Large-Scale Railway Networks Train Movements: a Dynamic, Interpretable, and Robust Hybrid Data Analytics System.

Alessandro Lulli, Luca Oneto, *Member, IEEE*,
Renzo Canepa, Simone Petralli, and Davide Anguita, *Senior Member, IEEE*

***Abstract*— We investigate the problem of analyzing the train movements in Large-Scale Railway Networks for the purpose of understanding and predicting their behaviour. We focus on different important aspects: the *Running Time* of a train between two stations, the *Dwell Time* of a train in a station, the *Train Delay*, and the *Penalty Costs* associated to a delay. Two main approaches exist in literature to study these aspects. One is based on the knowledge of the network and the experience of the operators. The other one is based on the analysis of the historical data about the network with advanced data analytics methods. In this paper, we will propose an hybrid approach in order to address the limitations of the current solutions. In fact, experience-based models are interpretable and robust but not really able to take into account all the factors which influence train movements resulting in low accuracy. From the other side, Data-Driven models are usually not easy to interpret, nor robust to infrequent events, and require a representative amount of data which is not always available if the phenomenon under examination changes too fast. Results on real world data coming from the Italian railway network will show that the proposed solution outperforms both state-of-the-art experience and Data-Driven based systems in terms of interpretability, robustness, ability to handle non recurrent events and changes in the behaviour of the network, and ability to consider complex and exogenous information.**

***Index Terms*— Railway Network, Train Movements, Running Time, Dwell Time, Train Delays, Penalty Costs, Experience-Based Models, Data-Driven Models, Hybrid Models, Interpretable Models**

I. Introduction

Railway Transport Systems (RTSs) play a crucial role in servicing the global society and the transport backbone of a sustainable economy. A well functioning RTS should met the requirements defined in the form of the 7R formula [1], [2]: Right Product, Right Quantity, Right Quality, Right Place, Right Time, Right Customer, and Right Price. Therefore, an RTS should provide: (i) availability of appropriate products (the provisioning of different categories of train), (ii) proper number of executed transportation tasks (enough trains to fulfill the request), (iii) proper quality of execution of transportation tasks (safety, correct scheduling, and effective conflicts resolution), (iv) right place of destination according to a timetable (correct transportation routes), (v) appropriate lead time (reduced *Train Delays*), (vi) appropriate recipients (focused on different customer needs and requirements), and (vii) appropriate price (both from the point of view of the customers and the infrastructure managers).

Alessandro Lulli, Luca Oneto, and Davide Anguita are with the DIBRIS, University of Genoa, Via Opera Pia 13, I-16145, Genoa, Italy (email: {alessandro.lulli,luca.oneto,davide.anguita}@unige.it). Renzo Canepa and Simone Petralli is with Rete Ferroviaria Italiana S.p.A., Via Don Vincenzo Minetti 6/5, I-16126, Genoa, Italy (email: {r.canepa,s.petralli}@rfi.it).

In this work we focus on the problem of analyzing the train movements in Large-Scale RTSs for the purpose of understanding and predicting their behaviour. Hence, we will study four important aspects: the *Running Time*, the *Dwell Time*, the *Train Delay*, and the *Penalty Costs*. The first one is the amount of time a train spends in travelling between two consecutive stations. The second one is the amount of time a train spends in a station. The third one is the difference between the actual arrival (or departing time) and the scheduled one in each of the stations composing the itinerary of a train. Finally, the fourth one is the penalty that the Infrastructure Managers (IMs) and the Train Operators (TOs) have to pay because of the delays in proportion to their responsibilities. These aspects are of paramount importance in the context of an RTS. Study them, and being able to predict their behaviour, allows to improve the quality of service, the train circulation, and the IMs and TOs management costs. More specifically, in relation with the 7R formula, it allows to improve the Right Quantity (improving circulation improves the network capacity without requiring massive public investments in new physical assets), the Right Quality (it helps the operators to understand how much a train needs from one checkpoint to another, to provide a timely resolution of the conflicts on the network and, to correctly schedule all the trains), the Right Time (efficiently predict the train transits improves the ability of the operators to maintain the correct train circulation), and the Right Price (it helps to minimize the penalties for the IMs and TOs).

A large literature covering the aforementioned problems already exists [3]. However, the majority of the works focus just on a single aspect of the train movements. The *Running Time* and *Dwell Time* have been exploited mainly to retrieve train positions and track occupations [4], [5], or to detect train conflicts [6], or to perform a correct dispatching [7]–[9]. The *Train Delay* prediction is the most investigated aspect of train movements [10]–[17]. Some works study how the *Train Delays* propagate in subsequent stations [18], for online track conflict predictions [15], and for deriving dependencies between trains [19], [20]. For what concern the study and the prediction of the *Penalty Costs* in [21] it has been studied the relation between *Penalty Costs* and *Train Delays* in the Britain's railway.

To the best knowledge of the authors, there is no work in the literature which deals comprehensively with all the aspects of the train movements as we will propose in this work.

From a methodological point of view, the models adopted in literature to solve the train movements related problems

can be grouped in two categories [3]. Models in the first category, called Experience-Based Models (*EBM*s), attempt to exploit the knowledge of the network in order to derive a model which takes into account the physical characteristics and limitations of the network (e.g. speed limits, usury, and slopes) and the trains (e.g. acceleration, weight, and number of wagons) together with the experience of the operators [3], [15], [19], [22]–[24]. Models in the second category, called Data-Driven Models (*DDM*s), are based on the analysis of the historical data about the network coming from the most recent Railway Information System with advanced analytic methods [3], [5], [25]. Both *EBM*s and *DDM*s have strengths and weaknesses. *EBM*s are usually low computational demanding, easy to interpret, and robust. A the same time, *EBM*s are usually not very accurate, hard to modify in order to contemplate complex phenomena (e.g. congestion of the network and weather conditions), and not dynamic (they tents to oversimplify the phenomenon not taking into account behaviour's drifts). On the other side, *DDM*s are much more accurate but they are also much more computational demanding (at least for building them and sometimes also for making predictions), often not easy to interpret (interpretability in learning from data is a crucial issue nowadays), not really robust (they do not handle well infrequent events), and not very dynamic (if the phenomena under examination change too fast with respect to the possibility to collect enough data about it).

For this reasons, in this work we propose an hybrid approach, that we will call Hybrid Model (*HM*), taking the best from *EBM*s and *DDM*s. In particular, the proposed *HM* will be interpretable (the *HM* will be easy to understand from an operator point of view), robust and dynamic (*HM* will handle well both infrequent events, like the passage of Freight trains, and fast changes of the train movements phenomena, like a timetable modification), easily extensible (it will be able to take into account complex phenomena like the congestion of the network and exogenous factors like the weather conditions), and able to take into account the knowledge about the network and the experience of the operators.

The rest of the paper is organized as follows. Section II describes the RTS train movements related problems. Section III focuses the attention on the particular case of the Italian RTS. Section IV presents the actual *EBM* and *DDM* exploited in the Italian RTS. In Section V we present our contribution: the *HM*. In Section VI we compare the performance of the *HM* against the *EBM* and the *DDM* on a set of real world data provided by Rete Ferroviaria Italiana (the Italian IM) showing the effectiveness of the proposed approach both in terms of dynamicity, interpretability, and robustness. Section VII concludes the paper.

II. Problem Description

A railway network can be easily described with a graph. Figure 1 depicts a simplified railway network where a train follows an itinerary characterized by a station of origin (station *A*), a station of destination (station *F*), some stops

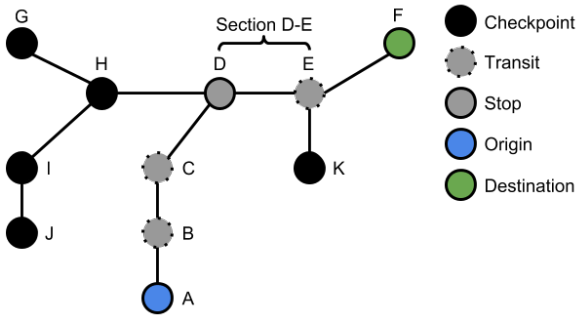

Fig. 1. A railway network. The itinerary of a train is depicted with the grey nodes where *A* is the origin station and *F* is the destination.

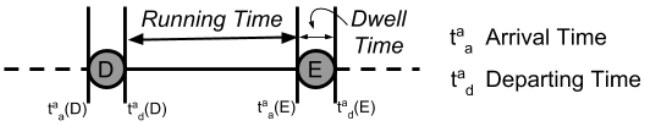

Fig. 2. *Running Time* and *Dwell Time*.

(stations *A*, *D*, and *F*) and some transits (checkpoints *B*, *C*, and *E*).

We call checkpoint a station without differentiating between a station where the train stops or transits and between actual stations and points of measure. In fact, not all checkpoints are actual stations since in long railway sections it is often needed to add a point of measure for following the trains with a better granularity.

The railway sections are the pieces of the network between two consecutive checkpoints, note that railway sections have also an orientation (e.g. transit *D* to *E* is different from transit *E* to *D*).

For any checkpoint in the itinerary, the train is scheduled to arrive and depart at different specified times, defined in the timetable, respectively t^s_a and t^s_d. Usually, the time references included in the timetable are approximated with a precision of 30s. The difference between the scheduled time and the actual time, either for arrival (t^a_a) or for departure (t^a_d), is defined as *Train Delay*. If the delay is greater than 30s, then a train is considered as delayed. Note that, for the origin station there is no arrival time, while for the destination station there is no departure time. We define the *Running Time* as the amount of time needed to depart from the first of two subsequent checkpoints and to arrive to the second one (see Figure 2, for railway section *D* to *E* the scheduled *Running Time* is $t^s_a(E) - t^s_d(D)$ while the actual *Running Time* is $t^a_a(E) - t^a_d(D)$) and the *Dwell Time* is the difference between the departure time and the arrival time in a fixed checkpoint (see Figure 2, in checkpoint *D* the scheduled *Dwell Time* is $t^s_d(D) - t^s_a(D)$ and the actual *Dwell Time* is $t^a_d(D) - t^a_a(D)$).

Furthermore, each train has an unique identifier from which it is possible to retrieve the category of the train (e.g. Regional, Freight, and High Speed). Analogously, each checkpoint has an unique identifier from which it is possible to retrieve the category of the network (e.g. Node,

High Speed, and Second Complementary Network). Train, network category, time of the day, and other factors allow to compute the Penalty Costs associated to a delayed train.

Based on these definitions, it is possible to describe the train movements related prediction problems that we will face in this work.

A. Running Time and Dwell Time Prediction

The prediction of the *Running Time* and *Dwell Time* are the first problems that we address. For a specific train, the problem is to predict the *Running Time* for all the subsequent railway sections it will traverse and the *Dwell Time* for all the subsequent checkpoints in which it will stop, updating these predictions every time it reaches the next checkpoint. Providing an accurate prediction of the *Running Time* and the *Dwell Time* allows to provide to the operators a clear understanding of how much time a train needs to complete the itinerary. Moreover, as we will describe later, the *Running Time* and the *Dwell Time* predictions can be exploited as a building block for the *Train Delay* predictors (see the *EBM* of Section IV-A).

B. Train Delay Prediction

The *Train Delay* prediction is the problem of forecasting the arrival and departing delay of a train for all the subsequent checkpoints in its itinerary, updating this predictions every time it reaches a new checkpoint. The prediction of the future delays is a problem of paramount importance and yields several benefits: a reliable information for the passengers currently on the trains or waiting in a checkpoint, a better exploitation of the railway network while maintaining the safety of the passengers and avoiding resource conflicts, better train rescheduling and dispatching, and more.

Note that, *Train Delay* prediction can be seen as a standalone task (see the *DDM* of Section IV-B) or it ca be retrieved from the combination of the *Running Time* and the *Dwell Time* predictions (see the *EBM* of Section IV-A).

C. Penalty Costs Prediction

In an RTS the IMs and the TOs have to pay penalties, when trains are delayed, in proportion to their actual responsibilities. For this reason, predicting the *Penalty Costs* is a strategic issue: an effective prediction system can be exploited to choose the best dispatching solution which minimizes both *Train Delays* and *Penalty Costs*. However, this problem is rather complex since the *Penalty Costs* computation is the result of a complex procedure that has to be fully understood.

Currently, in every State a document of management principles (for example, in Italy is the PIR[1]) defines the rules, agreed between the State, the IMs (e.g. Rete Ferroviaria Italiana is an Italian IM), and the TOs (e.g. Trenitalia is an Italian TO), that must be followed to solve the conflicts when one or more trains are delayed and the associated *Penalty Costs* that IMs and TOs have to pay based on their responsibilities. Such rules define the level of priority of each train based on different variables such as the category of the train and time of the day. For instance, during the daily commuter time slot, some Regional trains could have the same priority as the High Speed trains, even if the latter have usually higher priority. In order to enforce the IMs to follow these rules, if a train is delayed, the priorities also influence the *Penalty Costs* associated to a *Train Delay*. Consequently, in order to compute the *Penalty Costs*, it is required to retrieve a series of information regarding the trains and their itinerary. Although a deterministic relation exists to compute the penalties, not all this variables are known at the time of the train transit. The final penalty is usually agreed after the train have completed its journey (even after months). For example, the percentage of responsibility may be the results of a legal dispute between the IM and the TO.

[1] http://www.rfi.it/rfi/SERVIZI-E-MERCATO/Accesso-alla-rete/Prospetto-informativo-della-rete

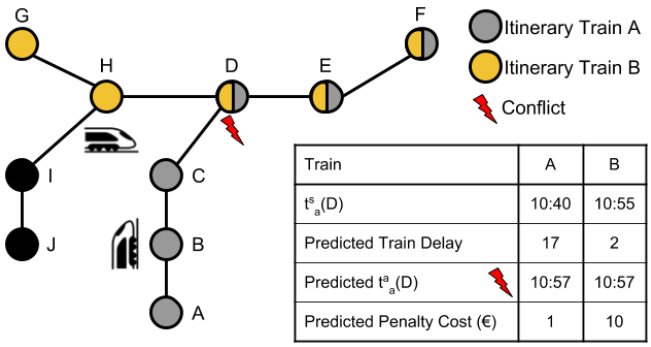

Fig. 3. Handling the conflicts using *Train Delay* and *Penalty Costs* prediction models. Exploiting the *Penalty Costs* prediction would result in stopping *Train A* because is less expensive for the IM. Exploiting, instead the delay would resort in stopping *Train B* for reducing the grater delay of *Train A*.

More in details, the *Penalty Costs* is the result of a deterministic combination of the following variables:

- the category of the train (e.g. a train belonging to the market service delayed of one hour has larger *Penalty Costs* than a Freight train affected by the same delay);
- the operational category of the train (e.g. if the itinerary is scheduled in the timetable, or it is created/modified in the last few days before the actual train transit);
- the type of railway section (similarly to the category of the train, the High Speed Lines are affected by a higher *Penalty Costs*);
- the amount of delay of the train (e.g. the average and maximum delay for Regional trains, or just the delay in the final checkpoint for Freight trains);
- the percentage of responsibility of the IMs, of the TOs, and of the exogenous factors (e.g. flooding and strikes).

D. Example

In this section, we present an example to show the usefulness of the predictive models described above.

Let us suppose to have two trains travelling the simplified railway network depicted in Figure 1, with two different itineraries as depicted in Figure 3. The first train, *Train A*, travels along its gray itinerary from checkpoint *A* to *F*, while the second one, *Train B*, travels its yellow itinerary from *G* to *F*. The two trains share three checkpoints in their itineraries

(checkpoint *D*, *E*, and the destination *F*). The timetable has been constructed in order to give the correct headway to the trains for safety and regularity purposes. Suppose also that *Train A* is in checkpoint *B*, and that *Train B* is in checkpoint *H*.

Exploiting the *Train Delay* predictor, we discover that both trains will arrive at approximately the same time in checkpoint *D*, leading to a conflict. Then, we have to decide which one of the two trains should have the priority over the other. Exploiting just the *Penalty Costs* prediction would result in stopping the *Train A* because is less expensive for the IM, while exploiting just the *Train Delays* prediction would resort in giving the priority to the *Train A* for reducing its grater delay. Considering instead both *Penalty Costs* and *Train Delays* predictions, would result in a more aware decision. In this case, the most reasonable solution is to stop *Train A* since it will negligibly increase its delay (few additional minutes) to make *Train B* go forward, possibly regaining some delay which is instead very costly for the IM (so, probably, it is a more important train).

III. The Italian Railway Transportation System

In this paper, we consider the specific case of the Italian RTS, which is substantially handled by just one IM, Rete Ferroviaria Italiana (RFI), which provided to us both the knowledge of the network and the data needed for this study.

According to the *International Union of Railways*, the Italian RTS is in the Top 3 and the Top 10 largest RTS respectively in Europe and Worldwide. RFI controls every day $\approx$10.000 trains travelling along the national railway network of $\approx$25.000km. Every train is characterized by an itinerary composed of an average of $\approx$12 checkpoints. This means that the number of train movements is greater than or equal to $\approx$300.000 per day. This results in more than one message per second and more than 10GB of messages per day to be stored.

Note that, every time a message describing the itinerary of a particular train is retrieved, the predictive models can take advantage of this new information both to make better predictions and to updated the model itself. This allows to have always the best performing models which exploits all the available information, and to follow the effects of small or big changes in the timetables that occur during the year.

Apart from the daily messages of the train movements, RFI is also able to provide all the information about the travelling trains and network characteristics needed to compute the *Penalty Costs* according to the PIR (see Section II-C).

Finally, other exogenous information regarding the network can be retrieved from many Italian freely available data sources which can help in improving the accuracy of the *DDM*s. In this work, we will take into consideration the weather information (see e.g. [26], [27]) since in previous works it has been shown to be an effective solution for improving the *DDM* accuracy [25].

IV. The Actual Systems

This section describes two different state-of-the-art approaches employed in RFI to tackle the problems described in this paper. In particular, RFI exploits both a *EBM* which is quite similar to the one described in [15] (although the latter includes process mining refinements which potentially increase its performance) and a *DDM* [25] that produces better predictions of the *Train Delays* with respect to *EBM*.

A. The Actual Experience-Based Model

The actual RFI *EBM* performs the predictions based on the knowledge of the railway network and the experience of the operators. It focuses mostly on the problem of predicting the *Running Time*. The *Dwell Time* is considered fixed to the difference between the scheduled departure and arrival time in a station. The *Train Delays* and the *Penalty Costs* are derived from the predicted *Running Times* and the fixed *Dwell Times* assuming that the percentage of responsibility for a delay is always 100% of RFI.

More in details, the idea of the *EBM* is to analyze the amount of time that a train needs to traverse each railway section of the network, taking into account the speed limits, the state of the network, the type of train etc. The timetables are produced taking in consideration such physical constraints and a working margin is kept for dealing with delays. Then, for each railway section and each train category, a coefficient, called *Gaining Time*, is computed which represents the time that be can regained in case of delay (the *Gaining Time* takes into account also a possible smaller *Dwell Time*). The *Gaining Time* is static, i.e. it does not change based on the state of the network, weather conditions, etc. The *Gaining Time*, is exploited to solve the *Train Delay* prediction problem. When predicting a delay, it is assumed that a delayed train is always able to regain, in a given railway section, an amount of time equals to its *Gaining Time*. Then, when RFI predicts the *Train Delay* in a subsequent checkpoint it subtracts from the current delay all the *Gaining Times* of the railway sections between the actual station until the considered checkpoint. Once the delay is computed, the *Penalty Costs* can be derived straightforward if, as in RFI, it is assumed that 100% of the delay costs is to impute to RFI, thanks to the deterministic formula that can be found in the PIR.

The *Gaining Times* of the RFI *EBM* do not depend on the time of the days, on the fact that it is a weekend or a weekday, on the train actual delay, on the network congestion, on the weather conditions since no easy relation can be retrieved. On the other side, the RFI *EBM* is quite robust and easy to understand from an operator perspective even if not very accurate and dynamic.

B. The Actual Data-Driven Model

Given the low accuracy of the *EBM*, in RFI it has been decided to exploit also the *DDM* developed in [25]. The *DDM* does not take into account the knowledge of the railway network nor the experience of the operators, but it is based just on the historical data about train movements, weather conditions, and weather forecasts. For this purpose, the *DDM* exploits advanced analytic methods able to extract accurate models of the future behaviour of each train. The

advantage of these methods is that there is no need of any a-priory knowledge of the underline physical system but, most of the time, they produce non-parametric models that are not easy to interpret nor supported by any physical intuition or interpretation. Moreover, in general, a great amount of historical data is needed in order to build an accurate model and it is not so easy to make these systems strongly dynamics. In fact, if for example the timetable changes, they require at least one month of data before achieving a reasonable accuracy.

The RFI *DDM* is composed of many *DDM*s that, working together, make it possible to perform a regression analysis on the past delay profiles in order to predict the future ones. In particular, for each train and for each checkpoint composing its itinerary, a set of *DDM*s is built to predict the delay in all the subsequent checkpoints. Consequently, the total number of *DDM*s to be built for each train is $\approx n(n-1)$ where n is the number of checkpoints visited by the train. These *DDM*s work together to estimate the delays of a particular train during its entire journey. For a single train, every time it arrives at (departs from) a specific checkpoint included in its trip, the *DDM*s take as inputs its previous sequence of arrival and departure *Train Delays*, Running Times, and *Dwell Times* to predict delay for all the subsequent checkpoints. These *DDM*s are also able to take into account the state of the congestion of the network and other exogenous variables (e.g. the weather information) [25]. The *DDM*s can be built using many different learning algorithms, exploiting the Random Forest (RF) usually leads to better results [28].

Unfortunately, the RFI *DDM* has some drawbacks. Many historical information about the trains are requested before performing the prediction, otherwise it perform badly (e.g. on new trains or after changes in the timetable). Moreover, each model composing the *DDM* is specific for one particular train and checkpoint limiting its interpretability on a larger scale (it cannot group similar trains or trains in the same category) and the complexity of the *DDM* is higher with respect to *EBM* (too many models to build). Finally, the *DDM* does not integrate the knowledge and the experience of the operators nor gives to the operators an interpretation of the *Train Delay* phenomenon.

V. The Proposed Hybrid Model

In this work, we propose an *HM* to perform the *Running Time*, *Dwell Time*, *Train Delay*, and *Penalty Costs* predictions, merging together the *EBM* and the *DDM* to exploit their strengths and limiting their weaknesses. The goal is to build accurate, dynamic, robust, and interpretable models able to provide insights for both solving the train conflicts and minimizing the *Train Delays* and the *Penalty Costs*.

Similarly to the *EBM*, the *HM* relies, on the top, on an interpretable model able to encapsulate the experience of the operators in the form of a decision tree and, at the bottom, the leafs, instead of being defined relying on the physical knowledge of the network as in the *EBM*, are constructed following the ideas of the *DDM* where the historical data about the network and other exogenous information (e.g. weather) are exploited via advanced analytic methods. Moreover, contrary to the *DDM*, the *HM* does not implement one model for each train and, contrary to the *EBM*, the *HM* does not groups all the trains just based on their category and railway section. In fact, the *HM* groups the trains based on a series of similarity variables, defined together with the RFI operators, which allow to have, from one side, robust statistics, thanks to the possibility to learn from a reasonable group of train, but also a rich feature set, to be able to capture the variability of the phenomena. The proposed *HM* is then able to be extremely dynamic: grouping the trains increases the number of historical data to exploit during the leaf creation and follow, in a reasonable amount of time, timetable changes and new train schedules, thanks to the robustness introduced by the *HM* experience based top level structure.

We exploited the above mentioned approach for predicting both the *Running Time* and the *Dwell Time*. For what concern the *Train Delay*, instead, we opted for the same solution of the actual RFI *EBM* (see Section IV-A). In fact, in order to predict the *Train Delay* at a desired subsequent checkpoint, we sum all the needed *Running Time* and *Dwell Time* predictions to the current train time and then we compute the difference between the estimated and the scheduled train time. Finally, in order to predict the *Penalty Costs*, we made use of the *HM* described in the previous paragraph to predict an auxiliary variable, the *Responsibility*, which is the percentage of responsibility of the IM for the delays. Then, combining the *Train Delay* and the *Responsibility* predictions, we were able to predict the *Penalty Costs* exploiting the deterministic relation described in the PIR.

The work has been conduced side by side with the RFI operators taking into account their needs and their working environment which is constrained, in terms of complexity of the solution, to something that can provide simple and effective insights.

In the subsequent subsections, we will first present in details how we constructed the above mentioned *HM* decision tree based top structure and its Data-Driven based bottom structure (see Section V-A), and then we will describe how this *HM* has been exploited for predicting the *Running Time*, the *Dwell Time*, the *Train Delay*, and the *Penalty Costs* (see Section V-B).

A. Hybrid Decision Tree

As described before, the *HM* exploits, as a basic structure, a top level experience based decision tree and a bottom level Data-Driven model which is able to easily take into account the network congestion state and other exogenous information, like the weather conditions, which are not easy to model with the experience. The top level structure can be easily adapted to the prediction task under examination. For instance, for the *Running Time* we are interested in considering each railway section separately, instead for the *Dwell Time* prediction it is better to differentiate each of the checkpoints. The variables that we consider in the top level structure, defined with the RFI experts, are a subset of the ones reported in Table I. Then, as leafs of the tree, instead of

TABLE I

DESCRIPTION OF THE *HM* TOP LEVEL DECISION TREE FEATURE SET.

Feature Name	Categorical	Description
Railway Section	Yes	The considered railway section
Railway Checkpoint	Yes	The considered railway checkpoint
Train Type	Yes	The considered Train Type
Daytime	No	The time of the day with an hourly granularity
Weekday	Yes	The day of the week
Last Delay	No	The last known delay with the following granularity in minutes ($[0, 2]$, $(2, 5]$, $(5, 10]$, $(10, 20]$, $(20, 30]$, $(30, 60]$, $(60, 120]$, $(120, \infty)$);
Weather Conditions	Yes	The weather conditions (Sunny, Light Rain, Heavy Rain, Snow).

TABLE II

DESCRIPTION OF THE *HM* BOTTOM LEVEL RF FEATURE SET.

Feature Name	Categorical	Description
Weather Information	Yes	Weather condition (Sunny, Light Rain, etc.) in all the checkpoints of the train itinerary (for the already traveled checkpoints we use the actual weather while for the future checkpoints we use the predicted weather conditions)
Past *Train Delays*	No	Average value of the past *Train Delays* in seconds & Last known *Train Delay*
Past *Dwell Times*	No	Average value of the past differences between actual and scheduled *Dwell Times* in seconds & Last known difference between actual and scheduled *Dwell Time*
Past *Running Times*	No	Average value of the past differences between actual and scheduled *Running Times* in seconds & Last known difference between actual and scheduled *Running Time*
Network Congestion	No	Number of trains traversing the checkpoints of the train itinerary in a slot of 20 minutes around the actual and scheduled times respectively for the past and future checkpoints
Network Congestion Delays	No	Average *Train Delay* of the trains traversing the checkpoints of the train itinerary in a slot of 20 minutes around the actual and scheduled times respectively for the past and future checkpoints

plugging an estimate of the quantity that we want to predict based on the experience of the operators and the knowledge of the network, we exploit a Data-Driven model able to learn from the historical data regarding all the trains which fall in that particular leaf (basically all trains which share similar characteristics and itinerary) plus additional complex features. In particular, each leaf is a RF regressor [28] (following the experience of the *DDM* developed in [25]), which predicts the quantity that we want to estimate based on a series of features designed with the RFI experts and based also on the lesson learned with the *DDM* [25]. These feature set is reported in Table II

The whole *HM* is constructed and updated incrementally as soon as a new train movement is recorded. In the top level decision tree, a new leaf is added each time we record a new train movement which belongs to a previously unexplored branch of the decision tree. Then, the RF regressor in the leaf is learned based on all the past train movements which fall in that particular leaf. In order to follow the changes in behaviour of the phenomena we forgot the train movements older than three months. The predictions phase, instead, is simpler: we just visit the tree considering the information that we want to predict and we exploit the correct RF regressor to make the actual prediction.

As described at the beginning of Section V, the *HM* will be exploited for predicting:

- the *Running Time*: in this case we exploit, in the *HM* top level decision tree, all the variables of Table I except the one relative to the checkpoints since *Running Time* is a property of the railway sections and do not depend on the checkpoints;
- *Dwell Time*: in this case we exploit, in the *HM* top level decision tree, all the variables of Table I except the one relative to the railway sections since *Dwell Time* is a property of the checkpoints and do not depend on the railway sections;
- *Responsibility*: in this case we exploit, in the *HM* top level decision tree, all the variables of Table I.

Figure 4 depicts an example of use of the *HM* for the *Running Time* prediction problem. As one can see from Figure 4, every time a new movement is recorded the *HM* is updated based on the information inside the train movement record and we exploit this new information about the travel of the train to update all the predictions about the subsequent railway sections.

B. Train Movements Predictors via Hybrid Model

In this section we describe how the previously described *HM* has been exploited for predicting the *Running Time*, the *Dwell Time*, the *Train Delay*, and the *Penalty Costs*.

1) Running Time Prediction: In this case we apply the *HM* described in Section V-A and we directly predict the values of the *Running Times*. Every time a train movement is recorded, the model and the predicted future *Running Times* are updated based on this new information.

2) Dwell Time Prediction: Regarding the *Dwell Time* prediction we exploit exactly the same approach described for the *Running Time* prediction. Note that, the only difference between the two models stays in the feature set of the *HM* top level decision tree (see Section V-A).

3) Train Delay Prediction: In order to predict the *Train Delays*, instead of building another *HM*, we exploit, similarly to the *EBM*, the *Running Time* and *Dwell Time* predictors as building blocks. Each time a prediction is required, we predict all the *Running Times* of the sections and all the *Dwell Times* of the checkpoints between the current checkpoint and the one for which we request the *Train Delay* prediction. Then, the desired result is obtained by summing all these

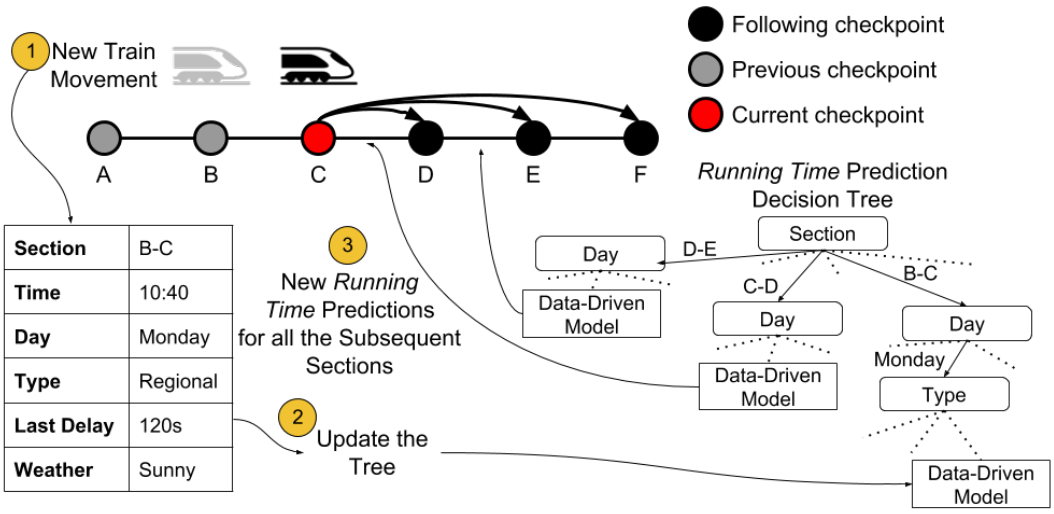

Fig. 4. The proposed *HM* for the *Running Time* prediction: updating the model every time a new movement is recorded and predicting the future *Running Time* in the subsequent sections.

times to the current time and subtracting from the results the scheduled time.

4) Penalty Costs Prediction: In order to compute the *Penalty Costs* of a particular *Train Delay*, we have to combine two quantities. First, we obtain the predicted *Train Delay* exploiting the approach described in Section V-B.3. Then, we predict the *Responsibility* with a new *HM* as described in Section V-A. Once these two predictions are available, we combine them with the deterministic relation described in the PIR, obtaining the *Penalty Costs* prediction.

Specifically, we compute the *Penalty Costs* P of a train as follows:

$$P = P_U \sum_{j \in \mathcal{I}} m_j r_j C_T C_N C_D, \tag{1}$$

where P_U is the unitary costs, $\mathcal{I}$ is the set of checkpoints composing the itinerary, m_j are the minutes of delay at section j, r_j is the percentage of responsibility for section j, C_T is coefficient relative to the type of the train T, C_N is the coefficient relative to the type of railway network N, and C_D is a coefficient which depends on the average and maximum delay registered for the train. The parameters m_j and C_D are estimated with the *Train Delay* predictor. The parameters r_j are estimated with the *Responsibility* predictor. More details about Eq. (1) can be found in the PIR.

VI. Experimental Evaluation

In this section we test the proposed *HM*, presented in Section V, against the actual RFI *EBM*, presented in Section IV-A, and *DDM*, presented in Section IV-B.

All the experiments have been conducted on a virtual machine in the Google Cloud Platform[2] (GCP). The machine is the *n1-standard-8* characterized by 8 core and 30GB of RAM and 500GB of SSD disk space. Each experiment has been repeated 30 times in order to ensure the statistical robustness of the results.

[2]Google Compute `https://cloud.google.com/products/`

A. Available Data

The experiments have been conducted exploiting the real data provided by RFI:

- data about train movements which contains the following information: Date, Train ID, Checkpoint ID, Actual Arrival Time, Arrival Delay, Actual Departure Time, Departure Delay and Event Type. The Event Type field can assume different values: Origin (O), Destination (D), Stop (F), Transit (T).
- data about the delay responsibilities: for every delay the percentage of RFI responsibility is available;
- timetables, including planning of exceptional train, cancellations, and *Gaining Time* of each section.

For the purpose of this work, RFI provided the access to the data of 12 months (the whole 2016 solar year) of train movements of one big Italian Region (Liguria). The data are relative to more than 2.500 trains and 146 checkpoints. The dataset contains 4.127.380 train passages.

From the PIR, freely available on the RFI website[1], we retrieved all the information needed to compute the *Penalty Costs* as described in Section V-B.4.

We also exploit, as exogenous information, the weather conditions from the weather stations in the area. For each checkpoint we consider the closest weather station to the railway station/line. We collect the data relative to the solar radiation and precipitations for the same time span of the train passages from Italian national weather service databases, which are publicly accessible for the Liguria Italian Region at [26]. From this data it is possible to extract both the actual and the forecasted weather conditions (Sunny, Rain, Heavy Rain, and Snow).

B. Key Performance Indicators

In the experiments, we exploit the following Key Performance Indicators (KPIs) for measuring the quality of the different models (in parenthesis we report the prediction

TABLE III
COMPARISON BETWEEN *HM* AND *EBM* FOR *Running Time* PREDICTION. (n) IS THE NUMBER OF TRAIN PASSAGES IN THE SECTION.

k	n	AASk *EBM*	AASk *HM*
1	7344	1.1	**0.9**
2	10672	1.7	**0.8**
3	22082	1.2	**0.9**
4	1013	1.4	**0.4**
5	25228	0.5	**0.4**
6	18090	0.8	**0.5**
7	398	3.2	**2.9**
8	12671	1.2	**0.6**
9	29357	1.4	**0.9**
10	5614	2.7	**1.5**
...			
AAS Regional		1.3	**0.8**
AAS High Speed		0.8	**0.6**
AAS Freight		1.9	**1.2**
AAS		1.3	**0.9**

TABLE IV
COMPARISON BETWEEN *HM* AND *EBM* FOR *Dwell Time* PREDICTION. (n) IS THE NUMBER OF TRAIN PASSAGES IN THE CHECKPOINT.

k	n	AACk *EBM*	AACk *HM*
1	49134	1.7	**0.7**
2	61888	**0.1**	0.3
3	22210	1.4	**1.2**
4	23629	2.4	**1.8**
5	29652	2	**1.6**
6	29271	1.3	**1**
7	33350	1.2	**0.9**
8	22508	0.5	**0.2**
9	33418	**0.8**	1
10	24307	**0.5**	0.9
...			
AAC Regional		**1.1**	**1.1**
AAC High Speed		**0.5**	0.7
AAC Freight		2.5	**1.5**
AAC		1.1	1

problem where they have been applied). These KPIs have been designed together with RFI based also on the lesson learned during the exploitation of the *DDM* [25]:

- AASk (*Running Time* prediction): the Average Accuracy for a particular Section k. AASk is computed as the averaged absolute value of the difference between the predicted and the actual *Running Times* in minutes;
- AAS (*Running Time* prediction): is the average over the different sections k of AASk
- AACk (*Dwell Time* prediction): the Average Accuracy for a particular Checkpoint k. AACk is computed as the averaged absolute value of the difference between the predicted and the actual *Dwell Times* in minutes;
- AAC (*Dwell Time* prediction): is the average over the different checkpoints k of AACk
- AAiCTk (*Train Delay* prediction): the Average Accuracy at the i$-th$ subsequent Checkpoint for Train k. For a particular Train k, the absolute value of the difference between the predicted delay and its actual *Train Delay* is averaged, at the i$-th$ subsequent checkpoint with respect to the actual checkpoint in minutes;
- AAiC (*Train Delay* prediction): is the average over the different trains j of AAiC.
- AAP (*Penalty Costs* prediction): is the Average Accuracy over the different trains between the predicted and actual *Penalty Costs* in Euros.

C. Results

In this section we compare the proposed *HM* for predicting *Running Times*, *Dwell Times*, *Train Delays*, and *Penalty Costs* against the *EBM* and *DDM*, by using the data described in Section VI-A and the KPIs described in Section VI-B

1) Running Time Prediction: In this first set of experiment we compare the *HM* with the *EBM* on the *Running Time* prediction problem. We could not compare them also with the *DDM* since it does not provide a solution for this problem [25].

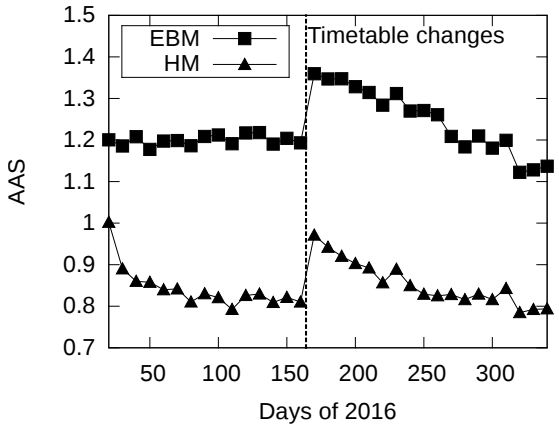

Fig. 5. AAS for the *Running Time* during the year.

Table III reports the AASk for a subset of the railway sections and the AAS also considering the different train types[3]. From the results it is possible to observe that:

- *HM* clearly outperforms the *EBM*;
- the improvement is more evident for Freight and Regional trains, instead for High Speed trains the two approaches provide similar results.

In order to show the ability of the proposed solutions to handle changes in the timetable, Figure 5 reports the value of AAS during the 2016. From Figure 5 it is possible to observe that:

- *HM* is constantly better with respect to the *EBM* during the whole year;
- *HM* needs really little time to learn a good model, for example in January after 10 days of data it reaches almost its optimal accuracy;
- *HM* and *EBM* exhibit an increase in the error in June (days from 180 to 210), this is motivated by a change in the timetable happened the 12nd of June.

Note that, even if the *HM* has a Data-Driven core, it is still robust and the *EBM* and much more dynamic of any *DDM*.

2) Dwell Time Prediction: For what concern the *Dwell Time* prediction problem, the approach, the results, and the comments are quite similar to the one made for the *Running Time* prediction problem. Table IV, analogously to Table III, reports the AACk for a subset of the checkpoints and the and AAC also considering the different train types[3]. From Table IV it is possible to observe that:

- in this problem *EBM* and *HM* provide similar results, being *HM* slightly better;
- similarly to the result for the *Running Time* prediction problem the *HM* approach results to be particularly effective for the Freight trains.

We do not report the equivalent of Figure 5 since results are basically the same.

[3]Because of confidentiality issues we cannot report the results and the ids for all the sections and all the checkpoints available.

TABLE V

COMPARISON BETWEEN *HM*, *EBM*, AND *DDM* FOR *Train Delay* PREDICTION. (n) MEANS THE NUMBER OF DAYS THAT THE TRAIN TRANSIT ACCORDING TO OUR DATASET. (–) MEANS NOT AVAILABLE SINCE DATA IS NOT ENOUGH TO BUILD THE MODEL.

AAiCTk		*EBM*	*DDM*	*HM*	*EBM*	*DDM*	*HM*	*EBM*	*DDM*	*HM*	*EBM*	*DDM*	*HM*
k \ i	n	1st			2st			3st			4st		
1	349	1	0.6	**0.5**	1.2	**0.7**	1	1.7	**1.1**	1.5	2.1	**1.4**	2.1
2	346	1	0.5	**0.4**	1.2	**0.8**	0.9	1.5	**1**	1.3	1.8	**1.2**	1.6
3	345	0.5	0.4	**0.3**	0.9	0.6	**0.4**	1.1	0.8	**0.6**	1.2	1	**0.7**
4	308	0.9	**0.5**	**0.5**	1.5	**1**	1.2	1.7	**1.3**	1.4	1.9	**1.4**	1.8
5	235	0.9	0.9	**0.7**	1.5	**1.3**	**1.3**	2	**1.5**	1.8	2.5	**1.8**	2.3
6	175	0.7	**0.4**	0.5	1.1	**0.7**	1	1.4	**0.7**	1.3	1.8	**1**	1.7
7	169	0.7	**0.4**	**0.4**	1	**0.6**	0.9	1.3	**0.6**	1.2	1.6	**0.8**	1.6
8	129	2.4	3.4	**1.6**	5.1	6.2	**3.9**	7.8	9	**6.4**	9.9	11.3	**8.2**
9	14	1.4	–	**1.1**	2.1	–	**1.7**	2.8	–	**2.2**	3.1	–	**2.6**
10	2	1.8	–	**1.1**	3.8	–	**1.9**	5.9	–	**3**	7.6	–	**4**
						...							
AAiC Regional		1.2	**0.8**	0.9	2.1	**1.5**	1.7	3	**2.2**	2.5	3.8	**2.8**	3.3
AAiC High Speed		0.7	0.7	**0.5**	1.2	1.1	**1**	1.6	**1.4**	**1.4**	2	**1.7**	1.8
AAiC Freight		1.9	3.5	**1.6**	3.6	5.2	**3.1**	5.3	6.9	**4.7**	6.9	8.2	**6.1**
AAiC		1	0.9	**0.8**	1.8	**1.5**	1.6	2.5	**1.8**	2.3	3.2	**2.1**	2.9

TABLE VI

COMPARISON BETWEEN *HM* AND *EBM* FOR *Penalty Costs* PREDICTION.

	EBM	*HM*
AAP Regional	4.15	**2.49**
AAP High Speed	0.2	**0.14**
AAP Freight	0.11	**0.1**
AAP	4.44	**2.71**

3) Train Delay Prediction: In this section we compare the *HM* with both the *EBM* and the *DDM* for the *Train Delay* prediction problem.

Table V reports the AAiCTk for a subset of the trains and subsequent checkpoints and the AAiC also considering the different train types[3]. From the results it is possible to observe that:

- both the *HM* and *DDM* perform better with respect to the *EBM* approach;
- the *HM* better predicts the delays in the subsequent checkpoint ($i = 1$);
- the *DDM* better predicts the delays when the distance from the actual checkpoint is larger;
- *DDM* is not able to perform the prediction for the trains for which we have too less information (i.e. infrequent trains) while *HM* is always able to provide an answer;
- for what concern the Freight trains, *DDM* provides the largest error while *HM* improves of $\approx 20\%$ over also the *EBM*.

4) Penalty Cost Prediction: In this section we compare the *HM* with the *EBM* on the *Penalty Costs* prediction problem. We could not compare them also with the *DDM* since it does not provide a solution for this problem [25].

Table VI reports the AAP considering the different train types[3]. From Table VI it is possible to observe that:

- the *HM* is much more effective with respect to the *EBM* for all the train categories;
- the difference is much more evident for the Regional trains which are also the most expensive in terms of *Penalty Costs* for RFI.

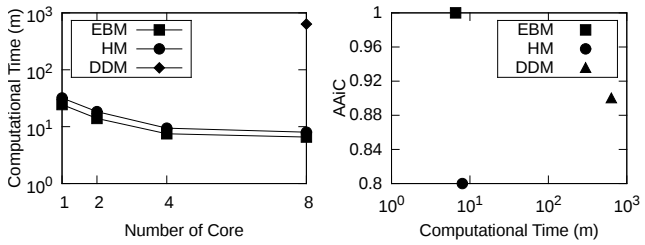

Fig. 6. Computational Time Evaluation.

D. Computational Requirements

Finally, we compare the computational requirements of the different models. Figure 6 depicts both the scalability varying the number of cores (left) and the trade-off between accuracy and computational requirements (right) for the *Train Delay* prediction case (AAiC with $i = 1$). The time reported on the axis is the time needed for performing the analysis of all the 12 months of data provided by RFI.

From Figure 6 we can observe that:

- *EBM* and *HM* have a similar scalability, the computational time decreases smoothly when more cores are added to the computation;
- *DDM*, when 8 cores are exploited, requires $100\times$ the time with respect to *EBM* and *HM* (we did not execute *DDM* with less than 8 cores because the computation required more than 10 hours);
- *EBM* and *HM* have similar computational requirements, being *HM* just slightly slower with respect to *EBM*;
- *HM* provides clearly the best trade-off between accuracy and computational requirements.

In conclusion, *EBM* is the fastest method but, with a small additional computational effort with respect to *EBM*, *HM* is able to deliver a model which is extremely more accurate with respect to *EBM* and *DDM*.

VII. CONCLUSIONS

In this work we dealt with the problem of understanding and predicting the train movements in Large-Scale Railway

Networks. In particular, our purpose was to predict the *Running Time* of a train between two stations, the Dwell Time of a train in a station, the *Train Delay*, and the *Penalty Costs*, four important aspects which fully characterize the train movements and that where never studied together before. For this purpose, we proposed, for the first time, an hybrid approach which is able to merge together two approaches adopted in literature: the one which develops models based on the knowledge of the network and the experience of the operators and the one based on the analysis of the historical data about the network with advanced analytic methods. The result is a dynamic, interpretable, and robust hybrid data analytics system able to handle non recurrent events, changes in the behaviour of the network, and ability to consider complex and exogenous information like weather information. Basically, the proposed approach is able to take the strengths of the two original approaches and to limit their weaknesses. Results on real world data coming from the Italian railway network shown that the proposed solution outperform both state-of-the-art experience and Data-Driven based systems.

ACKNOWLEDGMENTS

This research has been supported by the European Union through the projects IN2DREAMS (European Union's Horizon 2020 research and innovation programme under grant agreement 777596) and In2Rail (European Union's Horizon 2020 research and innovation programme under grant agreement 635900).

REFERENCES

[1] F. Restel, "The markov reliability and safety model of the railway transportation system," in *Safety and Reliability: Methodology and Applications-Proceedings of the European Safety and Reliability Conference*, 2014.

[2] T. Nowakowski, "Analysis of modern trends of logistics technology development," *Archives of Civil and Mechanical Engineering*, vol. 11, no. 3, pp. 699–706, 2011.

[3] F. Ghofrani, Q. He, R. M. Goverde, and X. Liu, "Recent applications of big data analytics in railway transportation systems: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 226–246, 2018.

[4] W. Daamen, R. M. P. Goverde, and I. A. Hansen, "Non-discriminatory automatic registration of knock-on train delays," *Networks and Spatial Economics*, vol. 9, no. 1, pp. 47–61, 2009.

[5] P. Kecman and R. M. P. Goverde, "Process mining of train describer event data and automatic conflict identification," *Computers in Railways XIII: Computer System Design and Operation in the Railway and Other Transit Systems*, vol. 127, p. 227, 2013.

[6] R. M. P. Goverde and L. Meng, "Advanced monitoring and management information of railway operations," *Journal of Rail Transport Planning & Management*, vol. 1, no. 2, pp. 69–79, 2011.

[7] T. Albrecht, "Reducing power peaks and energy consumption in rail transit systems by simultaneous train running time control," *WIT Transactions on State-of-the-art in Science and Engineering*, vol. 39, 2010.

[10] P. Kecman and R. M. P. Goverde, "Online data-driven adaptive prediction of train event times," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 465–474, 2015.

[8] H. Ko, T. Koseki, and M. Miyatake, "Application of dynamic programming to the optimization of the running profile of a train," *WIT Transactions on The Built Environment*, vol. 74, 2004.

[9] P. Lukaszewicz, "Energy consumption and running time for trains," Ph.D. dissertation, Doctoral Thesis). Railway Technology, Department of Vehicle Engineering, Royal Institute of Technology, Stockholm, 2001.

[11] R. Wang and D. B. Work, "Data driven approaches for passenger train delay estimation," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, 2015, pp. 535–540.

[12] 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.

[13] 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.

[14] 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 Winter Simulation Conference*, 2012.

[15] I. A. Hansen, R. M. P. 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.

[16] A. Berger, A. Gebhardt, M. Müller-Hannemann, and M. Ostrowski, "Stochastic delay prediction in large train networks," in *OASIcs-OpenAccess Series in Informatics*, vol. 20, 2011.

[17] W. Fang, S. Yang, and X. Yao, "A survey on problem models and solution approaches to rescheduling in railway networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2997–3016, 2015.

[18] A. D'Ariano, T. Albrecht, J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, and S. Sone, "Running time re-optimization during real-time timetable perturbations," *Timetable Planning and Information Quality*, vol. 1, pp. 147–156, 2010.

[19] H. Flier, R. Gelashvili, T. Graffagnino, and M. Nunkesser, "Mining railway delay dependencies in large-scale real-world delay data," in *Robust and online large-scale optimization*, 2009, pp. 354–368.

[20] T. H. Tsai, C. K. Lee, and C. H. Wei, "Neural network based temporal feature models for short-term railway passenger demand forecasting," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3728–3736, 2009.

[21] F. P. G. Marquez, R. W. Lewis, A. M. Tobias, and C. Roberts, "Life cycle costs for railway condition monitoring," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 6, pp. 1175–1187, 2008.

[22] O. Brünger and E. Dahlhaus, "Railway timetable & traffic-analysis, modelling," in *Simulation, Eurail press*, 2008.

[23] A. D'Ariano, M. Pranzo, and I. A. Hansen, "Conflict resolution and train speed coordination for solving real-time timetable perturbations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 208–222, 2007.

[24] R. M. P. Goverde, "A delay propagation algorithm for large-scale railway traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 3, pp. 269–287, 2010.

[25] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra, N. Mazzino, and D. Anguita, "Advanced analytics for train delay prediction systems by including exogenous weather data," in *IEEE International Conference on Data Science and Advanced Analytics*, 2016.

[26] Regione Liguria, "Weather Data of Regione Liguria," http://www.cartografiarl.regione.liguria.it/SiraQualMeteo/script/PubAccessoDatiMeteo.asp, 2018.

[27] Regione Lombardia, "Weather Data of Regione Lombardia," http://www.arpalombardia.it/siti/arpalombardia/meteo/richiesta-dati-misurati/Pagine/RichiestaDatiMisurati.aspx, 2018.

[28] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.