Enhancing the Understanding of Train Delays With Delay Evolution Pattern Discovery: A Clustering and Bayesian Network Approach

Article *in* IEEE Transactions on Intelligent Transportation Systems · January 2022

DOI: 10.1109/TITS.2022.3140386

CITATIONS
18

READS
455

3 authors, including:

Ping Huang
Southwest Jiaotong University
47 PUBLICATIONS **770** CITATIONS
SEE PROFILE

Francesco Corman
179 PUBLICATIONS **4,373** CITATIONS
SEE PROFILE

Enhancing the Understanding of Train Delays with Delay Evolution Pattern Discovery: A Clustering and Bayesian Network Approach

Ping Huang, Thomas Spanninger, Francesco Corman

***Abstract*—Train delay evolutions exhibit different patterns (i.e., increasing delays, decreasing delays, or unchanged delays), because of the effects of stochastic disturbances and pre-scheduled supplement/recovery times. The dynamics and uncertainty of the train delay evolution make train delay prediction a challenging task. This study presents a hybrid framework, called context-driven Bayesian network (CDBN), composed of a delay evolution pattern discovery model, i.e., a K-Means clustering approach, and a train delay prediction model, i.e., Bayesian network (BN), to address this problem. The clustering algorithm is used to uncover the delay evolution patterns, and classify the data into different categories, based on the delay jumps, i.e., the change of a delay from one station to a consequent station. The BN model, which considers the delays in previous stations to overcome the Markov property assumption, is used as the predictive model of train delays. The data in each category (classified by the clustering model) are used to train and test the BN model separately. We evaluated the BN model, the clustering algorithm, and the CDBN model, by comparing against their counterparts, respectively. The results show that: (1) the proposed BN structure has advantages over the common delay prediction models built on Markov property; (2) the clustering is effective, and it can extensively improve the accuracy of the predictive model; and (3) the CDBN outperforms the existing delay prediction models in wide usability, because of its more profound understanding of the delay evolution patterns.**

***Index Terms*—Train delays, delay evolution, pattern discovery, K-Means clustering, Bayesian network.**

I. Introduction

Railway systems are complex modes of transportation, which comprises several subsystems operating under various operational rules, and are subject to many interrelated and unforeseeable operation-interrupting events. Countless human- or equipment-related faults, failures, or anomalies can cause disturbances, leading to train delays in train operations [1]. Additionally, due to the scarce infrastructure of railway traffic and the high utilization of the network capacity, any small train delays easily spread throughout the network by propagating in time and space. Therefore, any train delay, if it is not well addressed, will not only intensively lower the efficiency of train operation, and decrease the benefits of the railway controllers, but lengthen passenger's travel time and increase their dissatisfactions. Therefore, once a delay happens, railway controllers are expected to take quick actions to reduce its impact [2], e.g., recovering the delays to stop it from influencing other trains, which is a critical step for intelligent transportation systems (ITS). In real-time decision-making, the dispatchers (controllers) not only need to consider the current train delays, but to estimate the future train delays to make comprehensive decisions. Such estimation is typically made based on the experience of the dispatchers. Although the dispatchers are experienced and try hard to make predictions, it is quite difficult for them to figure out the optima (e.g., the most accurate prediction) considering sequential actions. Further, train operations are complex processes that are influenced by many factors inside or outside the railway systems [3], and train operations present dynamics because of the changing of states over time and space [4], and the randomness of the disturbances and the uncertainty of the pre-scheduled supplement/recovery times [5]. It is thus a challenging task for dispatchers to accurately estimate the future train delays based on experience, while considering such influencing factors. Therefore, accurate and advanced traffic prediction models (e.g., delay prediction model, running time prediction model, and dwelling time prediction model) and pattern discovery models act as important supporting information for the controllers' decision making in the real-time decision-making process [6, 7].

Graph- and network-based models have been proposed for train operation prediction in wide usability and extensibility because of their high interpretation to train operations [8]. However, the graph- and network-based models are generally built on Markov property (any state is only dependent on its previous latest state), which means that any train delay is only influenced by its previous state. Although this simplification can enable reasoning and computations of the model, it also makes the models incapable of capturing the dynamics of the delay evolution at multiple stations, in delay prediction problems. Previous multiple states can be considered in the BN model, but it just captures the individual influences of the previous states on the current state, instead of the trends/dynamics embedded in the previous N states.

To address this drawback, we propose a hybrid framework, named context-driven Bayesian network (CDBN), which introduces a clustering algorithm for delay evolution pattern discovery to improve the train delay prediction accuracy of the Bayesian network model. The improvements of a train delay prediction model enhanced by a deep understanding of train delay evolution patterns have never been considered in existing studies. Therefore, the novelty of this paper lies in the combination of the K-means clustering algorithm and the BN model to improve the predictability of delay jumps in train

This work is supported by the Swiss National Science Foundation, Switzerland under the research project DADA, grant no. 181210. We are grateful for the useful contributions made by our project partners.

P. Huang is with the Institute for Transport Planning and Systems, ETH Zurich, Zurich 8093, Switzerland, and National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Southwest Jiaotong University, Chengdu, 610031, China; Thomas Spanninger and Francesco Corman are with the Institute for Transport Planning and Systems, ETH Zurich, Zurich 8093, Switzerland (e-mail: ping.huang@ivt.baug.ethz.ch; Thomas.Spanninger@ivt.baug.ethz.ch; francesco.corman@ivt.baug.ethz.ch;).

delay predictions. The contribution of the present study is four-fold:

(1) Quantifying the train delays interactions/dependence on a single train from a data-driven perspective (i.e., each delay is primarily influenced by how many previous delays?);
(2) Proposing a BN model that overcomes the limitation (Markov assumption) of common graph- and network-based models in train operation modeling;
(3) Using a clustering technique to deal with the dynamics within multiple past train delays to enhance the understanding of delay evolution patterns, thus improving the predictability of the delay jumps.
(4) Evaluating the proposed model from three perspectives: (a) showing the advantage of the proposed BN structure over other standard graph and network models; (b) showing the advantage of using clustering technique; and (c) showing the advantage of the CDBN model over other state-of-the-art machine learning models.

The remainder of the study is structured as follows. In Section II, we review the literature. In Section III, the problem is explained, and the proposed model is introduced. In Section IV, the implementation of the proposed model is demonstrated. In Section V, case studies with the Chinese and Dutch railway operation data are conducted. Finally, the present study is concluded and discussed in Section VI.

II. Literature review

The train operation prediction methods include graph- or network-based models, and machine learning models. Graph- or network-based models are mainly for predicting the future train delays based on the reasoning of the known train delay evolution patterns; whereas machine learning models are mostly employed to predict train delays, train running times, and train dwelling times based historic observations.

As train operations are composed of events, such as arrival and departure events, graph- and network-based models that can describe the train events are usually used to describe the train operations. The critical reason why the graph and network models have been widely used in both academic and practice for train operation lies in its high interpretability. The proposed methods include timed event graph, activity graph, and alternative graph [9-14]. These models are developed by graphically representing the train events, using the nodes of the graph where the relationships of train events are usually expressed by joint probability tables or density functions. These graph models are the earliest and widely used delay prediction models, because of their high interpretability to train operations. Recently, Bayesian networks have been applied for train delay prediction purposes. In general, there are two types of Bayesian network structure for train delay prediction. The first one takes the arrival and departure events of each train at each station and treats them as separate nodes of a Bayesian network architecture [15]. The other takes the arrival and departure events of two consecutive trains as the nodes of a Bayesian network [16]. Further, BN model has also been used in other problems to provide the decision support in railway dispatching, e.g., the prediction of the effects (i.e., the primary delays, the number of delayed trains, and the total delay times) of interruptions [17], the impacts of delays at stops on the network [18], and the duration of disruptions [19]. In addition, the train operations are treated as a Markov process, and a conditional Bayesian model was proposed to predict the train delay propagation in large railway networks [20]. Further, as an extension of BN, deep belief networks (DBN), have also been innovatively used for transportation network assignment optimization. The DBN was built to preprocess the data to improve the clustering effect of the K-Means clustering [21]. Markov model, which treats train operation as a chain, is also a widely used graph-based model in train delay predictions. However, because the Markov model is a chain rather than a network, it can be regarded as the simplest graph- and network-based models. The used Markov models in train delay prediction include one-order Markov model [22-24], i.e., only depending on previous one state, and N-order Markov model [25], i.e., depending on previous N states. Further, in [26], the authors established both homogeneous and non-homogeneous Markov models, considering the variability of train operations in stations and sections, to predict train delays and evaluate timetable robustness. The graph- and network-based models can well interpret the train operation process. Although the N-order Markov model considered the previous N states, it could only capture the individual influences of the previous states on the current state, instead of the changing trends/dynamics embedded in the previous N states. Because of the high uncertainty of train operations and the Markov property assumption of the models, the graph- and network-based models are incapable of improving the predictive accuracy through capturing the delay evolution patterns.

Other alternative approaches for train operation modeling are machine learning and data mining models, including unsupervised learning approaches and supervised learning. Unsupervised models, with only input features but without observed labels/targets, aim to discover the train delay patterns or train delay evolution patterns, e.g., the association analysis for delay relationship discovery [27], clustering algorithm for primary delay classifying [28], and train delay pattern discovery [29]. Supervised learning, with both input and output (labels or target), learn the knowledge and rules between the input and output features, and then uses these rules and knowledge to predict the future data. The supervised learning model can be used for train delay prediction, delay recovery prediction, running time prediction, train delay evolution pattern recognition, etc. Train delay prediction is the most widely studied aspect among train operation management problems [8]. The most widely used supervised learning model for train delay prediction is neural networks, including multi-layer perceptron [30-32], long short-term memory for train delay prediction [5, 33], convolutional neural network model [34], and graph convolution network [35]. In addition, as a variant of neural networks, extreme learning machine has also been extensively used in train delay predictions [36-39]. Other types of supervised models used in delay prediction include support

vector machine [3, 40], decision tree-based model [41-44], ensemble learning [45, 46], and the combination of multiple state-of-the-art neural networks [5, 47]. Other supervised learning methods used for train operation management and control include a support vector machine for train position [48], a decision tree model for delay recovery prediction [49], transfer learning and ensemble learning for delay jumps predictions [50], and a hybrid model (support vector machine and Kalman filter) for train running time prediction [51].

TABLE I. A SUMMARY OF GRAPH/NETWORK MODELS AND CLUSTERING MODELS.

Literature	Method	Research problem
[9-11]	Timed event graph	Delay prediction/propagation
[12]	Alternative graph	Timetable robustness
[13]	Activity graph	Delay propagation
[14]	Petri net	Delay prediction
[15-18]	Bayesian networks	Delay prediction
[19]	Bayesian networks	Duration of disruptions
[20]	Bayesian theory	Delay propagation
[22-26]	Markov model	Delay prediction
[21]	Clustering	Transport network assignment
[28]	Clustering	Disturbance clustering
[29]	Clustering	Delay pattern discovery

A review of the existing literature on graph and clustering models (a summary shown in Table I) reveals that researchers focused on applying a variety of graph and network methods built on Markov property for train operation prediction, or applying the clustering technique on pattern discovery. However, no study focuses on combing the two methods to solve the delay or delay jump prediction problems in train operations. Further, the existing studies failed to consider the past delay evolution patterns in train delay predictions (e.g., is it increasing, decreasing, or unchanged?), making them incapable of addressing delay jumps. Therefore, we innovatively combine the two techniques, namely clustering and BN model, in train delay prediction to address these problems. We take the past train delay evolution patterns into consideration to enhance the understanding of the prediction model on train delay evolutions. The predictive models are built on the (clustered) data with different patterns, which, therefore, is expected to provide more accurate predictions of train delays as support information for real-time train dispatching problems.

III. Problem Statement and Method

A. Problem statement

Let us assume three trains, Train 1, Train 2, and Train 3, respectively, with t_1, t_2, and t_3 min delay at station S_1 ($t_1 < t_2 < t_3$), operating from station S_1 to station S_3. Train 1 operates during a disruption (e.g., a break-down of other trains at station), and its delay is increasing from station S_1 to S_2 and S_3 due to track occupancy. Train 2 operates under a conflict-free situation but its delay cannot be recovered due to the minimal headway with its preceding train in these sections and the minimal scheduled dwelling times at stations S_1 to S_3. Therefore, the delays of Train 2 barely change in the following operations. Train 3 not only operates during a disruption, but its operation is totally conflict-free, and its delay can be absorbed by the supplement/recovery times prescheduled in the following sections and stations. All in all, these three trains have the same delay time at station S_2, as shown in Fig. 1(a). If we intend to predict the delays of these three trains at S_3, based on the train operation information (e.g., delay time) at S_2, which introduces a drawback for the existing graph- and network-based models, because the graph- and network-based predictive models were usually built on the assumption of the Markov property (i.e., the delays at S_3 are only dependent on the delays at S_2) [8]. The Markov property assumption makes the predictive model incapable of distinguishing the delay evolution patterns (i.e., is it increasing, decreasing, or unchanged delay?), leading to unsatisfactory performance of the models on delay jumps. Due to the dynamics of train operation over time, train delays are usually long-term self-related (not just dependent on the previous one state). This means that train delay prediction requires advanced models that can detect the dynamics of train delays over time. Actually, studies that were based on graphs or networks for train delay prediction, e.g., timed event graph [9-11], Markov model [22-25], and Bayesian network [15-18], all simplify the train operation as a Markov process. To address this problem, we proposed a delay prediction framework, which aims to mine the delay evolution patterns first, and then, use the delays with different evolution patterns to separately train a delay prediction model. Take the railway line in Fig.1(b) as an example. In this figure, 1, 2, …, S, S+1 represent train stations, and T_s^a and T_s^d represent the arrival delays and departure delays in station s, respectively. Station S is the current station (i.e., the train location), and the arrival delays in station S+1 (hereafter, we call it target station) are the to-be-predicted delays. The proposed model in the present study aims first to cluster the trains based on their delay changes in the previous sections, and then predict the train delays with the clustered data. The delay evolution pattern from station S to station S+1 is in fact assumed to depend on the delay evolution patterns in the previous sections (e.g., station S-1 to S). The proposed delay prediction framework is thus promising to improve the accuracy of train delay prediction in railway systems because of the training on data with different patterns.

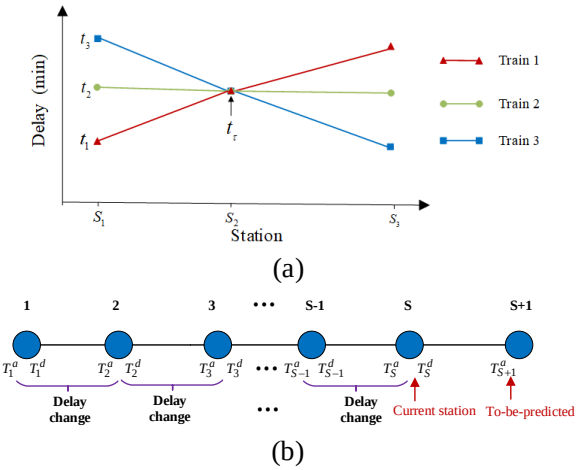


Fig. 1. Three different train delay evolution patterns (a), and the train delay prediction problem considering evolution patterns on a railway line (b).

B. The proposed method

The proposed method, to address the delay prediction problem (shown in Fig. 1), is shown in Fig. 2. The proposed method contains five steps, namely, the determination of train interactions/dependences, the determination of input for clustering model, pattern discovery, delay prediction model implementation, and model validation. First, with the train operation data, the delay interactions/dependences among a train (i.e., each delay is influenced by which delays) will be determined; Then, indicators for the delay evolution patterns will be determined, and these indicators will be the input of the clustering model to recognize the delay evolution patterns; Next, based on the determined indicators, a K-Means clustering model will be established to uncover the delay evolution patterns; In this process, train delays with different evolution patterns will be divided into the same cluster. Next, we use the data in each cluster to train/update an individual BN model, respectively, resulting in K (the number of clusters) BN models. Finally, the performance result of the proposed model will be determined by comparing its result with benchmark models. The pseudo code of the proposed CDBN model is shown in **Appendix A**. The proposed model, which is separately built on clustered delays, is expected to have deeper understanding of the delay evolution patterns, and perform better in train delay prediction tasks.

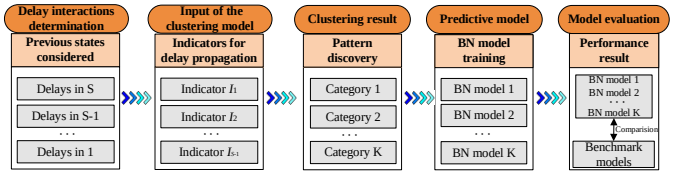


Fig. 2. Procedure of the proposed model (CDBN).

C. K-Means clustering

The K-Means algorithm belongs to unsupervised learning for data mining, which has strong robustness on high-dimensional and multi-collinear datasets [52]; The main argument for the K-Means algorithm is that it has shown outperformance over other common clustering algorithms (e.g., spectral clustering, agglomerative clustering, and BIRCH clustering) on train delay data in a previous study [28]. Further, the main novelty of our contribution lies in using delay evolution pattern discoveries to improve the performance of the Bayesian network model in delay prediction, not the method used.

The K-Means method can classify the data objects with high similarity into the same cluster, and the data objects with high dissimilarity into different clusters. It is based on the partition perspective, which takes distance as the metric of similarity measurement between data objects, i.e., the smaller the distance between data objects, the higher their similarity, and the more likely they are in the same cluster. The algorithm uses squared Euclidean distance to describe the distance between data objects [52]. Given a set of observations $D = \{x_1, x_2, \cdots, x_N\}$, it tries to minimize the within-cluster sum of squares:

$$\arg\min_{C} \sum_{k=1}^{K} \sum_{\vec{x}_i \in C_k} \| x_i - \boldsymbol{u_k} \|^2 \quad (1)$$

where $u_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$ is the mean of cluster C_k. The K-Means algorithm uses a heuristic iterative method to find the optima of Eq. (1), because it is a N-P hard problem. The heuristic iterative method is: (1) randomly select K points as the center of the corresponding cluster; (2) calculate the distance from all the points in the sample to the K centers, respectively; and (3) mark each sample with cluster whose center has the smallest distance to the sample. After each round of iteration, the new center of each cluster can be calculated, and the algorithm can converge, by repeating the above steps [52]. The pseudo code of the K-Means algorithm is shown in **Appendix B**.

D. Bayesian networks (BN)

Bayesian network (BN), a kind of directed acyclic graph (DAG), is a probabilistic model that describes the dependencies among a set of random variables [53]. In the BN, each node represents a random variable, and the arcs between each pair of nodes represent the probabilistic dependence between the variables, which is usually expressed by conditional probability tables for discrete variables and conditional probability distribution functions for continuous variables. An arc from variable V_i to V_j means that variable V_i impacts V_j; here, V_i is called the parent node of V_j, and V_j is called the child node of V_i. If there is a directed path from a node V_m to V_n, node V_m is called the ancestor of node V_n, and node V_n is called the descendant of node V_m. On the contrary, if there is no direct arc connection and not a single connection path between two nodes, there is no dependency relationship between these two variables, i.e., they are independent of each other.

BNs are built on the local Markov property, which means that a node is conditionally independent of its non-descendants given its parents. This property can simplify the joint distribution of the nodes of the networks, which means that the joint distribution for a BN can be expressed as P(node | parent (node)). Let us consider N random variables X_1, X_2, …, X_N in a BN. If we let X_n be a node in the graph, and *non* X_n be any set of nodes that are non-descendant of X_n, and *pa* X_n be a set of the immediate parents of X_n, then *non* X_n is conditionally independent of X_n, and the conditional relationship of the variables can be written as:

$$P(x_n \mid \text{non}\, x_n, \text{pa}\, x_n) = P(x_n \mid \text{pa}\, x_n) \quad (2)$$

where P is the conditional probability table or conditional probability distribution function. Therefore, by simplifying, the joint conditional probability distribution of the whole network can be written as:

$$P(x_1, x_2, ..., x_N) = \prod_{n=1}^{N} P(x_n \mid \text{pa}\, x_n) \quad (3)$$

The BN model will be used to predict the train delays in this study.

IV. Model Implementation

In this section, introduce the implementation of the proposed model, including the establishment and evaluation of both clustering model and predictive BN model.

A. Input of the clustering model

As mentioned before, train delays have different evolution patterns (increasing/decreasing/unchanged) due to the disturbances and pre-scheduled recovery times. Therefore, a clustering model will be first applied into the train delay data. Generally, train delays and train delay jumps (the changes of a delay from one station to next station) can both be used as the input of clustering model to uncover the delay evolution patterns. However, if we use the known train delays as the input for clustering, the clustering algorithm will intuitively classify the delays with different durations into different groups, leading to the dissimilarity of the distribution of the data in each cluster. In this regard, delay jumps are less dependent on delay durations, because the randomness of the disturbances and all delay durations can be recovered through prescheduled supplement/recovery times. Therefore, delay jumps have better representation the delay evolution patterns, and thus being selected as the indicators of delay evolution patterns (input of the clustering model).

In addition, unexpected disturbances and buffer/recovery time utilizations result in train delay changes from one station to another. Therefore, the prediction of of delay jumps is very challenging. Researchers have made a great effort to solve this problem [50, 51]. In our study, we use delay changes in the previous sections as input of the clustering model for delay evolution pattern discoveries. This means that the clustering model is supposed to uncover the cascading effects of delay changes in different sections, making the model capable of addressing the uncertainty of train delay jumps. This is a significant advantage over the existing studies that do not explicitly consider the delay evolution patterns.

We use train delay changes in sections as the input of the clustering model, since we focus on predicting the arrival delays of trains in stations (the prediction horizon is a section, i.e., aiming at predicting the arrival delays in the next station). In addition, supplement/recovery times are usually pre-scheduled in sections to recover train delays [54], in proportion to the train running times or distance. This means that any train delay can increase, decrease, or keep level in sections, because of the effect of disturbances and pre-scheduled supplement/recovery times. To incorporate the delay jumps in previous sections, we use the change values between the arrival and departure delays between the previous stations to identify the train evolution patterns. The calculations of the indicators between any two adjacent stations are as shown in Eqs. (4) and (5). Here, we assume station S is the current station, and the arrival delays in station S+1 are the to-be-predicted delays, as shown in Fig. 1(b)

$$I_n^a = T_s^a - T_{s-1}^a, s \in \{1,2,...,S\}; \quad (4)$$

$$I_n^d = T_s^d - T_{s-1}^d, s \in \{1,2,...,S\}; \quad (5)$$

where, I_n^a and I_n^d represent the arrival and departure delay changes between station s and station s-1, $s \in \{1,2,...,S-1\}$; T_s^a and T_s^d represent the arrival and departure delays in station s, respectively. It is worth noticing that the number of indicators is determined by the number of sections where the delay changes are considered for the delay propagation pattern discovery. In the case study section, we will investigate how many previous train delays should be considered for the prediction of an arrival delay.

B. Clustering results determination

After determining the input of the clustering model, the indicators will be fed into the K-Means algorithm to identify the delay evolution patterns. The clustering results of the K-Means algorithm are dependent on the parameter K (the number of initial clustering centers in Algorithm 2), because this parameter determines the final number of groups the data being classified into. Therefore, the most imperative task is to determine the parameter K. In this research, we will select the optimal parameter K based on a metric perspective.

We choose the parameter K based on the clustering evaluation metrics, which is a common approach to determining the clustering results [55]. Generally, the performance of a clustering model is also measured from two perspectives: the compactness of each cluster, and the separation between clusters. The compactness measures how chose the objects in a cluster are, and the separation measures how distinct a cluster is from other clusters. We choose two metrics to evaluate the clustering model, including a distance-based metric, namely, the Silhouette (S) score, and a variance-based metric, namely Calinski-Harabasz (CH) index [55], as shown in Eqs. (6) and (7).

$$\mathrm{S} = \frac{1}{K}\sum\nolimits_i \left\{ \frac{1}{n_i} \sum\nolimits_{x \in C_i} \frac{b(x) - a(x)}{\max\{a(x), b(x)\}} \right\} \quad (6)$$

$$\mathrm{CH} = \frac{\sum_i n_i d^2(c_i, c)/(K-1)}{\sum_i \sum_{x \in C_i} d^2(x, c_i)/(n-K)} \quad (7)$$

where x means an observation in the dataset; $a(x)$ is the average distance between x and other samples in the same cluster; $b(x)$ is the minimum average distance between x and samples in other clusters; D is the dataset; C_i is the i-th cluster; n represents the number of observations in D, n_i represents the number of observations in C_i. Detailed explanations of these two metrics can be seen in [55]. Better clustering results require smaller distances and variances of samples in a cluster; and larger distances and variances of samples between different clusters. Therefore, according to Eqs. (6) and (7), larger S and CH refer to better clustering results.

C. The BN paradigm for delay evolution

Constructing a BN model for predicting or classification consists of determining a dependency structure in-between the variables, and then learning its parameters (i.e., the intensity of probabilistic dependency). In this section, we introduce the BN structure for delay predictions, and the parameter learning of the BN model.

BN model is one of the widely used models in train delay prediction [15-18]. The BN models used in train delay prediction were developed by graphically interpreting the train operations, where train operation events, i.e., arrival and departure, are expressed by the nodes of the graph, and the relationships of train events are usually expressed by joint probability tables or density functions. Constructing a BN model to predict train delays have three critical steps: (1) determining the nodes of the model; (2) determining the causal relationships between nodes, i.e., the structure of the BN model; and (3) parameter learning, i.e., the determination of the joint probability distribution function. For the determination of the nodes of BN, we used the same idea used in the existing literature [15, 16], that the BN model is built on the reasoning of the train operation events. In other words, we use the known train operation events as the nodes of BN model to predict future delays.

Then, we need to determine the BN structure (causal relationships between BN nodes.) to predict train delays. Different BN structures have been proposed for different purposes in terms of railway traffic modelling. In general, there are two typical BN structures proposed for delay evolution and prediction. One is proposed for example in [15], of which each train operation event in the BN network was influenced by the previous two train operation events to capture the train evolution patterns in the two train operation events (parent nodes); The other is for instance proposed in [16], where the BN is built on the operation events of every two adjacent trains for considering the interactions or dependencies between trains. The former one [15], which uses the past two events as the parent nodes of the to-be-predicted node, is expected to capture the delay evolution patterns to some degree from the two parent nodes.

To incorporate train delay interactions, the structure of the BN model is proposed based on the paradigm of the BN in [15], i.e., any delay is dependent on delays in the past stations, as shown in Fig. 3. Here, we assume any delay is dependent on 2 $\times L$ (L is the number of previous stations considered) delays, rather than just one previous delay (i.e., Markov property, which is a common assumption in existing studies). Fig. 3 shows the causal relationships of train operation events in S+1 consecutive stations, where station S+1 is the target station, and the arrival delays in station S+1 are influenced by the delays in the previous L stations. In the case study section, the parameter L will be determined based on the train operation data.

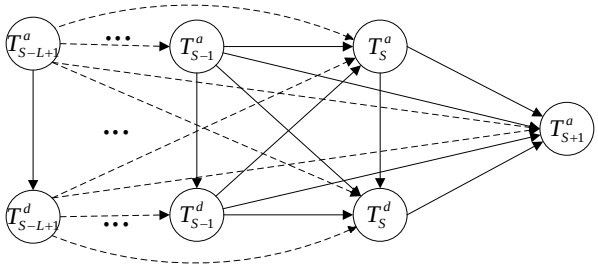


Fig. 3. The paradigm of the proposed BN structure for delay prediction.

D. BN update

After determining the BN structure, the parameters of the BN model (joint probability distribution functions) also need to be updated before it can be used for delay prediction. The BN parameters can either be given according to domain knowledge, or learn from data [53]. In this study, the parameters of the proposed BN model will be learned from the train delay data, and the nodes (train delays) of the BN are considered as continuous variables. Gaussian distributions will be set as the prior distribution of the BN nodes, and then, a maximum likelihood estimate (MLE) method will be used to update the parameters of the BN model and obtain the posterior probability distribution of each node and the joint probability distribution between nodes. The parameter learning using MLE method is as follows [56].

Given an acyclic graph, G, composed of N variables $X = \left\{x_1^{r_1}, x_2^{r_2}, ..., x_N^{r_N}\right\}$, we know from Section III-D that the joint distribution of these N variables can be written as Eq. (3). Therefore, what needs to learn from the training data are the joint conditional distribution of any given variable x_n and its parent variables $pa\ x_n$, i.e., $P(x_n | pa\ x_n)$. Here, r_n is the number of observations of variable x_n, and the value combinations of parent nodes $pa\ x_n$ are q_n. Given dataset $D = \left\{d_{x_n,1}, d_{x_n,2}, ..., d_{x_n,m}\right\}$ with m observations, according the MLE method, the optimal parameters are $\theta^* = \arg\max_\theta L(\theta | D)$, where $L(\theta | D)$ is the likelihood function. The MLE method estimate the optimal parameters of the network according to the degree of relief of the parameters and the samples. Therefore, the parameters that need to be learned from the data can be given by:

$$\theta = \left\{\theta_{ijk} \mid i = 1,..., N;\ j = 1,..., q_n;\ k = 1,..., r_n\right\} \tag{8}$$

Where $\theta_{ijk} = P\left(x_n = k \mid pa\ x_n = j\right)$. The loss function (log-likelihood) of the MLE method can be written as:

$$\begin{aligned} l(\theta | D) &= \log L(\theta | D) \\ &= \sum_{t=1}^{d} \log P(d_t | \theta) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{q_n}\sum_{k=1}^{r_n} m_{ijk} \log \theta_{ijk} \end{aligned} \quad , \tag{9}$$

where m_{ijk} is the number of samples in dataset D that satisfy $x_n = k$ and pa $x_n = j$. The optimal parameters can be obtained by solving (8) using Lagrange multiplier method [53].

E. Model performance evaluation

The CDBN model will be trained with the data in each cluster, respectively, resulting in K trained BN models with the same structure (as shown in Fig. 3), but different parameters (joint probability distribution functions). The performance of the proposed framework will be evaluated on the testing dataset. To systematically evaluate the proposed model, the following analyses will be conducted:

(1) Predictive result analysis. We compare predicted values with their respective realizations, i.e., showing the similarity of

the predicted values and the true values. In this process, qualitative methods, such as visualization techniques that can show the distribution of data will be used.

(2) Comparative analysis against other models. This step aims to show the advantages of the proposed model over other delay prediction models. We provide comparative analyses from three perspectives: (1) comparison between the proposed BN structure and the standard delay prediction model built on Markov property; (2) comparison between the proposed delay prediction framework (i.e., CDBN) and the same BN structure used in the CDBN trained with unclassified data; and (3) comparison between the CDBN with other state-of-the-art delay prediction models. The goal of the first analysis is to show the advantages of the proposed BN structure over the common Markov model, as the proposed BN structure in this study aims to expand the limits of the Markov property assumption. The goal of the second analysis is to show the significance of the delay evolution pattern discovery (i.e., the clustering) to train delay predictions, as we used the clustering technique to enhance the understanding of BN in the CDBN. The goal of the last comparative analysis is to show the improvements of the proposed framework to existing state-of-the-art train delay prediction models.

In addition, quantitative methods will be used to evaluate the performance of the proposed model, including cumulative distribution function (CDF) of the predicted result of each model, and three error-based metrics, i.e., the mean absolute error (MAE), the root mean squared error (RMSE), and the mean absolute percentage error (MAPE). The MAE, RMS, and MAPE calculations are shown in (10) to (12), respectively.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|e_i - o_i\right|, \tag{10}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(e_i - o_i\right)^2}, \tag{11}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{e_i - o_i}{o_i}\right| \times 100\%, \tag{12}$$

where N is the testing sample size, $\boldsymbol{o_i}$ is the observed value, and $\boldsymbol{e_i}$ is the estimated/predicted value.

V. Case Studies

A. Data description and analyses

The proposed model was tested on the train operation data from a Dutch railway line, namely the Amsterdam to Utrecht (A-U) main railway line, and a Chinese high-speed railway (HSR) line, namely the Wuhan-Guangzhou HSR (W-G). The sketch maps of these two railway lines are illustrated in Fig. 4.

The A-U railway line, which is approximately 45 km, is one of the main railway lines in the Netherlands. Trains operating on this line include both high-speed trains and regular trains. This line has a maximum speed of 140 km/h and includes three different trains, namely, the IC, LM, and SPR trains, which account for approximately, 47.3%, 2.5%, and 50.2% of the total train services. Train operation records in the southbound direction, namely, from Amsterdam to Utrecht were used in this study. It is worthy noticing that the train operation records in the Dutch railway line include the train operation records at each checkpoint. This means that the collected data in the Dutch railway line also include the train operation records in sections (between adjacent stations). The collected data from A-U railway line include 9,586 train records per checkpoint, and the data were from September 4, 2017 to December 8, 2017 (only weekdays).

The W-G HSR, which has a length of 1,096 km, is one of the longest main passenger railway lines in China. It intersects with other high-speed railway lines at the GZS, HYE, and CSS stations. 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. We focused on the trains operating in the northbound direction, i.e., from GZS station to HYE station. The dataset, covering the period from March 24, 2015 to November 10, 2016 (all days of the week), contains 57,796 train services per station in the GZS–HYE segment.

All the collected data were consisted of scheduled/actual arrival/departure times for each train at each station/checkpoint, train numbers, and dates. Table II lists four records of the train operation data in the database. For the W-G HSR, we focused on the prediction of train delays from SG to HYE station, five stations in total; for the A-U railway line, we focused on the prediction of train delays from Asb to Bkla, five stations/checkpoints in total. Stacking the data for predicting the delays at the target stations, we obtained 288,970 and 48,260 cases on W-G and A-U railway lines, respectively.

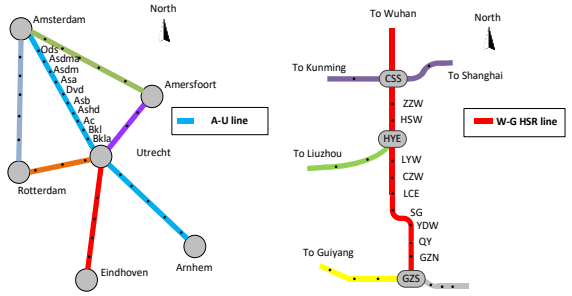


Fig. 4. Layout of (a) the A-U railway line and (b) the W-G HSR line.

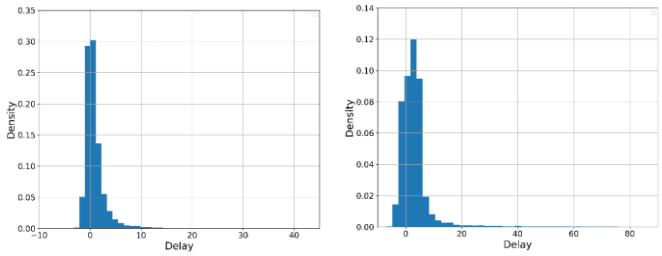
TABLE II
EXAMPLES OF THE TRAIN OPERATION DATA.

Train	Station	Date	AA	AD	SA	SD	OT
G6012	HYE	2016/11/9	9:55	9:55	9:55	9:55	II
G6014	HYE	2016/11/9	20:32	20:34	20:29	20:31	6
G6018	HYE	2016/11/9	14:46	14:46	14:41	14:41	II
G6020	HYE	2016/11/9	16:47	16:50	16:44	16:46	10

AA: actual arrival, AD: actual departure, SA: scheduled arrival, SD: scheduled departure, OT: occupied track. The passage tracks at the station are labeled with Roman characters, while the dwelling tracks are labeled with numbers.

The distributions of the train delays on A-U HSR and W-G railway are shown as Fig. 5, which shows that delays at both the railway systems, similar to other railway systems [16, 40], follow a long-tailed distribution. Because train delays follow a

long-tailed distribution, the long delays have low frequencies. The longest delay on A-U is approximately 50 min, whereas that on the W-G is approximately 130 min. However, the frequencies of such huge delays are extremely low. With regard to the delay horizons, we thus focused on the 90-minute delay prediction horizon on the W-G HSR, and the 45-minute delay prediction horizon on A-U railway, respectively, to establish a model with statistical significance. Any delay observations longer than 90 min on W-G HSR, and longer than 45 min on A-U railway were removed from the dataset. After the data processing, 99.8% of the data (48,180 cases) were left, and 99.7% of the data (287,972 cases) were left for A-U and W-G HSR lines, respectively. We randomly selected 30% of the them as testing data (14,454 cases on A-U, and 86,392 cases on W-G), and the remaining set is used as training data (33,726 cases on A-U, and 201,580 cases on W-G railway).



(a) (b)

Fig. 5. Delay distribution of (a) A-U HSR and (b) W-G railway.

With the train operation data, real-time train delay cases were extracted to show the different delay evolution patterns in practice. Fig. 6 shows three cases of a train (Train G552) in the data from station QY to SG on W-G HSR line. In Fig. 6, each line represents a possible situation of the train delay evolution between stations, e.g., the train delays first increased and then decreased on 2015/7/5, the train delays kept increasing between three consecutive sections on 2015/7/11, and the train delays kept unchanged between three consecutive stations on 2016/7/21. These cases show that train delay evolutions demonstrate different patterns, which highlights the necessity of distinguishing different delay evolution patterns in delay predictions.

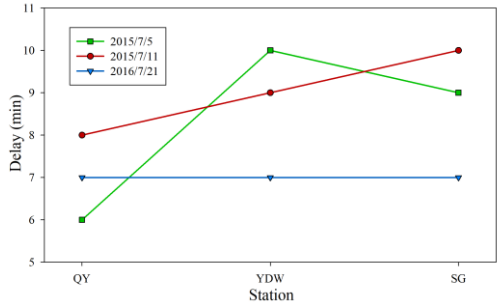


Fig. 6. Real cases of the arrival train delay evolutions.

B. *Investigation of delay dependences on a single train*

In Fig. 3, we determined the paradigm of the proposed BN structure. However, the amount of previous states that should be considered to minimize the predictive errors for each delay is to be determined (i.e., the parameter L in Fig. 3). In this section, we investigate the number of previous states (of the same train) that should be considered in each delay prediction from a data perspective. Here, we experimented on the unclustered data. To obtain the optimal BN structure, we show the predictive results of the BN models with the number of stations ranging from 1 to 5. This means that we used the delays of a train from the previous 1 to 5 stations as the parent nodes of the target delay. Fig. 7 shows the MAE and RMSE of the proposed BN paradigm with different previous states as inputs. It points out that the predictive errors are decreasing, with the increase of the number of stations. However, the results show that the errors have insignificant changes when the number of stations is larger than three. This is mainly because the effects of delays in far-away stations are already represented by the closer stations. To control computational cost, we chose the delays in the previous three stations as the inputs of the proposed BN model (i.e., the proposed BN structure is the BN shown in Fig. 3 taking the delays in the previous three stations as parent nodes, i.e., L = 3 in Fig. 3). Further, the results also indicate that the train operation is not a simple Markov process, wich again demonstrates the significance and contribution of the present study.

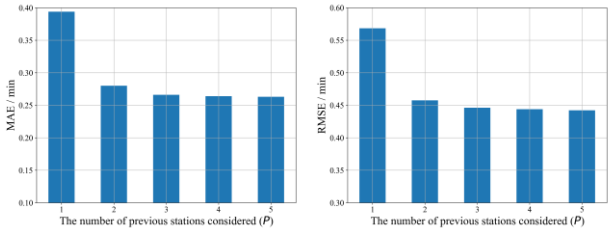


Fig. 7. The MAE and RMSE of BN models considering delays in different number of stations on A-U line.

C. *Investigations of indicators for delay evolution patterns*

In Section V-B, the number of previous stations that should be considered in the BN model was determined. The performance of the proposed model depends, to some extent, on the delay propagation pattern discovery. With the determined result for the BN model (i.e., L = 3) in Section V-B, we have two options on the indicators for delay pattern discoveries: (a) using the arrival and departure delay changes in the previous two sections (i.e., between the previous three stations) as input; (b) only using arrival and departure delay changes in the latest (i.e., previous one) section as input, as the delay evolution patterns from station S to station S+1 empirically have the highest correlation to the delay evolution patterns in the latest section (i.e., between station S-1 and S). To obtain the optimal indicators, we investigated predictive errors resulted from these two options, as shown in Table III. Here, the experiments were conducted by setting the number of clusters (K) at 4. The table shows that the BN model with the clustering indicators from the previous two stations (i.e., previous one section) can result in lower predictive errors. The main reason therefore is that the delay patterns in a section primarily depend on the previous pattern (e.g., delay changes from station S to S+1 are primarily dependent on those between station S-1 and S). Considering delay changes in earlier sections, in this test case, could be a disturbance of the clustering results. Therefore, we only use the indicators from the previous section, i.e., I_1^a and I_1^d in Eqs. (13) and (14), as the input of the clustering model.

$$I_1^a = T_S^a - T_{S-1}^a \quad (13)$$

$$I_1^d = T_S^d - T_{S-1}^d \quad (14)$$

where T_s^a and T_s^d are the arrival and departure delays in station s, respectively. With the determined input of BN and the clustering model, we can obtain the data used in this study for train delay prediction. The examples of the data (in minute) for clustering and prediction are shown in **Appendix C**.

TABLE III.
ERRORS RESULTED FROM DIFFERENT CLUSTERING INDICATORS ON A-U RAILWAY.

Candidate	MAE (min)	RMSE (min)
Previous one section	0.223	0.440
Previous two sections	0.225	0.470

D. Investigation of delay interactions between trains

In addition, we also investigated the delay interactions between trains. In other words, we experimented if considering more trains improves the prediction performance of the model. We therefore considered the interactions between two adjacent trains. The delay changes in the previous section of both trains, and the delay changes from the preceding train to the next train in the previous station as the input of the clustering model. For instance, to predict the delay of train N at station S+1 (train N just arrived station S), the I_1^a and I_1^d in Eqs. (13) and (14) of train N-1 and N, and the arrival delay change from train N-1 to N, and departure delay change from N-1 to N were selected as the input of the clustering model. We classified the data into four groups, and the BN models were trained on the clustered data. Here, the BN models also took the delays of both trains in the previous three stations (i.e., delays of train N-1 and N in station S, S-1, and S-2) as parent nodes.

Table IV shows that considering the train interactions failed to improve the performance of the predictive model, compared against the proposed method. This is mainly because considering train interactions inevitably has more inputs into the clustering model which results in inefficencies. However, train delay changes/jumps from one station to the subsequent station (e.g., from station S to station S+1) are primarily dependent on the delay jumps in the previous section (e.g., station S-1 to station S). The proposed model assumes that the delays with different propagation patterns are distinctively divided. Only the assumption is met, can the predictive performance of the BN model be significantly improved. However, considering more trains means more varieties of the delay propagation patterns, which could be harder for the clustering models to distinguish them.

TABLE IV.
THE COMPARATIVE RESULTS BETWEEN THE PROPOSED MODEL AND THE MODEL CONSIDERING TRAIN INTERACTIONS.

Railway	Experiment	MAE (min)	RMSE (min)
A-U	One train	0.223	0.440
	Two trains	0.242	0.454
W-G	One train	0.668	1.243
	Two trains	0.686	2.711

E. Pattern discovery

With the introduced metrics (S and CH) for clustering result determination in Section IV-B and the determined input/indicators of the clustering model in Section V-C, we can perform the K-Means clustering algorithm to discover the train delay evolution patterns. To explore the optimal *K* value, we performed the K-Means model with *K* ranging from 2 to 10, and recorded the S and CH values corresponding to each K in the training process. The results are shown in Fig. 8. This figure is a double y-axis figure, where the left y-axis represents the CH metric, and the right y-axis represents the S metric. Overall, Fig. 8 shows that the CH and S values first show an increasing trend and then show a decreasing trend with the increasing of the parameter K. As introduced the Section IV-B, the higher the S and CH values, the better is the clustering result. Fig. 8 shows that the CH and S metrics reach the highest values in K = 4. Therefore, the train operation data were finally divided into four clusters (patterns of delay changes).

The distributions of these four clusters in terms of arrival delay changes (I_1^a) and departure delay changes (I_1^d) are shown in Fig. 9. In Fig. 9, the horizontal and vertical axes represent the arrival and departure delay changes, respectively. This plot shows that these four clusters are distinctive on both A-U and W-G railway lines. From the results in Fig. 9, we can give illustrative names to each cluster: *decreasing delays/delay recovery* (***Category A***), *unchanged delays* (***Category B***), *small increasing delays* (***Category C***), *and large increasing delays* (***Category D***). The statistical results of these four categories are shown in Table V. These four clusters will be used to build a BN model, respectively, to enable the built BN models to capture the delay evolution patterns.

In addition, Fig. 9 shows that the largest delay jump on W-G HSR was about 60 min, while that on A-U was just 10 min. The huge difference of the horizon of the delay jumps on the two railway lines is mainly because the distance between every two stations/checkpoints on these two lines are significantly different. The average distance of two every station on W-G HSR is, approximately, 40 *km*, whereas that on the A-U railway line is, approximately, 3 *km*.

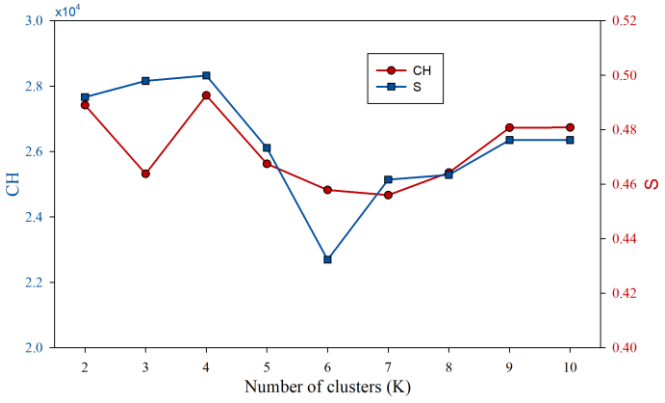


Fig. 8. The clustering results of different K values for A-U railway.

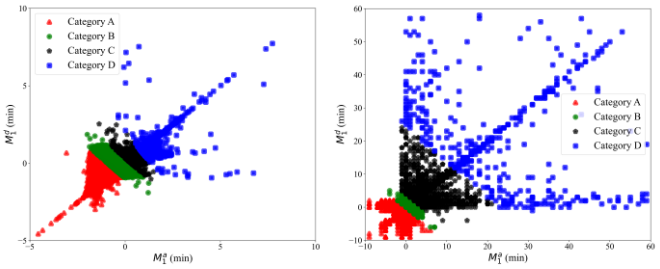


Fig. 9. Clustering result of data on (a) A-U HSR and (b) W-G.

TABLE V. STATISTICS OF EACH CLUSTER IN TERMS OF ARRIVAL AND DELAY CHANGES.

Line		I_1^a			I_1^d			
	Category	1st Q	median	3rd Q	1st Q	median	3rd Q	Sample
A-U	A	-0.98	-0.76	-0.59	-0.88	-0.75	-0.59	12,264
	B	-0.29	-0.2	0.01	-0.27	-0.19	0.03	23,325
	C	0.35	0.49	0.62	0.35	0.49	0.59	9,949
	D	0.96	1.14	1.35	0.72	0.86	1.08	2,642
W-G	A	-1	-1	-1	-1	-1	-1	112,652
	B	0	0	1	0	0	1	154,218
	C	2	2	3	2	2	3	20,433
	D	13	19	31	11	18	30	669

F. BN update and the probability in the BN

With the clustered data, we can train a BN model on each cluster, resulting in predictive BN models on A-U and W-G railways, respectively. In the BN, each node represents a random variable, and the arcs between nodes represent the probabilistic dependence between variables. In the proposed BN model, the variable to-be-predicted (i.e., the arrival delay in station S+1, T_{P+1}^a) depends on six parent nodes, (i.e., the arrival and departure delays in previous three stations, $T_{S-2}^a, T_{S-2}^d, T_{S-1}^a, T_{S-1}^d, T_S^a, T_S^d$). Therefore, the conditional probability of the proposed model can be denoted as $\mathrm{P}\left(T_{S+1}^a | T_{S-2}^a, T_{S-2}^d, T_{S-1}^a, T_{S-1}^d, T_S^a, T_S^d\right)$. In other words, the estimated values of the delays to-be-predicted (T_{P+1}^a) were the posterior probability distribution inferred based on its parent nodes. Fig. 10 shows the probability distributions of the parent nodes (conditions) and the prior and posterior probability distributions of the delays to-be-predicted in the BN model of each cluster.

G. Performance result of the proposed model

In this section, we evaluate the performance of the proposed framework. First, the predictive result of the model is demonstrated. Then, we evaluate the BN structure, clustering model, and the proposed hybrid model, respectively, by comparing against their counterparts.

1) Predictive result analyses

To better understand the predicted results of the proposed model, we first show the performance of the model, by comparing the distribution of the predicted delays against that of the observed delays. We first compared the predicted values against the actual values on the whole testing dataset, as shown in Fig. 11. Also, because the proposed model was built on clustered dataset, the distributions of the predicted delays and observed delays on each cluster was then investigated, as shown in Fig. 12. Figs. 11 and 12 showed that the distribution of the predicted values and the observed values are considerably symmetrical (the shape, width, height, and length of whiskers of every two neighboring violin plots) for all the testing data and each cluster, indicating the high identity and coincidence of the predicted values and the observed values.

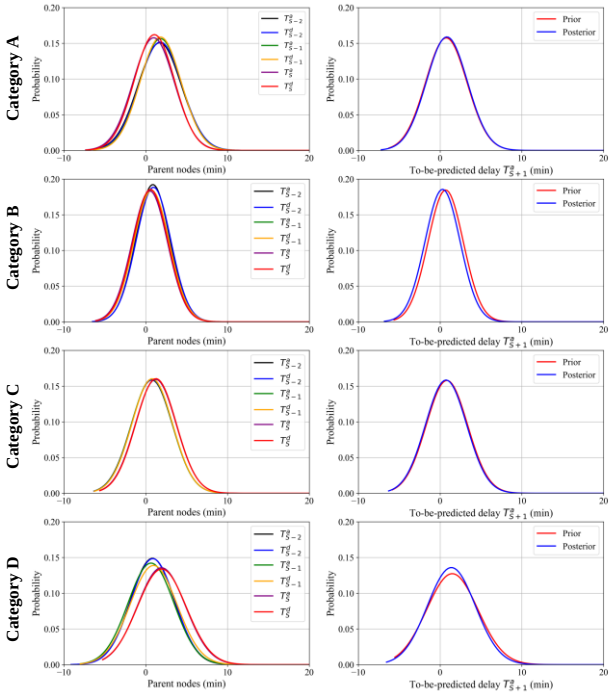


Fig. 10. Probability distributions in the BN models on A-U.

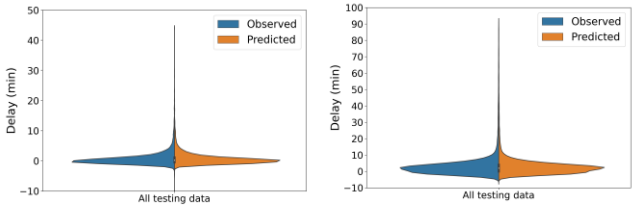


Fig. 11. Predicted values VS observed values on (a) A-U and (b) W-G.

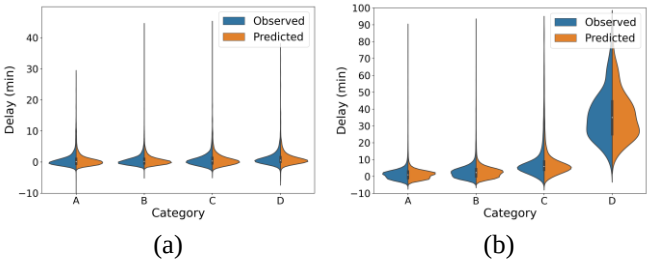


Fig. 12. Predicted values VS observed values for each category on (a) A-U and (b) W-G.

2) Evaluation of the BN structure

We first evaluated the performance of the proposed BN structure, by comparing against its performance with the typical graph or network models built on Markov property, since the predictive BN structure (the structure in Fig. 3 with L = 3) was proposed to overcome the Markov property in train delay predictions, These models include:

(1) Dynamical Markov model (DMM). In the DMM, the train delay evolution is treated as a dynamic process, and the probability transition matrix simultaneously updates over time and space. This model can be seen as a representative of many other models based on the Markov property assumption, such as Refs. [22-25].
(2) Hybrid BN model (HBN) in Ref [15]. In the model, each node takes the latest two train events as its parent nodes, which is an improvement over Markov property in existing train delay prediction studies.

We trained and tested the proposed BN, DMM, and HBN with the same unclustered training and testing data. therefore, we call them BN-ud, DMM-ud, and HBN-ud, respectively. The predictive results of them on the testing dataset are shown in Table V. We used an absolute metric, namely MAE, and a relative metric, namely MAPE, to show the performance, as shown in Eqs (10) and (12). Because MAPE cannot be calculated on observations equaling zero, the metric was calculated on the observed delays longer than 1 min. The results for A-U and W-G railway lines are shown in Table VI. The results in this table show that the proposed BN structure outperformed other Markov property-based models on the data of two railway lines. The results thus demonstrated the advantages of the proposed BN structure over other models built on Markov property, further indicating that the train operations cannot be simply treated as a Markov process.

TABLE VI.
EVALUATION RESULTS OF THE PROPOSED BN STRUCTURE ON A-U RAILWAY.

Line	MAE			MAPE		
	BN-ud	HBN-ud	DMM-ud	**BN-ud**	HBN-ud	DMM-ud
A-U	**0.266**	0.394	0.452	**16.90%**	22.70%	25.80%
W-G	**0.695**	0.703	0.704	**17.90%**	18.20%	18.30%

3) Evaluation of the delay evolution pattern discovery model

To show the significance of the clustering model, we demonstrate the comparison of the performances of the CDBN model against that of the same BN structure without K-Means clustering. Here, the BN model trained on unclassified data, is named "BN-ud". The only difference between the BN-ud and the CDBN is that the CDBN was trained on clustered data, while the BN-ud was trained on unclassified data. Specifically, the benchmark model (BN-ud) has the exact input information as the CDBN model (i.e., the information conveyed by the delays at the previous three stations/six states).

For the CDBN model, the BN structure (shown in Fig. 3 with L = 3) was, respectively, trained with the data in each cluster, resulting in four BN models that had the same structure but have different parameters (the joint probability distribution function). The testing data assigned with the same label as the training data were predicted by the corresponding trained BN model. For the benchmark, the same training and testing data used for the proposed model were used, and the data were used combinedly to train and test the BN-ud.

We selected the cumulative distribution functions (CDFs) of the predictive errors of each model on the testing dataset to show their performances, as shown in Fig. 13. Generally, the CDFs of the better model should be at the top and the left. Fig. 13 shows that the CDFs of the proposed model are at the top and left, indicating its higher accuracy in delay prediction than other benchmark models. This means that using the clustering technique is significant for improving the delay prediction accuracy.

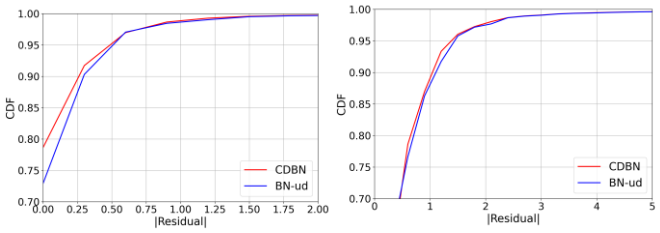


Fig. 13. The residual distribution of each model on (a) A-U and (b) W-G railway.

4) Evaluation of the CDBN

To demonstrate the performance of the CDBN model, we also selected other machine learning models in wide usability and extensibility in train delay prediction as benchmarks. In detail, the selected models include:

(1) Support vector regression (SVR): The SVR, whose main principle is to map data into a high-dimensional feature space using a nonlinear relationship, is a type of SVM suitable for regression problems [57]. The SVR model has been used in train delay prediction problems in Refs [3, 40, 51]. We optimized the kernel function and penalty coefficient. Finally, the kernel function was a linear kernel and its penalty function coefficient was 0.5.
(2) Artificial neural network (ANN) or multi-layer perceptron: In the ANN, the neurons between the adjacent layers are fully connected. The information flow is transferred from the input layer to the output layer [58]. The ANN has been used for delay prediction problems in Refs [30-32]. We optimized the number of hidden layers and the neurons in each layer. Finally, the ANN used in this study includes three hidden layers, each with 128 neurons.
(3) Random forest (RF): An RF uses an ensemble of decision trees $\{h(X,\beta_k),k=1,...\}$ for mapping vectors of predictor and dependent variables [59], where X is the input vector and β_k is an independent stochastic variable that decides the growth of every tree. The RF model has been used for delay prediction problems in Refs [41-43]. We optimized the number of trees in the forest and the depth of each tree. The RF in this study includes 300 trees and each has 12 splits.
(4) Long short-term memory (LSTM): Compared with the ANN, the recurrent connection from the output layer to the input layer at each time step of the LSTM has the advantage of capturing the information embedded in sequential data. As train delays are typical sequential data (time-series), the LSTM is expected to capture the train delay interactions. The LSTM has been used for train delay prediction in Refs [5, 33]. We also optimized the number of hidden layers and the units in each layer. Finally, three LSTM layers were used, each with 128 units.

All the benchmark models were trained and tested on the same dataset as the proposed CDBN model. This means that the SVR, ANN, RF, and LSTM were all trained and tested on the four categories, respectively. Also, an absolute metric (i.e., MAE) and a relative metric (i.e., MAPE) were used to show the performance of each model, as shown in Eqs (10) and (12).

First, we examined the performance of the model on each cluster and all the testing data. The comparative results of each delay prediction model for A-U and W-G railway lines are shown in Tables VII and VIII, respectively. These tables showed that the proposed model had the lower predictive errors on clusters for both A-U and W-G HSR railway lines, indicating its superiority over the benchmark models. In addition, the results showed that the predictive MAE and MAPE of the proposed model for all the clusters (the whole testing dataset) were 0.226 *min* and 14.2%, respectively, on A-U HSR, which had, on average, 8.7% and 11.6% improvements, compared against the best benchmark model (ANN). The predictive MAE and MAPE of the proposed model for all the clusters were 0.670 *min* and 17.1%, respectively, on W-G railway line, which had, on average, 10.5% and 11.9% improvements, compared against the best benchmark model (LSTM). These results also further demonstrated the significance of the proposed CDBN model.

TABLE VII.
PREDICTIVE ERRORS OF EACH MODEL ON DATA FROM A-U LINE.

Metric	Data	CDBN	LSTM	RF	ANN	SVR
MAE	Category A	**0.229**	0.296	0.282	0.268	0.341
	Category B	**0.183**	0.189	0.194	0.188	0.246
	Category C	**0.288**	0.331	0.322	0.321	0.416
	Category D	**0.359**	0.408	0.406	0.394	0.537
	Combined	**0.226**	0.258	0.255	0.248	0.322
MAPE	Category A	**13.4%**	17.6%	17.0%	16.0%	18.9%
	Category B	**12.9%**	13.4%	13.7%	13.4%	16.8%
	Category C	**16.1%**	22.8%	19.2%	18.8%	23.3%
	Category D	**19.3%**	26.2%	24.5%	24.0%	27.3%
	Combined	**14.2%**	17.5%	16.5%	16.0%	19.5%

Note: the bolds represent the best results.

TABLE VIII
PREDICTIVE ERRORS OF EACH MODEL ON DATA FROM W-G LINE.

Metric	Data	CDBN	LSTM	RF	ANN	SVR
MAE	Category A	**0.663**	0.703	0.712	0.708	0.775
	Category B	**0.654**	0.776	0.781	0.779	0.821
	Category C	0.780	**0.749**	0.865	0.823	0.893
	Category D	**2.124**	2.145	2.472	2.741	3.884
	Combined	**0.670**	0.749	0.764	0.759	0.815
MAPE	Category A	**20.0%**	20.4%	21.6%	21.5%	21.7%
	Category B	**16.9%**	20.9%	20.3%	20.2%	21.0%
	Category C	10.7%	**9.5%**	11.8%	11.3%	10.4%
	Category D	**6.5%**	7.0%	7.7%	8.7%	11.8%
	Combined	**17.1%**	19.4%	19.7%	19.6%	20.0%

Note: the bolds represent the best results.

5) Computational cost

Finally, to systematically evaluate the proposed model, we also recorded the computational cost of each model, including the models in the above three evaluation stages, as shown in Table IX. Here, for the computational cost of the CDBN, LSTM, RF, ANN, and SVR, it includes both the training time of these models and the computational cost of the clustering model. The comparison between BN-ud and HBN-ud shows that the increase of computational complexity by considering more previous delays is subtle, as delays in the previous three stations were considered in BN-ud, while delays in the previous two stations were considered in HBN-ud; the comparison between CDBN and BN-ud shows that the clustering model slightly increased the computational complexity, as the CDBN was trained on classified data, while BN-ud was trained on unclassified data. The comparisons between CDBN and LSTM, RF, ANN, and SVR show that the CDBN model had the lowest computational cost. This demonstrates the advantages of the CDBN model over other machine learning models, as the computational complexity is critical for the usability of the model in practice.

TABLE IX. COMPUTATIONAL COST OF EACH MODEL.

Model	Computational cost (s)		Testing time (s)	
	A-U	W-G	A-U	W-G
CDBN	0.55	2.10	0.05	0.21
DMM-ud	8.79	2.95	0.15	0.22
HBN-ud	0.09	0.31	0.01	0.06
BN-ud	0.10	0.39	0.01	0.09
LSTM	309.48	1137.10	0.39	1.56
RF	7.41	15.66	0.39	0.68
ANN	24.24	46.81	0.06	0.39
SVR	20.19	2489.11	0.10	567.23

Note: all the experiments were run on a laptop with an Intel Core i5 8250U CPU.

VI. CONCLUSION AND DISCUSSION

In this study, we proposed a hybrid framework, named context-driven Bayesian networks (CDBN) to improve the train delay prediction accuracy, by enhancing the model's understanding of train delay evolution patterns. The proposed model uses a K-Means clustering algorithm to identify the train delay evolution patterns, and then the data in different clusters are used to train a Bayesian network model, respectively. The comparison results of the proposed model against the selected benchmark models showed that the delay evolution pattern discovery is critical for the train delay prediction models, and it can extensively improve the train delay prediction accuracy.

Train operations demonstrate high dynamics over time, which requires more advanced models to uncover the delay evolution patterns. The proposed method, which uses a clustering technique to describe the dynamics in a more complex way than a simple straightforward predictive model, can address the complexity of the train operations. Also, the proposed BN structure, which takes multiple train delays as parent nodes, overcomes the Markov property assumption of the existing graph- and network-based train delay prediction

models. The accuracy of train delay prediction was substantially improved by the delay evolution pattern discovery model. In addition, the advantage of the CDBN over other models was demonstrated from different aspects, i.e., the evaluation of the BN structure, the evaluation of the clustering, and the evaluation of the CDBN. Last but not the least, the calculation of the delay jump/jump is easy, given the known/past train delays, which enables the proposed model to remain simple and efficient on delay pattern discovery.

The proposed model can still be improved in many different ways. This is because the proposed mode is built on a single train operation, which is incapable of capturing the interactions between adjacent trains. However, focusing on the delays of a single train also contributes to the literature and practice. This is because a few studies aimed at investating the delay propagation on a single train, and improving the accuracy of delay prediction models for a single train [15, 22-26]. In addition, one of the most important advantages of focusing on delays of a single train lies in that there is no need to consider the buffer times between trains in the predictive model. The number of train services is temporally different, e.g., it is different from hour to hour in the day, from season to season, and from year to year. This means that the headways between trains are usually different from time to time, which means the buffer times between trains are also different from time to time. Building models on a single train can allow us neglecting this factor. Otherwise, if the predictive models are built on multiple trains, the dataset in the time window with the same train frequency can only be used. This will extensively reduce the data volume of the models. For example, only 669 trains (training and testing combined) were included in Category D in the present study. If we train the model on each day of the week, the data volume for each day is, on average, smaller than 100, which severely harms the statistical significance of the predictive models. Therefore, with more data available, our future research will be focused on exploring the consideration of train interactions in the model. Also, exploring other ways/ideas to improve the accuracy of the train delay prediction models is one of our research interests.

References

[1] S. Binder, Y. Maknoon, and M. Bierlaire, "The multi-objective railway timetable rescheduling problem," *Transportation Research Part C: Emerging Technologies,* vol. 78, pp. 78-94, 2017.

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

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

[4] Y. Liu, X. Weng, J. Wan, X. Yue, H. Song, and A. V. Vasilakos, "Exploring data validity in transportation systems for smart cities," *IEEE Communications Magazine,* vol. 55, pp. 26-33, 2017.

[5] P. Huang, C. Wen, L. Fu, J. Lessan, C. Jiang, Q. Peng, *et al.*, "Modeling train operation as sequences: A study of delay prediction with operation and weather data," *Transportation Research Part E: Logistics and Transportation Review,* vol. 141, p. 102022, 2020.

[6] L. Li, X. Li, Z. Lu, J. Lloret, and H. Song, "Sequential behavior pattern discovery with frequent episode mining and wireless sensor network," *IEEE Communications Magazine,* vol. 55, pp. 205-211, 2017.

[7] H. Song, R. Srinivasan, T. Sookoor, and S. Jeschke, *Smart cities: foundations, principles, and applications*: John Wiley & Sons, 2017.

[8] C. Wen, P. Huang, Z. Li, J. Lessan, L. Fu, C. Jiang, *et al.*, "Train dispatching management with data-driven approaches: a comprehensive review and appraisal," *IEEE Access,* vol. 7, pp. 114547-114571, 2019.

[9] R. M. Goverde, "A delay propagation algorithm for large-scale railway traffic networks," *Transportation Research Part C: Emerging Technologies,* vol. 18, pp. 269-287, 2010.

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

[11] P. Kecman and R. M. Goverde, "Online data-driven adaptive prediction of train event times," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, pp. 465-474, 2014.

[12] F. Corman, A. D'Ariano, and I. A. Hansen, "Evaluating disturbance robustness of railway schedules," *Journal of Intelligent Transportation Systems,* vol. 18, pp. 106-120, 2014.

[13] T. Büker and B. Seybold, "Stochastic modelling of delay propagation in large networks," *Journal of Rail Transport Planning & Management,* vol. 2, pp. 34-50, 2012.

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

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

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

[17] P. Huang, J. Lessan, C. Wen, Q. Peng, L. Fu, L. Li, *et al.*, "A Bayesian network model to predict the effects of interruptions on train operations," *Transportation Research Part C: Emerging Technologies,* vol. 114, pp. 338-358, 2020.

[18] M. B. Ulak, A. Yazici, and Y. Zhang, "Analyzing network-wide patterns of rail transit delays using Bayesian network learning," *Transportation Research Part C: Emerging Technologies,* vol. 119, p. 102749, 2020.

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

[20] B. Li, T. Guo, R. Li, Y. Wang, Y. Ou, and F. Chen, "Delay Propagation in Large Railway Networks with Data-Driven Bayesian Modeling," *Transportation Research Record*, 2021, doi: 03611981211018471.

[21] J. Yang, Y. Han, Y. Wang, B. Jiang, Z. Lv, and H. Song, "Optimization of real-time traffic network assignment based on IoT data using DBN and clustering model in smart city," *Future Generation Computer Systems,* vol. 108, pp. 976-986, 2020.

[22] J. Barta, A. E. Rizzoli, M. Salani, and L. M. Gambardella, "Statistical modelling of delays in a rail freight transportation network," in *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 2012, pp. 1-12.

[23] A. Berger, A. Gebhardt, M. Müller-Hannemann, and M. Ostrowski, "Stochastic delay prediction in large train networks," in *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, pp. 100-111, 2011.

[24] İ. Şahin, "Markov chain model for delay distribution in train schedules: Assessing the effectiveness of time allowances," *Journal of rail transport planning & management,* vol. 7, pp. 101-113, 2017.

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

[26] M. Ş. Artan and İ. Şahin, "Exploring Patterns of Train Delay Evolution and Timetable Robustness," *IEEE Transactions on Intelligent Transportation Systems,* 2021, doi: 10.1109/TITS.2021.3101530.

[27] J. Wallander and M. Mäkitalo, "Data mining in rail transport delay chain analysis," *International Journal of Shipping and Transport Logistics,* vol. 4, pp. 269-285, 2012.

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

[29] F. Cerreto, B. F. Nielsen, O. A. Nielsen, and S. S. Harrod, "Application of data clustering to railway delay pattern recognition," *Journal of Advanced Transportation,* 2018, doi: 10.1155/2018/6164534.

[30] X. Chapuis, "Arrival Time Prediction Using Neural Networks," in *7th*

International Conference on Railway Operations Modelling and Analysis. Lille (France): International Association of Railway Operations Research, pp. 1500-1510, 2017.
[31] M. Yaghini, M. M. Khoshraftar, and M. Seyedabadi, "Railway passenger train delay prediction via neural network model," *Journal of advanced transportation,* vol. 47, pp. 355-368, 2013.
[32] J. T. Haahr, E. O. Hellsten, and E. van der Hurk, "Train Delay Prediction in the Netherlands through Neural Networks," 2019, https://orbit.dtu.dk/en/publications/train-delay-prediction-in-the-netherlands-through-neural-networks.
[33] C. Wen, W. Mou, P. Huang, and Z. Li, "A predictive model of train delays on a railway line," *Journal of Forecasting,* vol. 39, pp. 470-488, 2020.
[34] P. Huang, Z. Li, C. Wen, J. Lessan, F. Corman, and L. Fu, "Modeling train timetables as images: a cost-sensitive deep learning framework for delay propagation pattern recognition," *Expert Systems with Applications,* 2021, doi: 10.1016/j.eswa.2021.114996.
[35] D. Zhang, Y. Peng, Y. Zhang, D. Wu, H. Wang, and H. Zhang, "Train Time Delay Prediction for High-Speed Train Dispatching Based on Spatio-Temporal Graph Convolutional Network," *IEEE Transactions on Intelligent Transportation Systems,* 2021, doi: 10.1109/TITS.2021.3097064.
[36] X. Bao, Y. Li, J. Li, R. Shi, and X. Ding, "Prediction of Train Arrival Delay Using Hybrid ELM-PSO Approach," *Journal of Advanced Transportation,* 2021, doi: 10.1155/2021/7763126.
[37] E. Fumeo, L. Oneto, G. Clerico, R. Canepa, F. Papa, C. Dambra*, et al.*, "Big Data Analytics for Train Delay Prediction: A Case Study in the Italian Railway Network," in *Innovative Applications of Big Data in the Railway Industry*, ed: IGI Global, pp. 320-348, 2018.
[38] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra*, et al.*, "Train delay prediction systems: a big data analytics perspective," *Big data research,* vol. 11, pp. 54-64, 2018.
[39] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra*, et al.*, "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,* vol. 47, pp. 2754-2767, 2017.
[40] N. Marković, S. Milinković, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transportation Research Part C: Emerging Technologies,* vol. 56, pp. 251-262, 2015.
[41] M. A. Nabian, N. Alemazkoor, and H. Meidani, "Predicting Near-Term Train Schedule Performance and Delay Using Bi-Level Random Forests," *Transportation Research Record,* pp. 564-573, 2019.
[42] R. Nair, T. L. Hoang, M. Laumanns, B. Chen, R. Cogill, J. Szabó*, et al.*, "An ensemble prediction model for train delays," *Transportation Research Part C: Emerging Technologies,* vol. 104, pp. 196-209, 2019.
[43] J. Wu, Y. Wang, B. Du, Q. Wu, Y. Zhai, J. Shen*, et al.*, "The Bounds of Improvements Toward Real-Time Forecast of Multi-Scenario Train Delays," *IEEE Transactions on Intelligent Transportation Systems,* 2021, doi: 10.1109/TITS.2021.3099031.
[44] Z. Li, C. Wen, R. Hu, C. Xu, P. Huang, and X. Jiang, "Near-term train delay prediction in the Dutch railways network," *International Journal of Rail Transportation,* pp. 1-20, 2020.
[45] M. Al Ghamdi, G. Parr, and W. Wang, "Weighted Ensemble Methods for Predicting Train Delays," in *International Conference on Computational Science and Its Applications*, pp. 586-600, 2020.
[46] R. Shi, X. Xu, J. Li, and Y. Li, "Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization," *Applied Soft Computing,* 2021, 10.1016/j.asoc.2021.107538.
[47] P. Huang, C. Wen, L. Fu, Q. Peng, and Y. Tang, "A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems," *Information Sciences,* vol. 516, pp. 234-253, 2020.
[48] D. Chen, L. Wang, and L. Li, "Position computation models for high-speed train based on support vector machine approach," *Applied Soft Computing,* vol. 30, pp. 758-766, 2015.
[49] C. Jiang, P. Huang, J. Lessan, L. Fu, and C. Wen, "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,* pp. 353-363, 2019.
[50] P. Zhou, L. Chen, X. Dai, B. Li, and T. Chai, "Intelligent Prediction of Train Delay Changes and Propagation Using RVFLNs With Improved Transfer Learning and Ensemble Learning," *IEEE Transactions on Intelligent Transportation Systems,* 2020, doi: 10.1109/TITS.2020.3002785.
[51] P. Huang, C. Wen, L. Fu, Q. Peng, and Z. Li, "A hybrid model to improve the train running time prediction ability during high-speed railway disruptions," *Safety Science,* 2020, doi: 10.1016/j.ssci.2019.104510.
[52] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics),* vol. 28, pp. 100-108, 1979.
[53] T. D. Nielsen and F. V. Jensen, *Bayesian networks and decision graphs*: Springer Science & Business Media, 2009.
[54] I. A. Hansen and J. Pachl, "Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation," *Performance Evaluation,* 2014.
[55] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE international conference on data mining*, pp. 911-916, 2010.
[56] R. E. Neapolitan, *Learning bayesian networks* vol. 38: Pearson Prentice Hall Upper Saddle River, NJ, 2004.
[57] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing,* vol. 14, pp. 199-222, 2004.
[58] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems,* vol. 39, pp. 43-62, 1997.
[59] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R news,* vol. 2, pp. 18-22, 2002.

Ping Huang received the B.S. degree in Transportation and Ph.D degree in Transportation Planning and Management from Southwest Jiaotong University (SWJTU), Chengdu, China, in 2015 and 2020, respectively. He is currently a postdoc in the Institute for Transport Planning and Systems, ETH Zurich. His research interests include data mining, machine learning, decision making, and train operation control.

Thomas Spanninger received the B.S. and M.S. degrees in Statistics and Mathematical Methods in Economics, TU Vienna, in 2015 and 2018, respectively. He is current a doctorate student in the Institute for Transport Planning and Systems, ETH Zurich. His research interests include stochastic models of delays in railway systems, delay propagation in railway systems, and stochastic optimization problems.

Francesco Corman received MS degree in management engineering from Roma Tre University, 2006 and a Ph.D degree from Delft University of Technology, 2010. He currently holds the Chair of Transport Systems (Assistant Professor) at the Swiss Federal Institute of Technology - ETH Zurich, with main responsibilities in research and education in transport systems with particular focus on analytics and optimization methods for railways, public transport and logistics system and their interconnection, with special focus on their operations.

Appendix A: The proposed method (CDBN).

Input: the potential parent states in previous *S* stations and the to-be-predicted delay in station *S*+1, i.e.,

$X = \left\{T_1^a, T_1^d, ..., T_S^a, T_S^d\right\}$, $Y = T_{S+1}^a$.

Output: Train delay set (*D*) that impacts *Y*, indicators (*I*) that measure the delay evolution patterns, number of clusters (*K*), and predicted values for *Y* from K Bayesian network (BN) models.

Given the evaluation metrics for the predictive BN model (M_p).

For *i* = *S*, *S*-1, …, 1, **do:**

If $M_p\left[P\left(T_{S+1}^a \mid T_S^a, T_S^d, ..., T_i^a, T_i^d\right)\right] < M_p\left[P\left(T_{S+1}^a \mid T_S^a, T_S^d, ..., T_{i+1}^a, T_{i+1}^d\right)\right]$, **do:**

Let T_i^a and T_i^d be in *D*.

End

For j = 1, 2,…, S-1, **do:**

Calculate the indicators (the arrival delay change I_j^a, and departure delay change I_j^a).

Perform the clustering (C) algorithm with $I_1^a, ..., I_j^a$ and $I_1^d, ..., I_j^d$ being the input and an initialized parameter (K_o);

Obtain the clustering results $C_1, C_2, ...C_{K_o} = \mathrm{C}\left\{I_1^a, I_1^d, ..., I_j^a, I_j^d,\right\}$.

If $M_p\left[\sum_{n=1}^{K_o} P_{n,j}\left(T_{S+1}^a \mid D\right)\right] < M_p\left[\sum_{n=1}^{K_i} P_{n,j-1}\left(T_{S+1}^a \mid D\right)\right]$, **do:**

Let I_j^a and I_j^a in *I*.

End

Given the maximum number of clusters *N*.

Given the clustering model evaluation metrics (M_c)

For n = 2, 3, …, *N*, **do:**

Perform the clustering (C) algorithm with *I* being the input.

$\mathrm{K} = \arg\max_n \left(M_c^n\left[C_n(I)\right]\right), n \in \{1, 2, ...K\}$,

End

Perform the clustering model with the determined parameters *K* and *I*, and the dataset *D*;

Obtain the clustering results $C_1, C_2, ...C_K$;

Perform the BN model based on each cluster, i.e., $C_1, C_2, ...C_K$, respectively;

Perform evaluation with respect to M_c.

where M[*X*] means calculating the evaluation metrics based on *X*.

Appendix B: K-Means algorithm.

Input: dataset $D = \{x_1, x_2, \cdots, x_N\}$, and the initial number of clusters K_i.

Output: categories of every sample $C = \left\{C_1, C_2, \cdots, C_{K_i}\right\}$

Initialization: Randomly classify samples as K_i categories, and calculate the initial center of each category;

While $u_k \neq u_k$, **do:**

Calculating the sign (λ_i) of sample x_i, $\lambda_i = \arg\left(\min\left|x_i - u_k\right|\right)$

Classifying x_i into its nearest cluster, $C_{\lambda_i} = C_{\lambda_i} \bigcup \{x_i\}$

Updating the mean vector, $u_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$

Note: u_k is the mean vectors at *i*-th step u_k is the mean vector at (*i*+1)-th step.

Appendix C: Examples of the data for pattern recognition and delay prediction on A-U railway (min).

Train	Date	T_{P-2}^a	T_{P-2}^d	T_{P-1}^a	T_{P-1}^d	T_P^a	T_P^d	Y	I_1^a	I_1^d
863	2017/10/19	0.2	0.2	0.5	6.5	5.8	5.8	5.5	5.3	-0.7
4049	2017/10/19	0.8	0.7	0.0	1.3	1.9	1.7	0.6	1.9	0.4
3091	2017/11/7	-1.4	-1.4	-1.4	-1.4	2.5	2.6	2.2	3.9	4.0
1405	2017/9/29	-1.5	-1.5	-1.1	-1.1	-1.1	-1.1	-0.3	0.0	0.0
4045	2017/10/6	0.6	0.7	-0.4	-0.5	-2.2	-0.1	0.3	-1.7	0.4