Transportation Research Part C 104 (2019) 196–209

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

An ensemble prediction model for train delays☆

Rahul Nair[a,*], Thanh Lam Hoang[a], Marco Laumanns[b], Bei Chen[a], Randall Cogill[a], Jácint Szabó[b], Thomas Walter[c]

[a] IBM Research, Ireland
[b] IBM Research, Zurich, Switzerland
[c] Deustche Bahn AG, Germany

ARTICLE INFO

Keywords:
Train delays
Machine learning
Data mining
Big data

ABSTRACT

A large-scale ensemble prediction model to predict train delays is presented. The ensemble model uses a disparate set of models, two statistical and one simulation-based to generate forecasts of train delays. The first statistical model is a context-aware random forest that accounts for network traffic states, such as likely stretch conflicts and current headway's, exogenous weather, event, and work zone information. The second model is a kernel regression that captures train-specific dynamics. A mesoscopic simulation model that accounts for travel and dwell time variations as well as inferred track occupation conflicts, train connections and rolling stock rotations, is additionally considered. The models have been used in a proof of concept to forecast delays for nationwide passenger services network of Deutsche Bahn, which operates roughly 25,000 trains daily in Germany. Results demonstrate a 25% improvement potential in forecast correctness (fraction of predictions within one minute) and 50% reduction in root mean squared errors compared to the published schedule. The paper describes the models along with the big data challenges that were addressed in data storage, feature and model building, and computation.

1. Introduction

This paper addresses the problem of forecasting train delays in the medium term (up to 24 h in advance). Such forecasts are at the core of passenger facing services, where changes to the schedule, missed connections, or exceptional delays need to be communicated to users in advance. Timely dissemination of such adverse travel conditions can allow for recourse actions that limit the disruption to passenger journeys.

Train delays are caused by wide range of reasons from infrastructure problems relating to tracks, rolling stock, or wagons to administrative issues, and weather conditions. The UIC 450-2 standard delay attribution codes list more than 50 broad attributable reasons (UIC, 2009). On complex rail networks, with multiple operators, train delays are often the result of a complex interaction of factors such as priority rules, dispatcher response, track occupation, and overall network state. In such cases, it is often difficult to apportion delay cause to a single factor, or even determine which set of factors influence how delays propagate.

Delay estimation in railway networks has been well studied from the point of view of traffic control, schedule adjustments, and travel information systems. One set of methods to address delay propagation is through the use of simulation, either deterministic (e.g. Müller-Hannemann and Schnee, 2009) or stochastic (e.g. Berger et al., 2011). Another more recent set of methods has involved

☆ This article belongs to the Virtual Special Issue on "Machine learning".

* Corresponding author.
E-mail address: rahul.nair@ie.ibm.com (R. Nair).

https://doi.org/10.1016/j.trc.2019.04.026
Received 13 December 2018; Received in revised form 23 April 2019; Accepted 23 April 2019
Available online 16 May 2019
0968-090X/

the use of machine learning to estimate delays (see Ghofrani et al., 2018, for a recent survey).

In this paper we develop a purely data-driven methodology to generate forecasts, which combines simulation and statistical approaches in an ensemble. Ensemble models involve the use of multiple methods to generate forecasts, that are then combined to create a final forecast. The main rationale behind using such a framework is that gathering forecast from a diverse set of models reduces bias and error rates, since all models are unlikely to fail at the same time. As a result they offer better predictive performance than the individual models alone.

The work was undertaken as part of a proof of concept for Deutsche Bahn with the aim of increasing prediction quality by means of machine learning and big data. The pilot system was designed for online use to improve passenger information systems. The scale of the network, along with volume of predictions needed and practical computational requirements were also considered. As a result, the paper outlines a practical mechanism for on line, real-time forecasting of train delays.

The primary source of data used is the train passing message. These are messages generated by track-side sensors as trains pass through control points or stations. Using a corpus of train passing messages, we first reconstruct the infrastructure network and derive track capacities. For all the constituent models in the ensemble, we leverage the reconstructed service network and track capacities. We reconstruct the current network state, i.e. position information and delays of all trains, and key indicators related to near-term likely states primarily resource conflicts. For the machine learning models, a large set of influencing factors that are used to train the two model classes is derived from the network state. A context dependent linear ensemble is used to generate the final forecasts. Context-dependent here implies that the weights of the ensemble depend on the time of day, class of train, and operational status.

The main contributions of the paper are:

- The development of a practical, machine learning system for train delay prediction based on big data on operations, weather, and other exogenous data. Different from other work where only one or a few sources of data are used for prediction, we considered many types of available data, these include weather information, maintenance events, train message records at check-points, historical delays, schedule information, train class information and platform information. Besides these available data we also considered methods to reconstruct the entire network and to infer connection trains from the message data. To the best of our knowledge this is the first time in the literature such rich information was considered for train delay prediction problems.
- A large scale real world evaluation that demonstrates the value (and limitations) of such systems. Our evaluation is done for the entire train network in Germany with very large data at the scale that serves more than 36000 trains per day where we trained more than 22 models with 50 million training instances each.
- In term of modelling, this is the first time an ensemble method that combines statistical learning methods with simulation-based approaches is studied in the literature for train delay prediction problems. The results show that simulation-based approaches built based on domain knowledge by considering network reconstruction and connection train inference contributes a significant factor to the ensemble methods.

The paper is structured as follows. Section 2 presents the main challenges in the forecasting problem, along with a review of previous approaches. Section 3 presents the models and some systems aspects followed by the results in Section 4. The paper ends with a discussion in Section 5.

2. Motivation and prior work

Why is train delay forecasting difficult? We first outline some key challenges in generating accurate forecasts.

The long tail: Process times in railway networks can be highly unpredictable. This is particularly the case for short-term forecast horizons, where a range of operational aspects may influence delays. The *long tail* of delays make forecasting challenging. A naive model will learn patterns based on the bulk of the data and miss exceptional delays, thereby failing when the information is needed the most.

This long tail is best illustrated by a busy sub-network around the Hannover station as shown in Fig. 1. The segment between Vinnhorst (HVIN) and Hannover (HH) has an approximate travel time of six minutes. Fig. 2 shows the additional delay accrued by

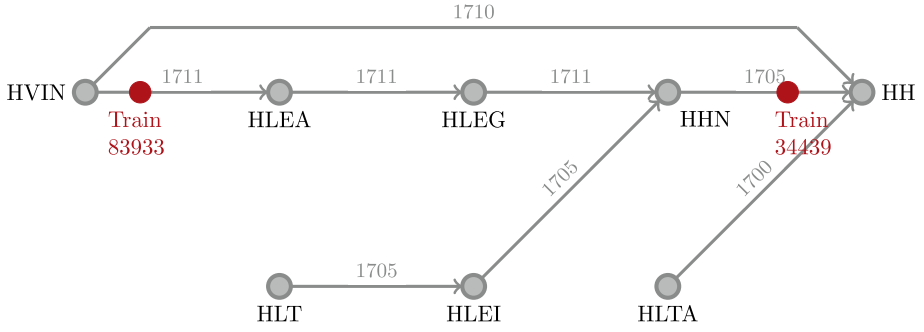

Fig. 1. Portion of sub-network around Hannover station showing stations/control points and track stretches.

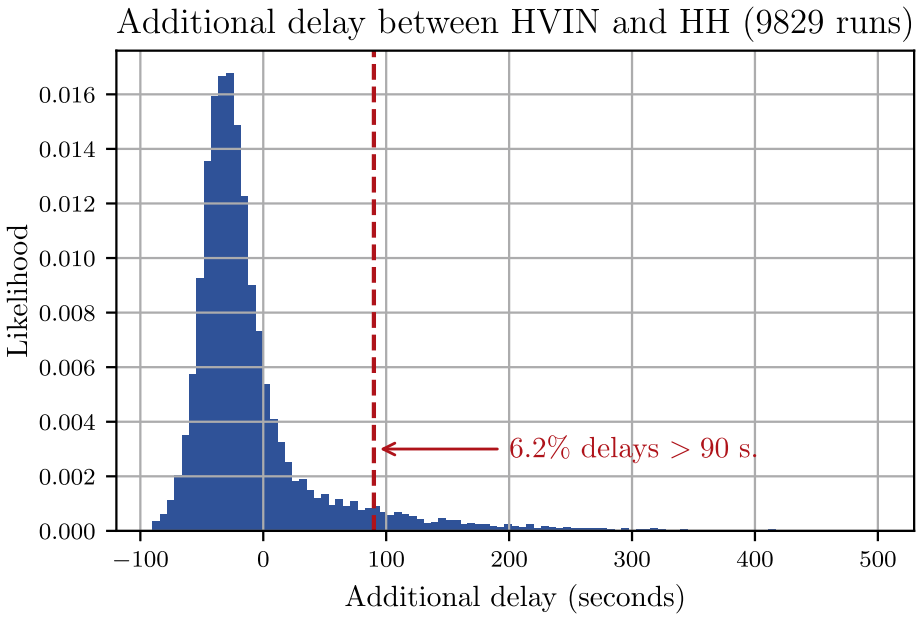

Fig. 2. Long tail of additional delays.

trains traveling on this segment. A non-negligible number of trains experience large delays for such a short trip, with delays as much as 500 s. Inspecting the train runs at the tail end of the distribution with a known set of attributable causes does not yield any direct causal links. In other words, no single cause can be blamed for the large delays. The lack of *discriminant* features make this prediction problem difficult.

Non-trivial resource dependencies: Railway networks have a fixed inventory (trains, tracks, stations). With the onset of delay, how schedules are impacted and how delays propagate throughout the network are difficult to estimate. Each major delay event has its own failure mode that make such assessments hard. While operational data on train movements may provide some estimates on how the system functions, several resources dependencies are not adequately represented in the data.

Take for instance a connecting train that may wait at a major stations to allow for passengers to transfer from a delayed train. Local dispatch may on occasion overrule any scheduled connections if platform resources were needed for other trains. Such dependencies can be difficult to ascertain purely from train movement data.

Scale: Lastly, nationwide services consisting of 25,000 trains daily covers a wide range of operating scenarios and service categories. Dynamics observed in one part of the network may not be directly related to other areas. This diverse range presents challenges from machine learning models, that seek to generalize information across the network. The scale also presents computational challenges in training models, incorporating up to date data, and generating real-time forecasts.

2.1. Previous work

A recent survey (Ghofrani et al., 2018) lists several works on data-driven delay estimation. A summary of related works is also presented in Corman and Kecman (2018) which classifies previous work based on approach, dynamics, and relations.

Several methods have considered machine learning for the prediction problem. Marković et al. (2015) present a support vector regression that takes as input seven features related to the train, schedule, infrastructure and headway to generate forecasts. They report on experiments around the station of Rakovica, Serbia and show mean forecast accuracy of 10 min. Gorman (2009) estimates freight congestion delays using a linear regression model that considers among other aspects, capacity, utilization, meets/passes, and preceding trains as features. Wang and Work (2015) use a regression model based on historical data to predict delays for passenger services for Amtrak. Barbour et al. (2018) use a support vector regression to estimate freight delays. They use origin–destination specific features, train priority and train counts as influencing factors. They do not report on prediction error, but show relative improvement over a baseline of historical mean forecasts. Recently, Oneto et al. (2017) demonstrate the use of shallow extreme learning machines. They demonstrate a distributed, in-memory approach using Apache Spark yields good performance. Additionally, they report that including weather data as a feature in their models yields a 10% improvement (see also Oneto et al., 2018). They define some novel, non-traditional forecast quality metrics to assess model performance.

Goverde (2010) use max-plus algebra to propagate delays along a scheduled timetable. They use a timed-event graph. A similar approach is shown in Kecman and Goverde (2013) for the Dutch railways. In this work they also describe conflict detection schemes to identify segments where multiple trains seek to occupy the same infrastructure delays.

A different data-driven approach is taken by Corman and Kecman (2018) where they present a probabilistic graphical model to assess a causal link between various network events. Here the state is determined by the delay of a 3-tuple of (train, station, and

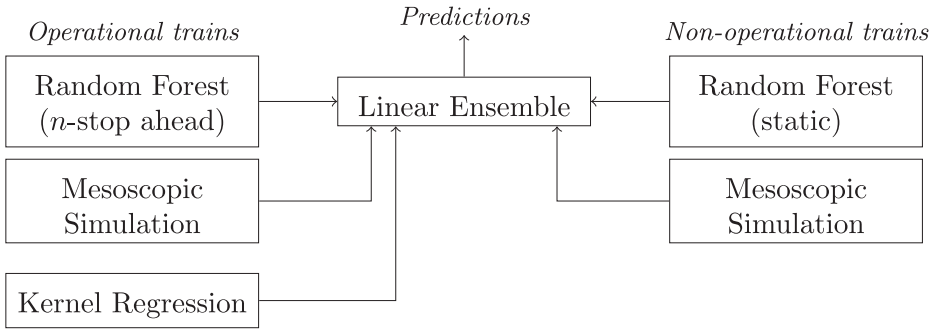

Fig. 3. Model structure.

arrival/departure indicator). They conduct experiments along a corridor with approximately 300 trains daily and report mean absolute error of 1.4 min for predictions horizons of 60 min.

Khadilkar (2016) presents a short-term delay prediction procedure for evaluating robustness of a network. The algorithm employs train priority information to assess how delays propagate when resources are in contention. Nabian et al. (2019) present a two level random forest model, the first level estimates the direction of the delay, while the second estimates the magnitude of the delay. Comparing with other machine learning methods, the two level model performed better on real world data from the Netherlands.

In recent work Lulli et al. (2018) tackle a similar problem of train movements using a hybrid model, where hybrid implies combined experience-based models (those that encode knowledge of networks, trains, and operators) and those that are data-driven based on operational data. The demonstrate prediction models for running time, delays based on a decision tree, where top-level splits are based on network factors and the lower level splits are from operation data. They show results on the Italian railway network.

3. Models

Formally, for a train i we seek to predict its delay sequence denoted by $\boldsymbol{d}_i = \{d_s(A),\ d_s(D) | s \in S_i\}$ which is a tuple of arrival (A) and departure(D) delays and S_i is a stop sequence for train i. A historical data set of delays is available for all trains. For each train in the historical data, we construct a corresponding feature sequence $\boldsymbol{x}_i = \{x_s(A),\ x_s(D) | s \in S_i\}$. Note that each feature vector $\boldsymbol{x}$ only involves observations available before the arrival/departure event. We seek to build a model M: $\mathbb{R}^n \rightarrow \mathbb{R}$ such that a forecast $\hat{\boldsymbol{d}}_i = M(\boldsymbol{x}_i)$ can be generated. We future distinguish two cases. For operational trains, a partial delay sequence is available. In this case, the feature vector includes any features related to previous delays. For non-operational trains, the entire sequence needs to be predicted.

The ensemble model proposed, as shown in Fig. 3 consists of two families of models, one for operational trains and another for non-operational trains. Non-operational trains are services that have departure times in the future. They have less contextual information and fewer features compared to operational trains. For example, track occupation conflicts are difficult to estimate so far ahead into the future and have limited value in generating a forecast.

Ensembles have been shown to work well in practice for a wide range of applications (Opitz and Maclin, 1999; Rokach, 2010). The main idea is to combine point forecasts from several methods to generate a single final forecast. The constituent methods should be *diverse*, i.e. forecast errors should be uncorrelated and all models should not all fail at once. A diverse set of models reduce forecast bias and ensembles outperform individual models. The overall prediction system is designed to be purely data-driven and allow for real-time operations. We next describe the data sources followed by details of each model.

3.1. Data

Several data sources were considered in building the set of influencing factors (features) for the models.

Train passing messages: Train passing messages are generated by track-side sensors when trains pass through specific locations. The locations can be at stations or at intermediate points, such as switches, collectively referred to as control points. There are roughly 1.3 million passing messages generated daily nationwide by all trains, including freight services. Table 1 shows a brief excerpt. Each train is identified by the service day (BTG) and a train number (ZN). Control points have a unique identifier BSTID. The message order (NR) and status (FSSTAT), used to indicate if its an arrival, departure, or pass-through event are also included. The schedule time (RSOLL) and actual time (IST) can be used to compute delays. The planned platform allocation (SGLSID) and actual platform

Table 1
Example train passing messages.

BTG	ZN	NR	FSSTAT	BSTID	RSOLL	IST	SGLSID	IGLSID	SSTRID
2014-01-01	2K	1	5	RXBA	2014-01-01 12:25:06	2014-01-01 12:25:54			4404
2014-01-01	2K	2	3	RB	2014-01-01 12:26:30	2014-01-01 12:27:14	195	195	4404

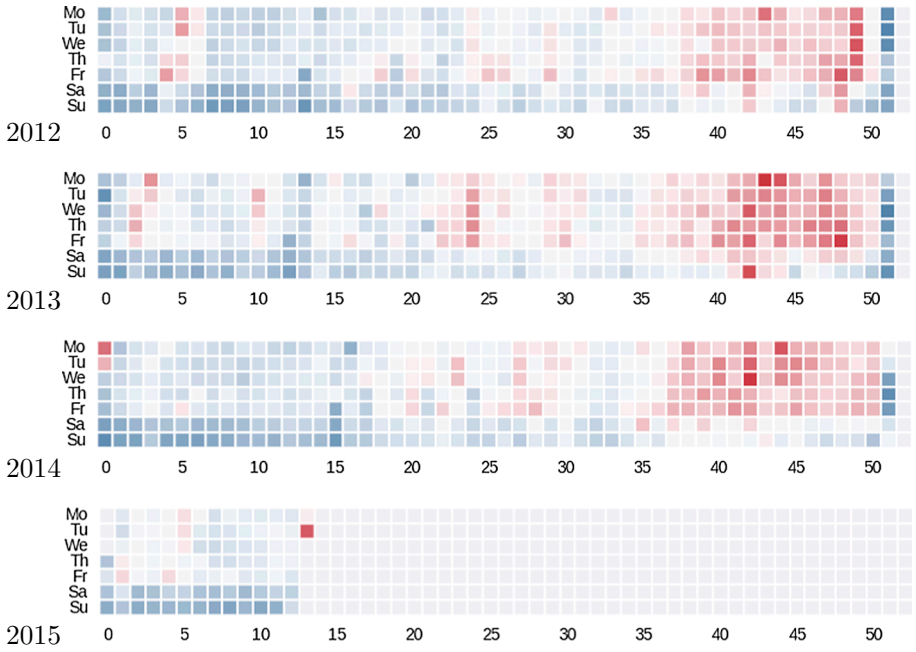

Fig. 4. Seasonal patterns of service reliability 2012 - April 2015 for all passenger services. Reliability here is measured as the fraction of services at all stops nationwide that arrive within one minute of scheduled arrival (red-lower reliability, blue-higher reliability). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

allocation (IGLSID) are included for messages that relate to arrival/departures at stations. The track stretch (SSTRID) indicates the stretch assigned the train. The data is generally of high quality with low failure rates (<0.5%) in the data generation pipeline. Some error detection routines were implemented to screen for potential errors in this data. We use 3.25 years of data (2012 - April 2015) for the subsequent analysis. Fig. 4 shows seasonal patterns in train operations with higher variability in the Fall months.

Train classification: Trains, referenced by their service day and train number (BTG, ZN) are classified into several major service categories and sub-categories. The categories can refer to high-speed trains, for example Inter-City Express (ICE), long distance, or overnight services. Each train is also associated with a line identifier, which indicates line variants linked to the service. Each train has a designated maximum speed.

Stations data: Attributes for the 5,600 stations including province, station classification, address and service types were considered.

Delay attribution data: By regulations, delays greater than 90 s must be reported and attributed to one of the major categories. This data is not available in real-time but at the end of each day. The data shows the time, location, trains involved, and the reason codes given by the field engineers for the delay. The data is not directly ingested by the model, but is used in two ways. First, given two trains and a track conflict delay reason, it is used to estimate how track conflicts are resolved, i.e. which train is more likely to be delayed. Second, the data is used to infer how local dispatchers hold connections at hubs and for how long.

Work zones: A database of work zone information indicating location, duration, and likely impact on different train categories was considered.

Weather: National weather data from 92 weather observatories were considered. Snow conditions, visibility and temperature were included. Weather generally does not have a large impact on delays. Based on delay-attribution data, less than <3% of delays are directly attributed to weather conditions. Weather related delays also have seasonal variability.

Event calendars: Federal and regional holiday calendars were included in the feature list.

Taken together, the data sources employed present a rich characterization of the operational state of networks daily.

3.2. Inferring the network

One key step to allow for feature extraction was network reconstruction and capacity estimation directly from the passing messages. We designed a method that generated train-class specific networks. The inferred network is employed for various down stream tasks, such as inferring train paths, conflict status estimation, and estimating typical travel time and travel time variability between given points of the network.

The method uses passing messages sorted by date, time and train. Subsequent control points are recorded from this sequence. If there are sufficient observations (to avoid erroneous connections), the control point, track stretch is recorded as an edge. Frequency of transitions from each outgoing edge is recorded at each vertex. Mean and standard deviation of travel times are also recorded. Next, a feasibility matrix for each station/control point is constructed. The feasibility matrix records pairwise edge feasible flows at

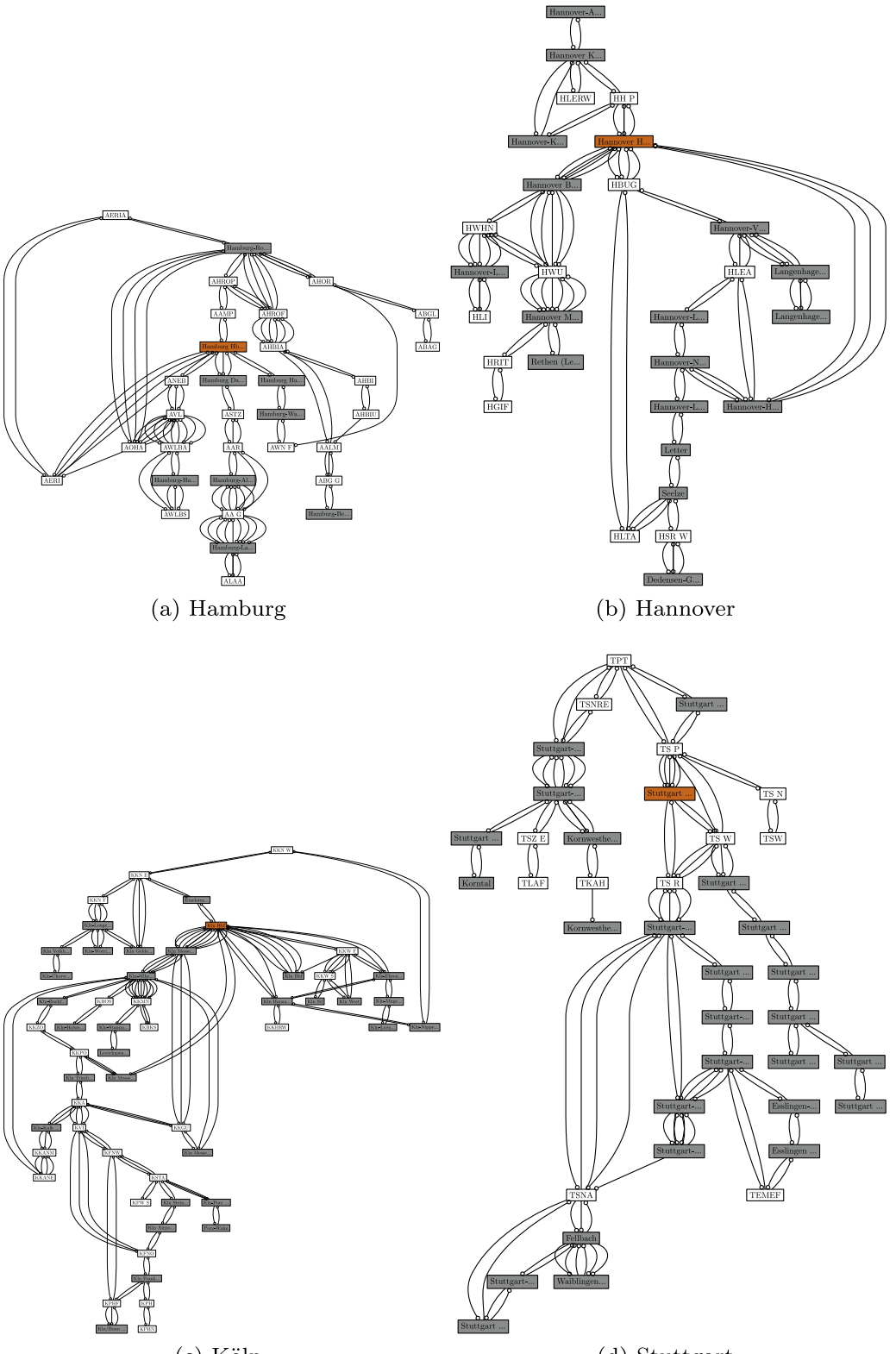

Fig. 5. Inferred sub-networks within 5 min mean journey time of selected major stations for January 2014 showing intermediate control points (grey) and inferred track capacity (number of links).

Table 2
Features for the random forest model classes.

Type	Feature	Op.	Static
Train-specific	Train class/sub-class/priority	Y	Y
	Maximum speed	Y	Y
	Delay at last 10 observations	Y	N
	Travel time to destination	Y	N
	Number of stops to destination	Y	N
	Journey pattern cluster	N	Y
	Historical delay of the last run	Y	Y
	Historical travel times	Y	Y
	Dwell times	Y	Y
Infra	Designated platform	Y	Y
	Station attributes	Y	Y
	Historical mean delay at tracks, platforms	Y	Y
	Actual platform	Y	N
	Track allocation	Y	Y
	Track/platform change status	Y	N
Network related	Station complexity (out-degree/in-degree)	Y	Y
	Actual Headway of k trains downstream	Y	N
	Planned headway of k trains at same track/platform	Y	Y
	Previous delays at station/track	Y	N
	Platform conflict indicators	Y	N
	Estimated gaps between conflict arrival times	Y	N
	Gaps between conflict arrival times normalized to scheduled gaps	Y	N
	Track occupation conflict indicators	Y	N
	Service connection matrix	Y	Y
	Conflict resolution probability	Y	Y
	Historical travel time through sub-network	N	Y
	Sub-network historical performance	N	Y
Connection related	Estimated delay of connected train	Y	N
	Delay of the train in its previous run	Y	Y
External	Weather features: wind, snow, rain, storm	Y	Y
	Calendar features (workdays, hour of day, days of year, months)	Y	Y
	Holiday features - Federal/regional/holidays and date before	Y	Y
	Maintenance features - information about maintenance, roadwork for each location	Y	Y

the station by identifying movements by two trains in a short time window. This is used to identify potential conflicts between trains when there are deviations from the schedule. Fig. 5 shows examples of inferred networks around some major stations.

One limitation of the reconstruction step is that no formal validation was performed. Reconstructed networks around several major hubs were inspected by hand and were found to be accurate. Since hubs have more complex geometry, one can expect remaining portions of the network to have good accuracy as well.

3.3. Random forest

Recall that we seek to build a model M: $\mathbb{R}^n \rightarrow \mathbb{R}$ to forecast delay sequences $\hat{\boldsymbol{d}}_i = M(\boldsymbol{x}_i)$. We first present the feature vector $\boldsymbol{x}_i$, since this is an important aspect in development of machine learning models and referred to as feature engineering. This step describes the process by which *influencing factors* are considered and incorporated into the model. Model form is almost secondary to feature engineering when it comes to forecast quality.

We had extensive consultations with subject matter experts to arrive at a set of factors that had explanatory power, i.e. could aid in generating forecasts, and could be estimated from data. We followed an iterative approach, where model designed included features in the model and evaluated performance impact. Domain experts then reviewed the results and suggested additional attributes to consider or commented to performance improvements. Through the iterative process the quality metrics were also refined as we better understood specific performance attributes. While at the start of the effort we focused on two performance metrics, by the end we reported 80 different quality measures.

Overall, roughly 350 features for operational trains, and 70 features for non-operational trains were constructed, a summary is shown in Table 2. The set of features can be categorized into five different types:

- **Train-specific**: train properties and real-time train state, this information directly influences the delay of each train in the network,
- **Infrastructure**: static and real-time information regarding the stations, platforms and tracks. This conveys infrastructure properties such as how busy the station is and how frequently delay happens at the given track.
- **Network related**: a class of features related to state of delays across the network. An important subset of features are track and

platform occupation conflicts. Conflicts are best described through an actual example network around Hannover (HH) shown in Fig. 1. Consider two trains going from HVIN and into HH. The first train 34439 departs HVIN at 2:09 pm and has a scheduled to dwell at HH from 2:17 pm till 2:20 pm. The second train 83933 passes through HVIN at 2:12 pm and is 10 min behind schedule and is expected to arrive at the same platform at 2:18 pm. Considered together, this is infeasible due to a platform occupation conflict. What actually happened in this example is that the dispatcher re-routed the train to another platform where train 83933 arrived at 2:21 pm - almost 3 min later.

Extending this in general, at any point in time, the feature generation routine aims to determine the likelihood of such downstream conflicts along tracks and platforms by considering arrival times within narrow time windows. We considered time windows of 1, 2 and 5 min. If two (or more) services are likely to share the same infrastructure, the conflict indicator is set to 1 or 0 otherwise. This indicator is assigned to both trains. At a subsequent step, using the delay attribution data, we evaluate which train accrued delays historically and assign the probability to an additional feature, if the two services have conflicted in the past.

- **Connection**: delay may be caused by connecting trains, i..e trains that wait for another train to allow for transferring passengers. We also include cases where a train that finishes a delayed run and is turning around for the next run. The data doesn't have information about such connections, we determine connected trains from data using rule-based approach.
- **External factors**: calendar features, weather information, roadworks and maintenance activities and holidays are included as these factors influence the delay.

With the feature set generated for the entire dataset, we experimented with various model types that provided best accuracy at acceptable compute costs. Support Vector Regression models was evaluated and found not to provide the best accuracy and is quadratic in training data volume. The models were also very sensitive to hyper-parameters. Performance for Generalized Additive Models was also lower. At the end we selected Random Forests. The choice was driven by forecast accuracy results and additionally the possibility for incremental training, i.e. updating of model parameters when fresh data is available without parsing the entire training dataset, and parallel training.

Two families of random forest models are trained. First, for operational trains, an n stop ahead model is trained for each $n = 1, ...,10$. The random forest is limited to 15 trees per model, models for departure and arrival time prediction are trained separately. Increasing the number of trees had no noticeable impact on forecast quality. For static/non-operational trains, two models, one for departure and one for arrivals were trained. Overall there are 22 random forests for both static and operational cases.

One advantage of random forests and decision trees are that they can be interpreted. However, for large scale models with tens of thousands of nodes and several trees, this can be challenging. Since trees in random forests split the data based on feature importance, the top portions of the tree show more influential factors. To evaluate feature impact on forecasts, we visualize the top 25% of the nodes in the tree. We cluster features into several thematic clusters. By examining how many nodes from the thematic cluster occur in the selected set of nodes, gives an indication of the importance of that theme. Greater the number of nodes from the thematic cluster, the more the feature theme is important. Fig. 6 shows these plots for different clusters. Train-specific attributes are shown as important, as are conflict and headway related features. Connection-related features are sparsely used, since this impacts a few high-priority intercity trains.

3.4. Kernel regression

The main idea behind the kernel regression model is to store a reference catalog of movements for each train. The forecast is then generated by a weighted sum of the reference catalog, where the weights are computed by measuring the similarity between the train of interest and the reference set. This approach has been successfully applied to trajectories from bus movements (Sinn et al., 2012; Nair et al., 2014; Andres and Nair, 2017).

For train movements, we define a reference set for each unique ordered set of stations, i.e. $\{S_1, S_2, S_3, ...,S_N\}$. Each trajectory in the reference set $\boldsymbol{M}$ is denoted by arrival-departure pairs for the ordered set of stations, i.e. $\boldsymbol{d}^m = \{(d_1^m(D), d_2^m(A), d_2^m(D), d_3^m(A), ...,d_N^m(A)\}$. Arrivals at the start stations and departure at the final station are ignored. For a partial trajectory, $\boldsymbol{d} = \{d_1(D), d_2(A), d_2(D), d_3(A), ...,d_L(A)\}$ of a train that has progressed till stop L, we seek to make forecasts for downstream stations $\widehat{d}_{L+h}$.

To measure similarity of two trajectories $\boldsymbol{x}, \boldsymbol{y}$ we define a Gaussian kernel,

$$\text{kern}(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|x - y\|^2/b) \tag{1}$$

where b is known as the bandwidth parameter which controls how the weights are spread in the reference set. Since services can catch up (or accrue additional delays), more recent delay observations are more important from a forecasting perspective than older observations. To account for this, a windowing parameter w is used to restrict how far back in history two trajectories should be compared. To standardize the kernel argument, the empirical variance $\sigma_i(s)^2$ at each station arrival/departure is employed. In terms of arrival departure delays, the kernel weights are computed as

$$\text{kern}(\boldsymbol{d}, \boldsymbol{d^m}) = \exp\left(-\frac{1}{b}\sum_{i=w}^{L}\sum_{s\in[A,D]}\frac{(d_i(s) - d_i^m(s))^2}{\sigma_i(s)^2}\right). \tag{2}$$

The forecast delays are then computed by

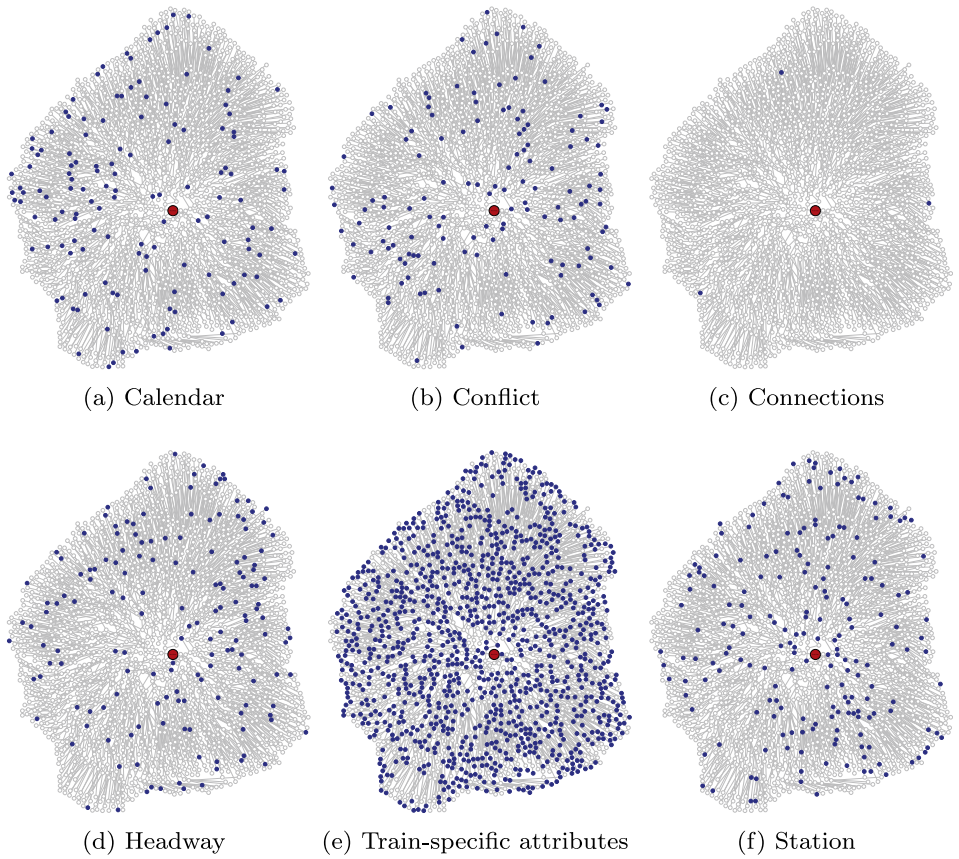

Fig. 6. Random Forest for operational trains showing roughly 25% of one of 15 trees for different feature categories. Highlighted nodes (blue) closer to the root node (red) show higher importance. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\hat{d}_{L+h}(s) = d_L + \frac{\sum_{m \in M} \text{kern}(\boldsymbol{d}, \boldsymbol{d_m})(d_{L+h}(s) - d_L^m)}{\sum_{m \in M} \text{kern}(\boldsymbol{d}, \boldsymbol{d_m})} \tag{3}$$

Delays can be represented in absolute or relative terms. We tested three different mechanisms to represent the trajectories as shown in Fig. 7. The first was based on travel time, the second on delays, and the third based on additional delay, i.e. delay accrued since the previous stop. Kernels based on delays and additional delays substantially outperformed those based on travel time. The procedure based on additional delay implied that the reference set itself needed to be recomputed each time there was an observation. This was computational prohibitive, and the final implemented model was based on delays.

Some practical considerations need to be addressed. The kernel regression model is used only for operational trains, i.e. for trains where some passing messages have been obtained. We tested a robust kernel estimator for non-operational trains with dispatch times in the future, but this did not perform well. Since there is a kernel reference catalog for each service (ordered sequence of stops), and a large number of kernel catalogs - several thousand - needed daily, mechanism to cache the catalogs for quick retrieval were necessary. On very infrequent trains, the catalog can be sparse leading to lower quality forecasts. A threshold based heuristic was used to discard forecasts generated by reference catalogs with fewer than 100 trajectories. We additionally tested two variants of the model, one where trajectory similarity was computed solely based on station arrival/departure messages, and one including intermediate passing messages, such as those generated at control points. The inclusion of additional data in the latter case lead to no increases in forecast accuracy. This is partially due to higher empirical variance observed at intermediate control points.

3.5. Simulation model

A conceptually different approach compared to a statistical or machine learning model is to use simulation for prediction. In railway research as well as industrial practice, simulation is a very popular tool to study various aspects starting from the network layout to the stability of timetables and the resulting service quality for passengers, whereas the level of detail ranges from microscopic models to macroscopic models (White, 2005). A lot of efforts have focused also on agent-based simulation of transport networks (Balmer et al., 2008; Fellendorf, 1994). Due in large part to the lack of sufficient data for fine-grained calibration of a large set of parameters, agent-based simulators have traditionally been used for planning applications, such as transit modeling, network

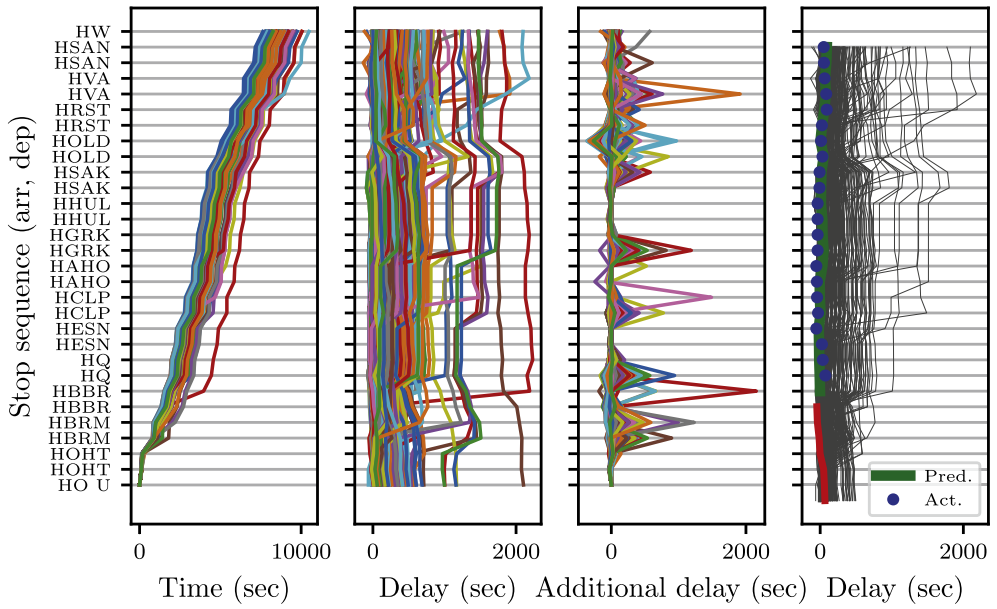

Fig. 7. Kernel plots (383 sample trajectories).

design and traffic assignment (Spiess and Florian, 1989). Current simulators have been shown to scale to nation-wide size (Osogami et al., 2012; Suzumura et al., 2012).

For the present purpose of predicting network-wide delays in near real-time, we use a variant of the mesoscopic integrated train and passenger simulation model described in Szabó et al. (2017). The model is fine enough to capture the most important interactions while at the same time allowing for a fast enough computation (about 2000× real-time on a desktop machine). To generate the predictions, a new simulation is run for the entire network in every minute, initialized by setting the train positions according to the most recent status messages received in near real-time.

A simulation model uses known physical or organizational properties and mechanisms of the socio-technical system under concern in order to describe the temporal evolution of the system's state via state-transition functions. In our case, the system objects are the individual trains and the railway network with its components (stations, control points and stretches) as described in Section 3.2. Trains move according to the timetable, trying to fulfill the scheduled arrival and departure times as good as possible subject to operational rules, such as minimum departure times and minimum dwell times at stations, minimum required connection times between trains and minimum headway times between trains entering the same stretch. Physical constraints also apply, such as minimum technical travel time on the tracks, and conflicting routes inside stations. Trains are considered as independent agents, moving at discrete time points in the network according to a pre-defined behavior model, and interacting with other agents such as network elements and trains. Between trains the interaction is indirect through the underlying network via track occupation conflicts and minimum headway requirements as well as through operational rules such as connections and rolling stock rotations. A simplified safety system is being used based on minimum headway times upon entering a stretch. From the data we observed that in the railway network considered this time is around 90 s, so we used this value as a strict minimum headway time along stretches. As neither track length data nor train velocity measurements were available, a simplified train movement model is used, where the travel time of a train between two stations is estimated as a certain quantile of the observed travel time distribution between the stations, and a constant speed is assumed along the stretch. The best quantile proved to be 40%. A possible explanation is that this value is close to, but somewhat smaller than the empirical median, as it best represents the free travel time on a stretch without being obstructed by stretch conflicts. Using the free travel time in the simulation allows delayed trains to continuously recover from delays in case no further conflicts arise. In this way, although in our model the travel speed does not explicitly depend on the delay, it is possible for train drivers to speed up and use time reserves to reduce delays.

The main simplification of our configuration compared to the usual simulation use case is that we do not consider passenger flows, because no occupancy data has been available in this setup. This means that only trains and network elements act as agents, but passengers do not, which leads to faster computation times. On the downside, delays caused by prolonged boarding and egress processes due to crowding could not be reflected in the model, but would be easy to integrate in the future once real-time information related to train occupancy becomes available.

Beside these simplifications, which were done to adjust the model to the available static and dynamic data, the model was also extended to reflect some key inter dependencies between trains that affect the propagation of train delays. Connections, which are significantly more important in long-distance railways than in the urban mass transit scenario considered in Szabó et al. (2017), have been accounted for by estimating a *connection matrix* between all possible pairs of trains at each station. If a connection is deemed significant, the most likely waiting time threshold was estimated and subsequently used in the simulation according to the frequently

used rule that a departing train waits for a delayed feeder train up to the given threshold. In a similar way also the train-to-train interactions due to rolling stock rotations were estimated and then simulated. That is, if a train departing at its first station uses the same rolling stock as a previously arriving train, then a delay is propagated. The same threshold principle as above was used, with the interpretation that beyond a certain incoming delay, the departing train will likely use a replacement rolling stock. The operational rules concerning connections and rolling stock rotations are usually known and could be explicitly modeled, but in line with our data-driven approach we estimated them from the historical movement data, using the current timetabling year. Nevertheless, we expect the prediction accuracy to increase when the actual rules and thresholds are used instead.

3.6. System aspects

Given the large number of models and predictions to be made, we briefly describe some of the systems considerations.

While the end goal of the pilot was for a online system, the tests reported in the paper were done in an offline manner with a controlled computational budget. This influenced some of the design choices: model forms in particular, since support for incremental training was considered to be important for deployment. The test system run times are indicative and could be further optimized, something we did not seek to do during this pilot.

We used distributed storage and processing engine (Hadoop + Spark) for data persistence and model training. Model training for the Random Forest used the standard MLLib library and was performed on three IBM Power 8 Servers, each with 24 cores, 192 hardware threads, and 1 TB RAM. In general hyper-parameter of random forest can be trained automatically using grid-search or Bayesian optimization approach (Snoek et al., 2012). However, our training data is very large (a few ten millions) where the random forest models for operational trains took roughly 20 h to train on this cluster. Moreover, we have more than 22 operational models needed to train so auto hyper-parameter tuning is not considered because of the large search space. Instead we consider manual tuning of two hyper-parameters: the number of trees in the forests and the training size. The other hyper-parameters such as the tree depth, the discrete bin-size, the data and feature bootstrapping hyper-parameters are kept as default values recommended by MLLib. We varied the training size in 5, 25, 50 and 100 millions and tree size in 5, 15, 25. Finally we choose training size as 50 millions and the number of trees as 15, as beyond this did not yield improved forecasts. Scoring for nationwide forecasts were done in roughly 5 min which fits the requirement for scoring response from the client.

Roughly 7400 kernel regression models were needed daily, based on the number of patterns in the schedule. These were constructed as needed and cached. On one IBM Power 7 server with 128 cores and 512 GB of RAM, a fresh complete model construction took roughly 25 min. Scoring for this model class is more expensive and took roughly 1 h for the tests. To demonstrate online feasibility, we experimented with a stream computing implementation using IBM Infosphere Streams that was able to score the models in roughly 8 min on a single compute node. The simulation models took roughly 1 h to train and 40 s to score. The ensemble mechanisms were also quick and performed in a few seconds.

4. Results

We performed rigorous quality assessments and studied quality metrics in the following dimensions. Forecast correctness C_k was measured as the fraction of forecasts within k minutes of the ground truth. We considered k as 1, 3 and 5 min. Forecast error was measured as the root mean squared error (RMSE) in seconds. The baseline results were based on schedule-based forecasts. Since forecast quality varies by forecast horizon, results were broken down by different time horizons. Additionally, the forecast quality was also measured for different train classes, and operational status. Particular focus was paid on the small sample of delayed trains.

To test forecast quality, a cut-off time during the day was established and the system fed with all the data up until that cut-off point. The models were trained based on all the data up until the previous day. Once the forecasts were generated, the ground truth was revealed and quality assessments made. A total of 12 testpoints were conducted. Table 3 shows a summary of selected metrics from one test point. Results were reasonably consistent across all the tests and presented results are representative. The baseline represents the schedule.

Forecast correctness was evaluated for component models and Ensembles. While overall, the Ensemble outperformed the constituent models, in some specific categories some component models outperformed the overall model. For example, Table 4 shows results for operational trains in different delay bins with the bold values showing best performing model. Simulation model results here show higher performance for larger delay buckets. This suggests that the linear weights could be further optimized. We tested combinations of different models.

One edge case for online prediction was during the cases when trains with long delays did not report their position. This can happen due to equipment failure for example when a train is stalled between control points. Since by design, a forecast is refreshed only when a train passing message is received, the delay estimated is out of date. The lack of position data is information that can be leverage to adjust the prediction. We simply shift the forecast by the difference between the current time and the last updated time.

There are several practical takeaways from these results. Since delay distributions, our target variable, vary considerably based on time horizon, train types, service classes, and current delay, performance measurement can be challenging. One set of methods is unlikely to perform well across all categories. Such characteristics can be only revealed with rigorous assessment of performance measures, which in turn must be defined by business/end-user needs. Quality variations in the forecast must be well understood before deployments in production.

The results suggest the greatest potential for improvements is in shorter term forecasts of operational trains using such data-driven models. This is since the context is better understood with sufficient data to better predict delay propagation. There are benefits for

Table 3
Forecast correctness (along with improvements over baseline) for a single test point.

Type	C_1		C_3		C_5		RMSE		N
ALL operational trains	55.3%	+16	81.5%	+17	89.5%	+10	331.34	−136	29762
ALL non-operational trains	60.5%	+7	81.1%	+5	87.8%	+2	407.65	−22	206351
ALL trains in horizon = < 10 min	83.9%	+37	96.9%	+22	98.9%	+12	110.25	−168	6746
ALL trains in horizon 10 – 30 min	64.5%	+19	88.9%	+15	95.3%	+9	147.75	−148	13830
ALL trains in horizon 30 – 90 min	54.4%	+8	80.8%	+9	90.0%	+5	268.84	−61	39874
ALL trains in horizon 90 – 180 min	55.3%	+7	77.0%	+5	84.4%	+3	504.38	−30	57384
ALL trains in horizon 180–end of day	62.4%	+6	82.1%	+4	88.1%	+2	401.38	−24	117283
ALL urban services	57.6%	+9	77.0%	+6	83.6%	+3	506.80	−30	83096
ALL night-services	62.2%	+7	84.2%	+6	91.0%	+3	294.02	−33	147381
ALL long-distance trains	31.7%	+0	62.3%	+6	75.5%	+7	785.26	−138	5636
Operational only in delay bin <6	63.0%	+16	91.4%	+13	97.4%	+2	106.08	−36	24701
Operational only in delay bin [6 – 10)	20.9%	+20	38.5%	+38	61.3%	+61	267.35	−190	2699
Operational only in delay bin [10 – 20)	16.9%	+16	31.1%	+31	43.7%	+43	470.15	−369	1582
Operational only in delay bin > = 20	8.6%	+8	18.8%	+18	30.9%	+30	1771.18	−584	780
All operational trains in SB class	58.7%	+27	81.4%	+20	89.1%	+13	369.94	−106	8533
All non-operational trains in SB class	57.4%	+7	76.5%	+5	83.0%	+2	520.17	−23	74563
All operational trains in NV class	58.1%	+14	84.5%	+16	91.4%	+9	201.38	−145	18494
All non-operational trains in NV class	62.8%	+6	84.2%	+5	90.9%	+2	305.01	−19	128887
All operational trains in FV class	25.7%	−1	61.8%	+9	77.9%	+12	702.47	−223	2735
All non-operational trains in FV class	37.2%	+3	62.8%	+3	73.2%	+3	856.01	−66	2901

Table 4
Ensemble improvements over constituent models for operational trains.

Delay bins	Cases	C_1	C_3	N
<6	Baseline	58.40%	87.10%	12926
	KR	76.60%	**97.56%**	12898
	Simulation	73.97%	95.84%	13082
	RF	72.88%	97.40%	12926
	Ensemble 1 (KR + RF)	77.12%	97.53%	13082
	Ensemble 2 (All)	**78.29%**	97.40%	13082
[6, 10)	Baseline	0.00%	0.00%	
	KR	16.47%	42.41%	613
	Simulation	**28.20%**	54.80%	624
	RF	26.80%	**60.69%**	608
	Ensemble 1 (KR + RF)	24.83%	57.69%	624
	Ensemble 2 (All)	27.88%	57.37%	624
[10, 20)	Baseline	0.00%	0.00%	
	KR	14.59%	36.90%	233
	Simulation	21.87%	50.78%	256
	RF	27.34%	**51.42%**	245
	Ensemble 1 (KR + RF)	26.95%	51.17%	256
	Ensemble 2 (All)	**27.73%**	50.00%	256
> = 20	Baseline	0.00%	0.00%	
	KR	0.00%	14.28%	14
	Simulation	14.28%	**71.42%**	14
	RF	7.69%	46.15%	13
	Ensemble 1 (KR + RF)	14.28%	50.00%	14
	Ensemble 2 (All)	**14.28%**	64.28%	14

medium and longer term, however these are less pronounced. While we initially scoped to do two-day ahead forecasts, predictions beyond 24 h were only marginally better than the schedule. Some services were harder to predict, long distance services and trains with exceptional delays. Additional methodological considerations are needed for such cases. Overall, the results demonstrate the value of big data and machine learning for railway systems when applied with domain expertise.

5. Conclusions

A large-scale, data-driven ensemble forecasting system for train delays is outlined in this paper. The results, based on tests for the Deutsche Bahn passenger services network show high fidelity forecasts. Our evaluations show that the forecast accuracy depends on a range of operational characteristics such as service type and current delay for operational trains. Ensembles performed overall better than constituent models as expected.

On the practical side, several pre-processing steps were necessary before the models were trained. Primarily routines for network infrastructure inference, de-noising input data, and efficient feature generation routines were required.

Several enhancements can be considered for future work. While our initial goal was to create a general prediction model for all cases, the idea of state-dependent models could be explored, i.e. delayed trains have separate class of models. Delayed trains are a very small fraction of the overall data and demonstrate a different empirical distribution. A state-dependent pipeline may yield significant improvements over a general specification.

For operational trains, our *n*-stop ahead model was based on studying the distribution of delay, alternative specifications (e.g. per segment level models) could also be considered.

The model performance is sensitive to hyper-parameters, such as outlier thresholds and the kernel bandwidth. While we did tuning these, a holistic approach to set these parameters is likely to be beneficial.

Acknowledgment

We thank Mathieu Sinn and Jonathan Epperlein for reviews of an earlier draft.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.trc.2019.04.026.

References

Andres, M., Nair, R., 2017. A predictive-control framework to address bus bunching. Transport. Res. Part B: Methodol. 104, 123–148.
Balmer, M., Meister, K., Rieser, M., Nagel, K., Axhausen, K.W., 2008. Agent-based simulation of travel demand: Structure and computational performance of MATSim-T. ETH, Eidgenössische Technische Hochschule Zürich, IVT Institut für Verkehrsplanung und Transportsysteme.
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. Transport. Res. Part C: Emerg. Technol. 93, 211–227.
Berger, A., Gebhardt, A., Müller-Hannemann, M., Ostrowski, M., 2011. Stochastic delay prediction in large train networks. In: OASIcs-OpenAccess Series in Informatics. Vol. 20. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, pp. 100–111.
Corman, F., Kecman, P., 2018. Stochastic prediction of train delays in real-time using bayesian networks. Transport. Res. Part C: Emerg. Technol. 95, 599–615.
Fellendorf, M., 1994. VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority. In: 64th Institute of Transportation Engineers Annual Meeting, Dallas, TX. pp. 1–9.
Ghofrani, F., He, Q., Goverde, R.M., Liu, X., 2018. Recent applications of big data analytics in railway transportation systems: a survey. Transport. Res. Part C: Emerg. Technol. 90, 226–246.
Gorman, M.F., 2009. Statistical estimation of railroad congestion delay. Transport. Res. Part E: Logist. Transport. Rev. 45 (3), 446–456.
Goverde, R.M., 2010. A delay propagation algorithm for large-scale railway traffic networks. Transport. Res. Part C: Emerg. Technol. 18 (3), 269–287.
Kecman, P., Goverde, R.M., 2013. An online railway traffic prediction model. In: RailCopenhagen2013: 5th International Conference on Railway Operations Modelling and Analysis, Copenhagen, Denmark, 13-15 May 2013. International Association of Railway Operations Research (IAROR).
Khadilkar, H., 2016. Data-enabled stochastic modeling for evaluating schedule robustness of railway networks. Transport. Sci. 51 (4), 1161–1176.
Lulli, A., Oneto, L., Canepa, R., Petralli, S., Anguita, D., 2018. Large-scale railway networks train movements: a dynamic, interpretable, and robust hybrid data analytics system. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). IEEE, pp. 371–380.
Marković, N., Milinković, S., Tikhonov, K.S., Schonfeld, P., 2015. Analyzing passenger train arrival delays with support vector regression. Transport. Res. Part C: Emerg. Technol. 56, 251–262.
Müller-Hannemann, M., Schnee, M., 2009. Efficient timetable information in the presence of delays. Robust Online Large-Scale Optim. 5868, 249–272.
Nabian, M.A., Alemazkoor, N., Meidani, H., 2019. Predicting near-term train schedule performance and delay using bi-level random forests. Transp. Res. Rec.
Nair, R., Bouillet, E., Gkoufas, Y., Verscheure, O., Mourad, M., Yashar, F., Perez, R., Perez, J., Bryant, G., 2014. Data as a resource: real-time predictive analytics for bus bunching. In: Proceedings of the Annual Meeting of the Transportation Research Board.
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 Trans. Syst., Man, Cybernet.: Syst. 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 Res. 11, 54–64.
Opitz, D., Maclin, R., 1999. Popular ensemble methods: an empirical study. J. Artif. Intell. Res. 11, 169–198.
Osogami, T., Imamichi, T., Mizuta, H., Morimura, T., Raymond, R., Suzumura, T., Takahashi, R., Ide, T., 2012. IBM Mega traffic simulator. IBM Res., Tokyo, Japan, IBM Res. Rep. RT0896.
Rokach, L., 2010. Ensemble-based classifiers. Artif. Intell. Rev. 33 (1–2), 1–39.
Sinn, M., Yoon, J.W., Calabrese, F., Bouillet, E., 2012. Predicting arrival times of buses using real-time gps measurements. In: Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on. IEEE, pp. 1227–1232.
Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms. In: In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc., pp. 2951–2959. http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.
Spiess, H., Florian, M., 1989. Optimal strategies: a new assignment model for transit networks. Transport. Res. Part B: Methodol. 23 (2), 83–102.
Suzumura, T., Kato, S., Imamichi, T., Takeuchi, M., Kanezashi, H., Ide, T., Onodera, T., 2012. X10-based massive parallel large-scale traffic flow simulation. In: Proceedings of the 2012 ACM SIGPLAN X10 Workshop. ACM, pp. 3.
Szabó, J., Blandin, S., Brett, C., 2017. Data-driven simulation and optimization for incident response in urban railway networks. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. AAMAS'17. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 819–827.

UIC, June 2009. Assessment of the performance of the network related to rail traffic operation for the purpose of quality analyses – delay coding and delay cause attribution process (450-2). Tech. rep., UIC.
Wang, R., Work, D.B., 2015. Data driven approaches for passenger train delay estimation. In: Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on. IEEE, pp. 535–540.
White, T., 2005. Alternatives for railroad traffic simulation analysis. Transport. Res. Rec.: J. Transport. Res. Board 1916, 34–41.