of FCLL-Net. The comparison of FCL-IT and FCLL-Net can demonstrate the ability of FCLL-Net to capture the interactions between stations.
6) Hybrid FCNN and LSTM model used for capturing interactions between stations (FCL-IS). This model also only contains an FCNN component and an LSTM component. In the FCL-IS model, the FCNN component is fed with non-operational features, whereas the LSTM component is fed with operational features to capture the interactions between stations. Other parameters in FCL-IS are the same as those of FCLL-Net. The comparison of FCL-IS and FCLL-Net can demonstrate the ability of FCLL-Net to capture the interactions between trains.
7) Deep brief networks (DBNs). DBNs are probability generating models stacked using restricted Boltzmann machines (RBM) (Hinton and Salakhutdinov, 2006), which try to explain the relationships between the observation data and label data using joint distribution. An RBM has three parts, including input layer (visible unit), hidden layer (hidden unit), and bias unit. Training DBN model has two steps, including pre-training and fine-tuning (Hinton et al., 2006). Pre-training is a greedy layer-wise training process which trains the RBMs from the bottom to the top with an unsupervised learning method. In fine-tuning stage, the back-propagation algorithm is used to update the weight value and bias value of the whole network. The parameters of DBN, which were optimized in this study, include the hidden layer, the number of nodes in each layer, learning rate, and activation function. For the hidden layers and the number of nodes in each layer, we also used the method to optimize the two parameters of the FCLL-Net and MLP to optimize those of the DBN. The number of nodes in each layer was set at 256, the learning rate was chosen from 0.0005, 0.001, 0.003, 0.005, 0.01, and 0.05, the activation function was chosen from *relu* and *sigmoid.* Finally, the model included two hidden layers, each with 256 units, the learning rate was 0.001, and the activation function was *relu.*
8) A hybrid model containing stacked autoencoders (SAE) and fully connected neural networks (FCNN), named (SAE+FCNN). An autoencoder is a kind of neural network models used for unsupervised learning (Bengio et al., 2007). SAE is the stack of autoencoders to form a deep network to extract the features and obtain the representations of the input. In the SAE-FC model, the SAE model is first trained using unsupervised learning method to obtain the representations of input data. Then, the FCNN was trained with a supervised learning method to update its parameters, where the input is the output of the SAE and the output is the target (to-be-predicted) data. We optimized the hidden layers, the number of nodes in each layer, and the activation function of both SAE and FCNN. Finally, the structure of the SAE was (N, 256, 128, 256, N), where N is the dimension of the input data (350 in this study); the activation function of SAE was *relu* function. The parameters of FCNN is the same with the baseline

model MLP introduced before.

In the training, validating, and testing of these base models, the same train records that were used for the FCLL-Net model were applied in the training, validation, and testing datasets. Also, both the operational and non-operational features were used in the base models. Two commonly used prediction quality metrics were applied, namely mean absolute error (MAE) and mean absolute percentage error (MAPE), as defined in Eqs. (11)-(12), which can be calculated based on the observed delays and the predicted delays. The main purpose of a predictive model is to analyze the delayed states (longer than 4 minutes, as labeled by Chinese Railway Corporation) at future checkpoints, and longer delays have larger impacts on train operations. All metrics are simultaneously calculated on the short delays, which are located in the interval of [4, 30] minutes, and on the long delays, which are located in the interval of (30, max] minutes. In the following analyses, the MAE and MAPE calculated on the short delays are respectively denoted by "MAE$^{\#}$" and "MAPE$^{\#}$", and those calculated on the long delays are denoted by "MAE$^{\#\#}$" and "MAPE$^{\#\#}$".

$$\mathrm{MAPE} = \frac{1}{K}\sum_{n=1}^{K}\left|\frac{\hat{y}_n - y_n}{y_n}\right| \times 100\% \quad , \tag{12}$$

where K is the data volume, y_n is the observed train delay, and $\hat{y}_n$ is the predicted train delay.

5.2.2. *Model performance on one-step prediction*

The models were first run to make one-step predictions of delays from GZN to CSS on the W-G HSR line, and from HD to CS on the X-S HSR line. These two HSRs have different features, as introduced in Section 4.1, and are therefore suitable for demonstrating the generalizability of the proposed model.

The overall relative performance of FCLL-Net was first compared against the baseline models when applied to the operations of the W-G and X-S HSR lines. The predictive performances of the FCLL-Net model against those of the baseline models for the W-G and X-S HSR cases are exhibited in Table 6. CLL-Net distinctly outperforms other common train delay prediction models, i.e., MLP, DELM, RF, SVR, BN, and MM, on short and long delays on both HSR lines in terms of MAE and MAPE. Compared with the widely used delay prediction models, it presents average improvements of 17.7% and 19.0% on the W-G and X-S HSRs, respectively. Further, FCLL-Net also has satisfactory improvements compared with the state-of-the-art deep learning models, i.e., SAE + FCNN, DBN, FCL-IT and FCL-IS. Compared with the SAE + FCNN, DBN, FCL-IT and FCL-IS models, it presents average improvements of 9.4% and 13.1% on the W-G and X-S HSRs, respectively.

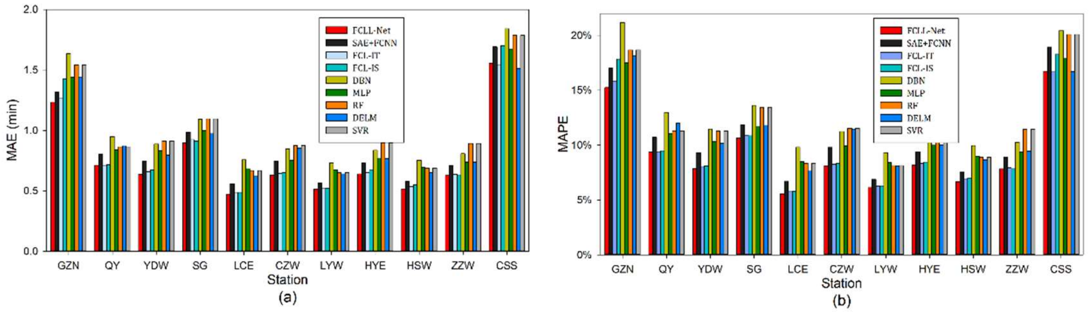
The training times for each model were also recorded, and are exhibited in the last column of Table 6. It shows that the simple statistics-based models, BN and MM, are much more efficient than the machine learning models; however, there is a trade-off between accuracy and training efficiency. Although FCLL-Net takes approximately 106 and 54 minutes for each run using data from the W-G and X-S HSR lines, respectively, on the utilized device (an AMD Ryzen 7 2700 CPU), it only requires one training cycle for use in practice. Further, the TensorFlow package supports the speeding up of training on multiple GPUs, which means that the training time can be reduced by training with better and more GPUs.

Table 6 The overall performance of models on W-G and X-S HSRs.

HSR line	Model	MAE$^{\#}$(min)	MAPE$^{\#}$(%)	MAE$^{\#\#}$(min)	MAPE$^{\#\#}$(%)	Time cost (min)
W-G HSR	FCLL-Net	0.659	8.162	1.876	3.511	106.722
	SAE+FCNN	0.746	9.391	1.947	3.678	19.867
	FCL-IT	0.673	8.339	2.113	3.823	88.652
	FCL-IS	0.687	8.468	1.961	3.674	90.898
	DBN	0.887	11.430	2.134	3.974	61.650
	MLP	0.807	10.144	2.278	4.140	43.240
	RF	0.872	11.038	2.163	4.046	17.670
	DELM	0.798	10.261	2.104	3.914	8.471
	SVR	0.872	11.038	2.163	4.046	4.315
X-S HSR	FCLL-Net	0.989	10.679	2.731	3.951	54.231
	SAE+FCNN	1.215	13.479	2.855	4.314	10.500
	FCL-IT	1.007	10.874	3.203	4.389	35.949
	FCL-IS	1.104	11.668	2.741	3.994	44.925
	DBN	1.556	17.281	3.213	4.868	33.783
	MLP	1.378	15.063	3.226	4.714	21.492
	RF	1.409	15.828	3.122	4.740	7.201
	DELM	1.158	12.871	3.044	4.460	6.436
	SVR	1.201	13.262	3.250	4.808	2.483

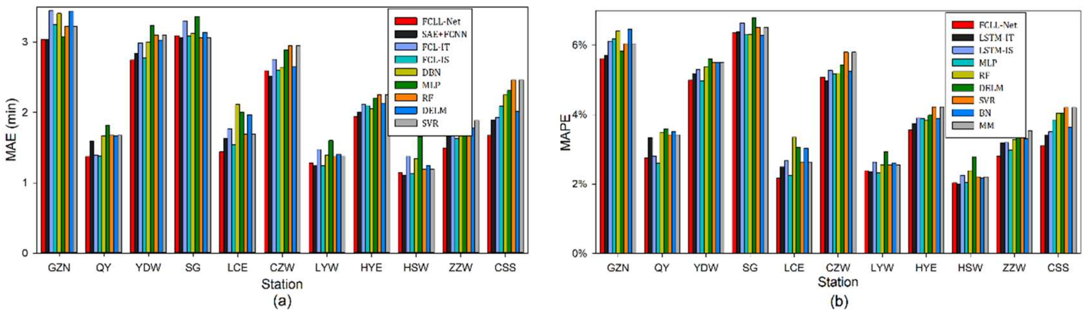
The metrics followed by an octothorpe represent calculating on short delays, and metrics followed by two octothorpes represent calculating on long delays.

The predictive errors of the models at each station of the W-G and X-S HSRs were respectively analyzed. It can be observed from Figs. 12-15 that FCLL-Net outperforms other baseline models in terms of both MAE and MAPE on both short and large delays, except for the prediction of short delays at the HD station. This is not unexpected, as this station is close to the original station. The model's input includes only the limited features of the one last station and section, which means that the full advantages of the FCLL-Net model are not utilized. In addition, as demonstrated in Section 4.1, the frequency of train services on the X-S line is lower than that on the W-G line, which means that less interaction could be captured by FCLL-Net.



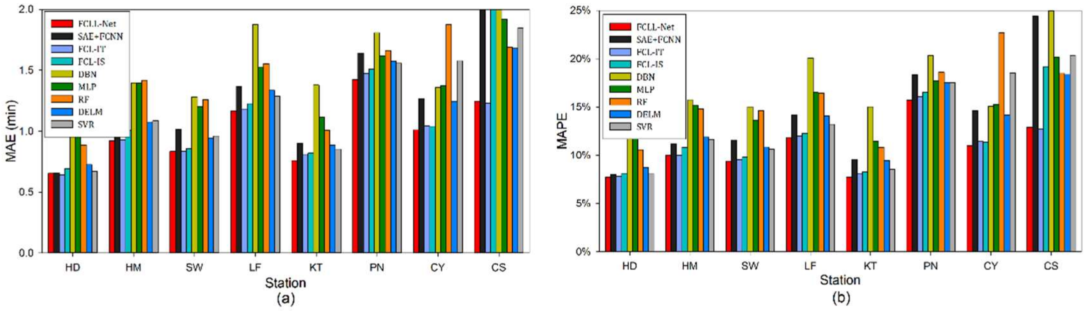
(a) (b)

Fig. 12. MAE (a) and MAPE (b) of each model on short delays of W-G HSR.



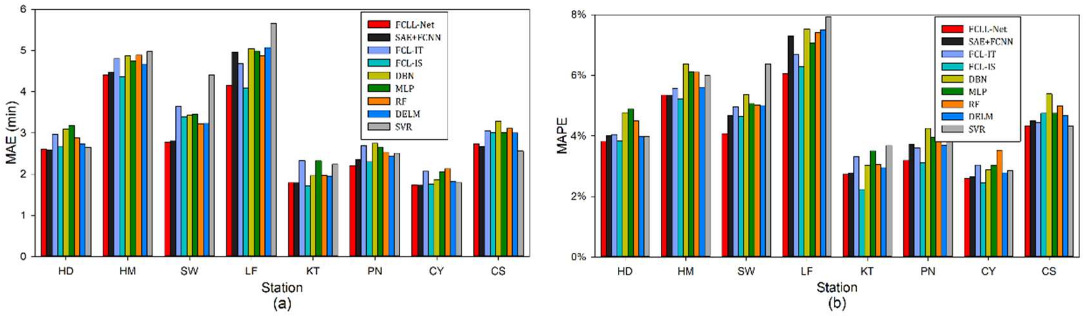
(a) (b)

Fig. 13. MAE (a) and MAPE (b) of each model on long delays of W-G HSR.



(a) (b)

Fig. 14. MAE (a) and MAPE (b) of each model on short delays of X-S HSR.



(a) (b)

Fig. 15. MAE (a) and MAPE (b) of each model on long delays of X-S HSR.

In addition, the delay prediction results of some important stations on the W-G and X-S HSR lines were analyzed. As the data analyses in Section 4.1 reveal, delay severity exhibits a difference from stations on the W-G HSR; the CZW and LYW stations have the longest average delay. Further, the HYE and CSS stations on the W-G HSR intersect with other HSR lines. As such, further attention should be paid to the delay prediction at these stations. Delays on the X-S station do not exhibit any trend, and there is no intersection station on this line. Therefore, the SW, PN, CY, and CS stations were chosen as the important stations on this line, as trains have a higher dwelling frequency at these stations, and higher dwelling frequencies of delayed trains can increase the total waiting time for passengers; this is one of the most important occurrences that dispatchers try to minimize when rescheduling delayed trains. The improvement of the FCLL-Net at these stations compared to the common delay prediction models and the state-of-the-art models is also presented. Table 7 reveals that the FCLL-Net exhibits improvements over both the common delay prediction models, i.e., MLP, RF, DELM, and SVR, and the state-of-the-art deep learning models, i.e., SAE+FCNN, FCL-IT, FCL-IS, and DBN. At the important stations of the W-G and X-S HSRs, it presents respective average improvements of 17.5% and 20.4% compared to the common delay prediction models, and respective average improvements of 9.5% and 16.0% compared to the state-of-the-art deep learning models.

Table 7 The improvements of FCLL-Net at important stations compared against the delay prediction models and the state-of-the-art models.

line	Station	State-of-the-art models				Common delay prediction models			
		MAE#	MAPE#	MAE##	MAPE##	MAE#	MAPE#	MAE##	MAPE##
W-G	CZW	12.7%	14.2%	1.3%	1.6%	25.0%	27.4%	9.3%	8.9%
	LYW	12.5%	14.7%	4.5%	3.1%	21.6%	25.0%	11.0%	10.3%
	HYE	11.3%	11.5%	6.1%	7.1%	23.1%	23.8%	12.1%	12.6%
	CSS	8.3%	9.9%	17.9%	16.3%	8.0%	10.5%	27.6%	23.0%
X-S	SW	16.3%	18.1%	16.0%	17.0%	23.6%	24.4%	22.2%	24.1%
	PN	11.5%	11.9%	12.3%	13.2%	12.2%	12.0%	12.7%	17.9%
	CY	14.1%	16.2%	6.5%	5.5%	28.3%	37.8%	11.1%	14.7%
	CS	39.8%	39.7%	9.2%	9.4%	37.6%	33.4%	6.5%	7.8%

The metrics followed by an octothorpe represent calculating on short delays, and metrics followed by two octothorpes represent calculating on long delays

In train operations, delay jumps mean that trains which suffer second disturbances or delays are recovered using time supplements and buffer times in the timetable. The accurate prediction of delay jumps is one of the most difficult tasks in delay prediction, but is important for dispatchers. With the testing data and the predicted results, we calculated the observed and predicted delay change values using the observed and predicted delays at the target station minus the observed delays at the current station. Then, we treated the increase or decrease/recover delays over 1 min as delay jumps; otherwise, they are treated as system deviations. Table 8 shows five cases of the data used for evaluating the model performance in delay jump.

Table 8. The dataset used for delay jump evaluation.

D_C	D_T		Change value		Jump?	
	O	P	O	P	O	P
1.00	2.00	1.76	1.00	0.76	No	No
40.00	40.00	40.23	0.00	0.23	No	No
0.00	5.00	6.88	5.00	6.88	Yes	Yes
8.00	10.00	10.29	2.00	2.29	Yes	Yes
9.00	7.00	8.11	2.00	0.89	Yes	No

D_C indicates the observed delays at the current station; D_T indicates the delays at the target station; O and P represent the observed and predicted values, respectively.

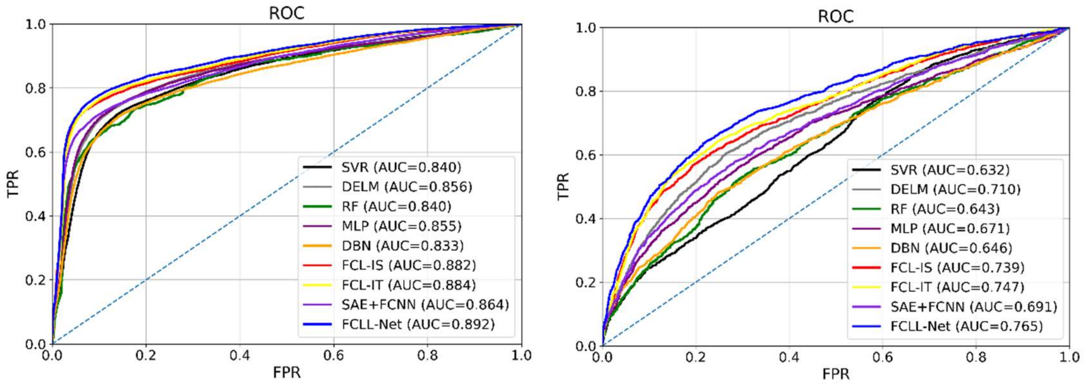
The discretization operation above turns the prediction problem into a classification problem, which can be evaluated using the performance metrics of the receiver operating characteristic (ROC) curve and the area under the curve (AUC). ROC curves are used to choose the most appropriate cut-off point at which the highest true positive rate (TPR) and the lowest false positive rate (FPR) are reached. The ROC curves were plotted with the FPR as the x-axis and the TPR as the y-axis. The definitions of TPR and FPR are shown as the confusion matrix in Table 9, and by Eqs. (13)-(14).

$$TPR = \frac{TP}{TP + FN} \tag{13}$$

$$FPR = \frac{FP}{TN + FP} \tag{14}$$

Table 9. Confusion matrix

	Predicted	
Actual	No. of True Positives (TP)	No. of False Negatives (FN)
	No. of False Positives (FP)	No. of True Negatives (TN)



(a) (b)

Fig. 16. ROC and AUC of each model for the (a) W-G and (b) X-S HSR lines.

Fig. 16 indicates the ROC and AUC of each model, in which the diagonal dashed line represents random prediction. The larger the area surrounded by the curve, the better the performance. This figure demonstrates that the ROC curve of FCLL-Net, with AUC = 0.892 on the W-G HSR and AUC = 0.765 on the X-S HSR, encircles those of the baseline models, which means that FCLL-Net also presents the best performance on trains with delay jumps. This also confirms the better performance of FCLL-Net for delay prediction. In addition, because the MM and BN models only use the known delays to predict future delays, they are incapable of capturing the delay jump information without learning information from other operational and non-operational features; this led to their performances being even worse than random prediction.

5.2.3. Model performance on Q-step prediction

The models were also run to make Q-step predictions, which means that the target was $y_{I,\ P+Q}$, as indicated in Section 2.2. The model performance on two-step and three-step predictions was respectively investigated at the important stations of these two HSR lines, namely the CZW, LYW, HYE, and CSS stations on the W-G HSR line, and the SW, PN, CY, and CS stations on the X-S HSR line. The data from the last five stations and sections of five trains were also used as the input for the models. The parameters of all the models were the same as those of the models in one-step prediction. The results of the two-step and three-step predictions of each model on the W-G and X-S HSRs are exhibited in Tables 10-13. It can be seen that FCLL-Net outperforms other baseline models for the short and long delays on both HSR lines in terms of both MAE and MAPE.

Table 10 Two-step predictive errors of models at important stations on W-G HSR.

Metrics	Station	FCLL-Net	SAE+FCNN	FCL-IT	FCL-IS	DBN	MLP	RF	DELM	SVR
MAE$^{\#}$ (min)	CZW	**0.870**	1.000	0.881	0.900	1.395	1.143	1.205	1.086	1.295
	LYW	1.123	1.269	**1.120**	1.138	1.663	1.352	1.388	1.399	1.556
	HYE	**0.984**	1.100	0.991	1.028	1.691	1.177	1.210	1.180	1.159

	CSS	**2.434**	2.974	2.510	2.667	3.281	2.608	2.792	2.490	2.793
MAPE# (%)	CZW	**10.571**	12.437	10.627	10.934	18.095	14.687	15.225	13.680	16.559
	LYW	13.235	15.194	**13.191**	13.425	21.090	16.527	17.020	17.298	19.355
	HYE	**11.460**	12.955	11.614	11.964	21.059	14.046	14.406	14.194	13.640
	CSS	**24.528**	30.245	24.951	26.393	34.952	26.764	29.086	25.380	28.883
MAE## (min)	CZW	4.447	**4.250**	4.589	4.373	4.809	4.879	4.392	5.050	5.235
	LYW	**5.975**	6.016	6.364	6.129	6.359	6.476	6.214	6.421	6.562
	HYE	5.626	**5.514**	5.805	5.606	5.990	5.850	6.094	5.896	6.052
	CSS	**4.471**	4.801	5.106	4.986	4.930	4.915	5.427	5.015	5.258
MAPE## (%)	CZW	8.505	**8.245**	8.751	8.493	9.296	9.397	8.582	9.733	9.989
	LYW	**10.465**	10.690	11.017	10.783	11.197	11.512	10.921	11.282	11.868
	HYE	8.958	8.841	9.233	**8.822**	9.648	9.352	9.973	9.526	9.720
	CSS	**7.455**	8.044	8.418	8.450	8.332	8.204	8.895	8.327	8.811

The bold fonts represent the best results, and the metrics followed by one and two octothorpe(s) represent errors calculating on short and long delays, respectively.

Table 11 Three-step predictive errors of models at important stations on W-G HSR.

Metrics	Station	FCLL-Net	SAE+FCNN	FCL-IT	FCL-IS	DBN	MLP	RF	DELM	SVR
MAE# (min)	CZW	**1.433**	1.703	1.452	1.495	1.908	1.743	1.816	1.671	1.865
	LYW	**1.256**	1.419	1.274	1.322	1.764	1.621	1.663	1.539	1.634
	HYE	**1.554**	1.808	1.576	1.678	2.092	1.874	1.807	1.807	1.923
	CSS	3.048	3.580	**2.975**	3.395	3.951	3.467	3.457	3.071	3.414
MAPE# (%)	CZW	**16.615**	20.376	16.846	17.397	23.048	20.710	22.020	19.960	22.612
	LYW	**14.309**	16.545	14.596	15.146	21.588	19.676	20.151	18.529	19.427
	HYE	**17.383**	20.736	17.726	18.566	24.513	21.683	20.669	20.903	22.102
	CSS	30.323	35.801	**29.603**	33.170	41.058	35.440	34.866	30.940	34.580
MAE## (min)	CZW	9.951	**9.711**	10.222	10.012	10.028	10.355	9.588	10.095	10.497
	LYW	7.577	7.384	7.630	7.646	8.064	8.346	**7.292**	7.958	8.315
	HYE	10.193	**10.062**	10.724	10.511	10.296	10.896	10.132	10.506	11.002
	CSS	**5.619**	6.086	6.137	6.464	6.422	6.813	6.570	6.101	7.182
MAPE## (%)	CZW	18.320	**17.889**	18.743	18.417	18.572	18.946	17.817	18.592	19.056
	LYW	13.689	**13.409**	13.702	13.808	14.510	14.812	13.309	14.513	14.991
	HYE	17.164	**17.000**	18.024	17.817	17.293	18.251	16.942	17.799	18.478
	CSS	**9.701**	10.583	10.361	11.214	11.268	11.682	11.433	10.560	12.188

The bold fonts represent the best results, and the metrics followed by one and two octothorpe(s) represent errors calculating on short and long delays, respectively.

Table 12 Two-step predictive errors of models at important stations on X-S HSR.

Metrics	Station	FCLL-Net	SAE+FCNN	FCL-IT	FCL-IS	DBN	MLP	RF	DELM	SVR
MAE# (min)	SW	1.570	1.643	1.566	**1.560**	2.447	2.988	2.212	1.755	1.751
	PN	**1.694**	2.152	1.804	1.866	2.382	2.135	2.152	1.854	2.129
	CY	**2.210**	2.866	2.338	2.450	2.586	2.642	3.215	2.508	2.972
	CS	**2.387**	2.736	2.468	2.616	2.803	2.823	3.261	2.328	2.781
MAPE# (%)	SW	15.524	16.572	**15.516**	15.564	25.105	28.932	21.261	18.367	17.849
	PN	**18.934**	24.543	20.240	21.335	26.488	23.537	23.983	20.950	24.046
	CY	**24.235**	31.988	25.694	26.503	28.749	29.791	37.106	28.151	33.716
	CS	**23.585**	27.923	24.062	25.805	30.026	28.313	34.583	23.962	29.300
MAE## (min)	SW	8.882	**8.653**	9.818	8.967	10.875	19.147	8.856	9.939	9.444
	PN	**3.640**	4.407	3.950	4.010	4.576	4.505	4.681	5.024	6.823
	CY	**4.028**	4.536	4.423	4.379	4.930	5.057	4.166	5.228	5.300

	CS	4.929	**4.627**	6.393	5.237	5.260	5.657	5.067	5.451	4.996
MAPE$^{\#\#}$ (%)	SW	12.642	**12.390**	13.722	12.789	15.938	25.186	12.914	14.105	13.778
	PN	**4.855**	6.537	5.348	5.461	6.664	6.399	6.860	7.291	9.858
	CY	**6.154**	7.062	6.407	6.748	7.703	7.493	6.903	7.663	8.187
	CS	7.686	**7.358**	8.815	8.303	8.156	8.477	7.847	8.098	7.927

The bold fonts represent the best results, and the metrics followed by one and two octothorpe(s) represent errors calculating on short and long delays, respectively.

Table 13 Three-step predictive errors of models at important stations on X-S HSR.

Metrics	Station	FCLL-Net	SAE+FCNN	FCL-IT	FCL-IS	DBN	MLP	RF	DELM	SVR
MAE$^{\#}$ (min)	SW	**1.765**	1.806	1.784	1.858	2.729	3.666	2.500	2.043	1.938
	PN	**2.438**	3.092	2.504	2.733	3.421	3.197	3.177	2.802	3.083
	CY	**2.474**	3.050	2.566	2.702	3.308	3.090	3.658	2.884	3.277
	CS	3.635	4.262	**3.616**	4.164	4.348	4.135	4.938	3.665	4.243
MAPE$^{\#}$ (%)	SW	17.975	18.441	**17.854**	18.701	27.603	37.025	26.184	21.326	19.828
	PN	**26.251**	34.810	27.370	29.016	37.283	34.786	35.598	31.041	34.365
	CY	**26.824**	33.426	27.751	28.850	36.397	32.927	41.206	32.088	36.339
	CS	**33.033**	41.911	33.456	37.717	41.544	39.397	50.575	35.047	42.096
MAE$^{\#\#}$ (min)	SW	9.931	**9.779**	10.042	9.887	12.603	13.879	11.260	11.681	10.354
	PN	**8.169**	10.114	8.710	8.698	8.973	8.512	8.478	9.521	11.795
	CY	**5.400**	6.311	6.441	5.847	5.951	6.344	5.737	6.688	7.870
	CS	**7.153**	7.183	8.153	7.645	8.052	7.417	7.088	7.606	7.689
MAPE$^{\#\#}$ (%)	SW	14.783	**14.500**	14.721	14.777	18.688	19.224	17.194	16.422	15.304
	PN	**12.914**	16.130	13.822	14.028	14.328	13.180	13.470	14.772	17.933
	CY	**8.584**	10.062	9.035	9.440	9.513	9.969	9.360	10.333	12.413
	CS	**11.403**	11.665	11.997	12.370	13.490	11.850	11.734	12.208	12.368

The bold fonts represent the best results, and the metrics followed by one and two octothorpe(s) represent errors calculating on short and long delays, respectively.

5.3. Sensitivity analysis

To determine the degrees of impact of the model components and influencing factors that affect delay prediction accuracy, a break-down analysis of the model structure was conducted. In detail, this section attempts to determine the importance of the FCNN component and both LSTM components in FCLL-Net for prediction accuracy. As different components are fed with different influencing factors, and as the two LSTM components are used for different functions (i.e., FCNN is fed with non-operational features, LSTM component 1 is fed with operational factors to capture interactions between trains, and LSTM component 2 is fed operational features to capture interactions between stations), the importance of each component in FCLL-Net can represent the importance of the influencing factors. For example, the importance of LSTM components 1 and 2 can reveal the importance of interactions between trains and stations, respectively, which were the motivations used to formulate this study.

The impacts of the model components were investigated by training models with individual components being excluded one by one; for example, to analyze the impact of the FCNN component on delay prediction, the FCNN component was removed from FCLL-Net, and the non-operational features were removed from the training, validation, and testing datasets. The importance of model components was quantified by comparing their loss functions with that of FCLL-Net, as shown in Fig. 17. In this figure, “FCLL-Net” represents the proposed model, “LSTM 1” represents the model in which LSTM component 1 was removed from FCLL-Net, “LSTM 2” represents the model in which LSTM component 2 was removed from FCLL-Net, and “FCNN” represents the model in which the FCNN component was removed from FCLL-Net. It can be seen from the figure that removing any component will lead to an increase of the loss function. From the bar graph, it is evident that LSTM component 1 is the most important part, which means that interaction between trains is the most important factor. Additionally, the impact of interactions between stations is as important as the collaborative effects of the five non-operational features, as the importance of LSTM component 2 is similar to that of the FCNN component. These results confirm the motivation that the interactions in train operation are significant factors in train delay prediction.

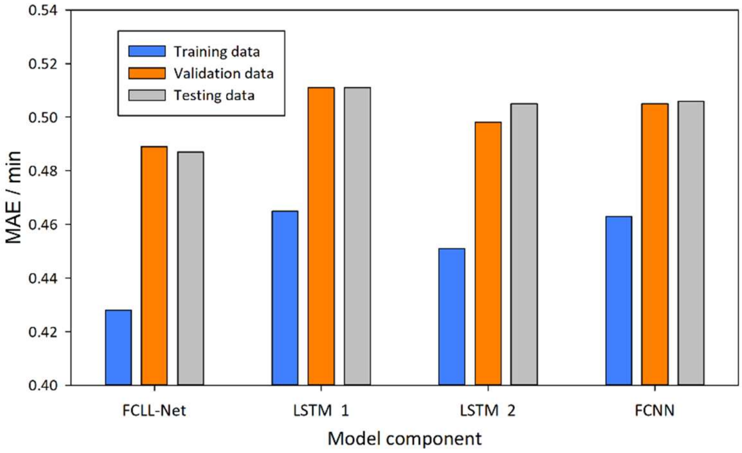


Fig. 17. Analysis of sensitivity to model components.

Finally, delays may propagate from the primary delay train to the succeeding trains due to the train interval and buffer time constraints, and delays can also propagate from the occurrence location to following stations that the delayed trains will pass through due to the train running time and time supplement constraints. In other words, train delays will simultaneously propagate between trains and stations. The proposed model integrates two LSTM components with sequential input to capture the delay propagation effects between trains and stations, respectively. Therefore, the break-down analysis of the proposed model can also reveal the influences of the delay propagation on future train delays. In Fig. 17, the model 'LSTM 1' just captures the delay propagation effects between stations, and the model 'LSTM 2' just captures the delay propagation effects between trains. By comparing the result of 'LSTM 1' and 'LSTM 2' models with that of FCLL-Net, the propagation effects between trains and stations can be determined, respectively. For example, on the training dataset, the MAE of FCLL-Net, 'LSTM 1', and 'LSTM 2' were 0.428, 0.460, and 0.451, respectively. This means that ignoring the delay propagation effects between trains and stations could increase on average 0.032 min (7.5%) and 0.023 min (5.4%) predictive errors on each train, respectively. The comparison results of models 'LSTM 1', 'LSTM 2', and FCLL-Net on training, validation, and testing dataset quantitatively show the propagation effects on future train delays.

6. Conclusions

In this study, an innovative train delay prediction model (FCLL-Net) that combines two distinctive neural network architectures, namely fully-connected neural network (FCNN) and long short-term memory (LSTM), was proposed to account for both non-operational and operational factors in the delay prediction process. One of the key features of the FCLL-Net model is that it captures a sequence of train operations and makes use of historical data to capture the cumulative interactions between trains and stations. The proposed model was applied in two real-world HSR lines in China to assess its predictive performance, generalizability, and computational requirements as against six widely used delay prediction models from the existing literature, namely the multilayer perceptron (MLP), deep extreme learning machine (DELM), random forest (RF), support vector regression (SVR), Bayesian network (BN), and Markov model (MM), and two state-of-the-art deep learning models, namely an LSTM-based model for capturing interactions between trains (FCL-IT) and another LSTM-based model for capturing interactions between stations (FCL-IS). A sensitivity analysis was also conducted to break-down the importance of model components and influencing factors. The main findings from these experiments are summarized as follows:

(1) In predicting the delays of a particular train, it is beneficial to consider the interactions between trains and stations in terms of prediction accuracy.

(2) Based on the commonly used prediction performance metrics, the proposed FCLL-Net model considerably outperforms other widely used delay prediction models.

(3) FCLL-Net was also shown to be superior to other state-of-the-art models in terms of generalizability or transferability. Further tests show the potential for the FCLL-Net model, once trained, to be applied to different HSR lines for delay prediction.

(4) Among the factors that affect train operation and thus delay, the interactions between trains and stations are at least as important as the non-operational features. Therefore, it is necessary to model train operation as sequences to capture these interactions.

Our future work will involve an analysis of train heterogeneities, such as time and space differences in timetables. To achieve this, the datasets will be divided into different samples in terms of stops, time periods, and stations. Our goal is to develop train delay propagation model applicable for general railway networks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Nature Science Foundation of China [grant numbers 71871188, U1834209], the Science & Technology Department of Sichuan Province [grant number 2018JY0567], the State Key Laboratory of Rail Traffic Control [grant number RCS2019K007], the Doctoral Innovation Fund Program of Southwest Jiaotong University [grant number D-CX201827], and the China Scholarship Council [grant number 201707000038]. We are grateful for the useful contributions made by our project partners.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., 2016. Tensorflow: A system for large-scale machine learning, *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265-283.

Akusok, A., Björk, K.-M., Miche, Y., Lendasse, A., 2015. High-performance extreme learning machines: a complete toolbox for big data applications. *IEEE Access* 3, 1011-1025.

Al-Ibrahim, A., 2010. *Dynamic Delay Management at Railways. Semi-Markovian Decision Approach*. Rozenberg Publishers.

Büker, T., Seybold, B., 2012. Stochastic modelling of delay propagation in large networks. *Journal of Rail Transport Planning & Management* 2(1-2), 34-50.

Barbour, W., Mori, J.C.M., Kuppa, S., Work, D.B., 2018. Prediction of arrival times of freight traffic on us railroads using support vector regression. *Transportation Research Part C: Emerging Technologies* 93, 211-227.

Barta, J., Rizzoli, A.E., Salani, M., Gambardella, L.M., 2012. Statistical modelling of delays in a rail freight transportation network, *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, p. 286.

Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., 2007. Greedy layer-wise training of deep networks, *Advances in neural information processing systems*, pp. 153-160.

Berger, A., Gebhardt, A., Müller-Hannemann, M., Ostrowski, M., 2011. Stochastic delay prediction in large train networks, *OASIcs-OpenAccess Series in Informatics*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Briggs, K., Beck, C., 2007. Modelling train delays with q-exponential functions. *Physica A: Statistical Mechanics and its Applications* 378(2), 498-504.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J., 2014. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological* 63, 15-37.

Carey, M., Carville, S., 2000. Testing schedule performance and reliability for train stations. *J Oper Res Soc* 51(6), 666-682.

Carey, M., Kwieciński, A., 1994. Stochastic approximation to the effects of headways on knock-on delays of trains. *Transportation Research Part B: Methodological* 28(4), 251-267.

Cerreto, F., Nielsen, B.F., Nielsen, O.A., Harrod, S.S., 2018. Application of data clustering to railway delay pattern recognition. *Journal of Advanced Transportation* 2018.

Cerreto, F., Nielsen, O.A., Harrod, S., Nielsen, B.F., 2016. Causal Analysis of Railway Running Delays, *11th World Congress on Railway Research (WCRR 2016)*.

Chen, D., Wang, L., Li, L., 2015. Position computation models for high-speed train based on support vector machine approach. *Applied Soft Computing* 30, 758-766.

Chollet, F., 2015. Keras: Deep learning library for theano and tensorflow. *URL: https://keras. io/k* 7(8), T1.

Corman, F., D'Ariano, A., Hansen, I.A., 2014. Evaluating disturbance robustness of railway schedules. *Journal of Intelligent Transportation Systems* 18(1), 106-120.

Corman, F., Kecman, P., 2018. Stochastic prediction of train delays in real-time using Bayesian networks. *Transportation Research Part C: Emerging Technologies* 95, 599-615.

Gaurav, R., Srivastava, B., 2018. Estimating Train Delays in a Large Rail Network Using a Zero Shot Markov Model, *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1221-1226.

Ghofrani, F., He, Q., Goverde, R.M., Liu, X., 2018. Recent applications of big data analytics in railway transportation systems: A survey. *Transportation Research Part C: Emerging Technologies* 90, 226-246.

Gorman, M.F., 2009. Statistical estimation of railroad congestion delay. *Transportation Research Part E: Logistics and Transportation Review* 45(3), 446-456.

Goverde, R.M., 2010. A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C: Emerging Technologies* 18(3), 269-287.

Goverde, R.M., Corman, F., D'Ariano, A., 2013. Railway line capacity consumption of different railway signalling systems under scheduled and disturbed conditions. *Journal of Rail Transport Planning & Management* 3(3), 78-94.

Goverde, R.M., Hansen, I., Hooghiemstra, G., Lopuhaa, H., 2001. Delay distributions in railway stations, *9th World Conference on Transport Research, Seoul, Korea, July 22-27, 2001*. Citeseer.

Graves, A., Mohamed, A.-r., Hinton, G., 2013. Speech recognition with deep recurrent neural networks, *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, pp. 6645-6649.

Grossberg, S., 2013. Recurrent neural networks. *Scholarpedia* 8(2), 1888.

Haahr, J.T., Hellsten, E.O., van der Hurk, E., 2019. Train Delay Prediction in the Netherlands through Neural Networks.

Hansen, I.A., Goverde, R.M., van der Meer, D.J., 2010. Online train delay recognition and running time prediction, *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, pp. 1783-1788.

Harris, M., 2006. Analysis and modelling of train delay data. MSc Thesis, University of York, UK.

Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18(7), 1527-1554.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786), 504-507.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation* 9(8), 1735-1780.

Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2006. Extreme learning machine: theory and applications. *Neurocomputing* 70(1-3), 489-501.

Huang, P., Javad, L., Wen, C., Peng, Q., Fu, L., Li, L., Xu, X., 2020a. A Bayesian network model to predict the effects of interruptions on train operations. Transpor. Res. Part C: Emerging Techn. 114, 338–358.

Huang, P., Wen, C., Fu, L., Peng, Q., Li, Z., 2020b. A hybrid model to improve the train running time prediction ability during high-speed railway disruptions. Saf. Sci. 122, 104510. https://doi.org/10.1016/j.ssci.2019.104510.

Huang, P., Wen, C., Fu, L., Peng, Q., Tang, Y., 2020c. A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems. Inf. Sci. 516, 234–253.

Huang, P., Wen, C., Peng, Q., Jiang, C., Yang, Y., Fu, Z., 2019. Modeling the Influence of Disturbances in High-Speed Railway Systems. *Journal of Advanced Transportation* 2019, https://doi.org/10.1155/2019/8639589.

Huisman, T., Boucherie, R.J., 2001. Running times on railway sections with heterogeneous train traffic. *Transportation Research Part B: Methodological* 35(3), 271-292.

Jiang, C., Huang, P., Lessan, J., Fu, L., Wen, C., 2018. Forecasting primary delay recovery of high-speed railway using multiple linear regression, supporting vector machine, artificial neural network, and random forest regression. *Canadian Journal of Civil Engineering* 46(5), 353-363.

Kecman, P., Corman, F., Meng, L., 2015. Train delay evolution as a stochastic process, *6th International Conference on Railway Operations Modelling and Analysis-RailTokyo2015*.

Kecman, P., Goverde, R.M., 2015a. Online data-driven adaptive prediction of train event times. *IEEE Transactions on Intelligent Transportation Systems* 16(1), 465-474.

Kecman, P., Goverde, R.M., 2015b. Predictive modelling of running and dwell times in railway traffic. *Public Transport* 7(3), 295-319.

Khadilkar, H., 2016. Data-enabled stochastic modeling for evaluating schedule robustness of railway networks. *Transportation Science* 51(4), 1161-1176.

Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lessan, J., Fu, L., Wen, C., 2019. A hybrid Bayesian network model for predicting delays in train operations. *Computers & Industrial Engineering* 127, 1214-1222.

Lessan, J., Fu, L., Wen, C., Huang, P., Jiang, C., 2018. Stochastic model of train running time and arrival delay: a case study of wuhan–guangzhou high-speed rail. *Transportation Research Record* 2672(10), 215-223.

Li, D., Daamen, W., Goverde, R.M., 2016. Estimation of train dwell time at short stops based on track occupation event data: A study at a Dutch railway station. *Journal of Advanced Transportation* 50(5), 877-896.

Liaw, A., Wiener, M., 2002. Classification and regression by randomForest. *R news* 2(3), 18-22.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y., 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17(4), 818.

Malavasi, G., Ricci, S., 2001. Simulation of stochastic elements in railway systems using self-learning processes. *European Journal of Operational Research* 131(2), 262-272.

Marković, N., Milinković, S., Tikhonov, K.S., Schonfeld, P., 2015. Analyzing passenger train arrival delays with support vector regression. *Transportation Research Part C: Emerging Technologies* 56, 251-262.

Meester, L.E., Muns, S., 2007. Stochastic delay propagation in railway networks and phase-type distributions. *Transportation Research Part B: Methodological* 41(2), 218-230.

Milinković, S., Marković, M., Vesković, S., Ivić, M., Pavlović, N., 2013. A fuzzy Petri net model to estimate train delays. *Simulation Modelling Practice and Theory* 33, 144-157.

Murali, P., Dessouky, M., Ordóñez, F., Palmer, K., 2010. A delay estimation technique for single and double-track railroads. *Transportation Research Part E: Logistics and Transportation Review* 46(4), 483-495.

Nabian, M.A., Alemazkoor, N., Meidani, H., 2019. Predicting Near-Term Train Schedule Performance and Delay Using Bi-Level Random Forests. *Transportation Research Record*, 0361198119840339.

Nair, R., Hoang, T.L., Laumanns, M., Chen, B., Cogill, R., Szabó, J., Walter, T., 2019. An ensemble prediction model for train delays. *Transportation Research Part C: Emerging Technologies* 104, 196-209.

Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., Muharemagic, E., 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2(1), 1.

Olah, C., 2015. Understanding lstm networks. *GITHUB blog, posted on August* 27, 2015.

Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., Anguita, D., 2017. Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47(10), 2754-2767.

Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., Anguita, D., 2018. Train delay prediction systems: a big data analytics perspective. *Big data research* 11, 54-64.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12(Oct), 2825-2830.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5(3), 1.

Şahin, İ., 2017. Markov chain model for delay distribution in train schedules: Assessing the effectiveness of time allowances. *Journal of Rail Transport Planning & Management*.

Sak, H., Senior, A., Beaufays, F., 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling, *Fifteenth annual conference of the international speech communication association*.

Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. *Statistics and computing* 14(3), 199-222.

Svozil, D., Kvasnicka, V., Pospichal, J., 1997. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems* 39(1), 43-62.

Van der Meer, D.J., Goverde, R.M., Hansen, I.A., 2009. Prediction of train running times using historical track occupation data, *WCTR*. Delft University of Technology, pp. 1-62.

Wallander, J., Mäkitalo, M., 2012. Data mining in rail transport delay chain analysis. *International Journal of Shipping and Transport Logistics* 4(3), 269-285.

Wang, S.L., Ma, J.H., 2013. Variable-Carriage Vehicle Scheduling Based on Segment Point. *2013 International Conference on Computer Science And Artificial Intelligence (Iccsai 2013)*, 16-20.

Wang, Y., Zhang, D., Liu, Y., Dai, B., Lee, L.H., 2019. Enhancing transportation systems via deep learning: A survey. *Transportation research part C: emerging technologies* 99, 144-163.

Wen, C., Huang, P., Li, Z., Lessan, J., Fu, L., Jiang, C., Xu, X., 2019. Train dispatching management with data-driven approaches: a comprehensive review and appraisal. *IEEE Access* 7, 114547-114571.

Wen, C., Li, Z., Lessan, J., Fu, L., Huang, P., Jiang, C., 2017. Statistical investigation on train primary delay based on real records: evidence from Wuhan–Guangzhou HSR. *International Journal of Rail Transportation* 5(3), 170-189.

Wen, C., Peng, Q., Chen, Y., Ren, J., 2014. Modelling the running states of high-speed trains using triangular fuzzy number workflow nets. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 228(4), 422-430.

Yaghini, M., Khoshraftar, M.M., Seyedabadi, M., 2013. Railway passenger train delay prediction via neural network model. *Journal of advanced transportation* 47(3), 355-368.

Yang, Y., Huang, P., Peng, Q., Jie, L., Wen, C., 2019. Statistical delay distribution analysis on high-speed railway trains. *Journal of Modern Transportation*, 1-10.

Yuan, J., Goverde, R., Hansen, I., 2002. Propagation of train delays in stations. *WIT Transactions on The Built Environment* 61.

Yuan, J., Hansen, I.A., 2007. Optimizing capacity utilization of stations by estimating knock-on train delays. *Transportation Research Part B: Methodological* 41(2), 202-217.

Zilko, A.A., Kurowicka, D., Goverde, R.M., 2016. Modeling railway disruption lengths with Copula Bayesian Networks. *Transportation Research Part C: Emerging Technologies* 68, 350-368.