IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 July 2024, accepted 8 September 2024, date of publication 18 September 2024, date of current version 10 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3463022

RESEARCH ARTICLE

XGBoost in Public Transportation for Multi-Attribute Data: Delay Prediction in Railway Systems in Real-Time

SONDOSS CHTIOUI[1,2,3], SEBTI MOUELHI[1,2], SÉBASTIEN SAUDRAIS[1,2], TOUFIK AZIB[1,2], (Member, IEEE), MARC ILLE[3], MELANIE MOREL[3], AND FREDERIC ORU[3]

[1]ESTACA'Lab, ESTACA, Laval Campus, 53000 Laval, France
[2]ESTACA'Lab, ESTACA, Paris-Saclay Campus, 78066 Saint-Quentin-en-Yvelines, France
[3]EGIS Rail, 69006 Lyon, France

Corresponding author: Sondoss Chtioui (sondoss.Chtioui@estaca.fr)

This work was supported in part by Egis Rail, Lyon, France.

ABSTRACT Predicting delays in metro and tram services is a complex task that requires advanced approaches, such as powerful machine learning tools. This study addresses this topic by applying the XGBoost and Bayesian Optimization (BO) algorithms, which offer a prediction horizon of 15 minutes instead of the next-station prediction, which leaves a very short time for the operator to react if we want to use the prediction for the next station. Our research strongly emphasizes methodological validation, with daily evaluations against real-time data. This process is reinforced by collaboration with the Operational Control Center (OCC) to ensure robustness. The 15-minute delay strikes a balance, giving control center operators sufficient notice to orchestrate traffic management, mitigate disruption, and take timely action. With an exemplary real-world accuracy of 95%, the results of our model have been validated by the OCC. Future efforts will include the seamless integration of predictive capabilities into real-time display systems for the OCC, providing innovative information to optimize traffic flows and ensure punctuality in urban rail systems.

INDEX TERMS Urban rail systems, train departure delay prediction, extreme gradient boosting machine, Bayesian optimization, operational control center, planned timetable, real timetable, predictive algorithms to assist the OCC in preventing metro and tramway delays up to 15 minutes in advance.

I. INTRODUCTION

Rail operations, particularly those linked to trams and metros, perceive the rail network as a shared resource set. They must manage access to these resources safely while complying with the throughput and service quality objectives expected by the rail system. This means ensuring traffic safety, protecting train access to the tracks, and preserving the passenger experience [1]. On each railway line, operations are characterized by very high traffic flow, resulting in short intervals between trains. The operations department, managed by the OCC system, is responsible for implementing the transport plan by optimizing delays for each train throughout the day, while managing equipment, drivers, lines, and ensuring passenger information.

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang.

Delays pose a significant challenge for OCC, particularly during peak hours. This problem tends to get worse over time, creating a domino effect in which one delay can lead to another. Although procedures have been implemented to reduce delays by OCC, the main challenge lies in the time constraints to regulate the line. This time limitation has repercussions on preceding and following trains, as well as on the overall flow of traffic. As a result, delays frequently exceed 5 minutes, causing passengers to crowd together on platforms. The frequency of this traffic depends on the time elapsed between the initial incident causing the delay and the regulatory intervention of the operators.

Several state-of-the-art approaches employ machine learning, especially deep learning models, to predict railway delays. These models often utilize historical data, weather conditions, and other relevant features to make accurate predictions [2]. However, the real-time applicability of these

models remains a concern. The predictions generated are typically instantaneous, leaving railway operators with minimal time to react and implement preventive measures [3]. In the realm of predicting delays in public transportation systems, various approaches have been explored to effectively anticipate potential incidents [4]. Among previous works, machine learning models have emerged as a promising strategy to address this complex issue. The existing methodologies in railway delay prediction, although powerful, fall short of addressing the time-sensitive nature of operational decision-making. The inability to provide timely predictions means that operators are often unable to take proactive measures to mitigate the impact of delays [5]. The gap between prediction generation and implementation hinders the practical utility of these advanced models in real-world railway operations. In contrast to the current landscape, our research addresses the limitations of existing delay prediction models.

In the dynamic rail transportation landscape, real-time delay prediction plays a central role in optimizing operations and improving overall efficiency. The use of artificial intelligence techniques for predicting delays throughout the daily running of trains helps solve the problem of time management to regulate traffic. This provides operators with more time to anticipate delays and take action by reducing the likelihood of the impact on other trains. Railway systems are intricate networks where the synchronization of numerous factors is essential to maintain a seamless flow of operations [6]. Delays, whether caused by unforeseen events, passenger behavior, or scheduled events, can have cascading effects on the entire network. The utilization of advanced algorithms such as XGBoost [3] offers a promising avenue for predicting delays with a level of precision that can significantly impact decision-making and resource allocation. XGBoost, well-known for its efficiency in handling structured data and feature importance, is utilized independently in this context. This powerful algorithm is chosen for its ability to consider diverse attributes influencing delays by creating a robust model capable of capturing the temporal patterns inherent in train movements throughout the day, focusing solely on the capabilities of this algorithm.

We propose a multifaceted approach inspired by [3], using XGBoost and BO. Our method differs in two important ways: first, the type of input used, and second, the prediction horizon. While in [3] predictions are made for the next station, we propose to predict within a 15-minute window into the future, covering at least the next five stations. The aim is to unravel the complexities of real-time delay prediction in rail systems and to highlight the potential of XGBoost to transform the use of operational data through tailored preprocessing. By venturing into real-time rail operations with this 15-minute forecast window, we are enabling operators to react quickly and use predictions effectively. Predicting and managing delays is becoming central to ensuring the smooth and efficient operation of transport services. This research explores the intricacies of delay forecasting and addresses the challenges of implementing these forecasts in real-world scenarios, with a particular focus on harnessing the power of XGBoost.

The inputs to the model described in our methodology are based on the LSTM approach [2], which incorporates historical data for each output. In our case, this includes the journey history for each train. The introduction of a novel approach, where forecasts are issued 15 minutes before the expected delay time, ensures that operators receive timely predictions. This advance notice allows them to proactively implement corrective measures, minimizing disruption to the timetable and improving the passenger experience. The primary objective of this study is to provide delay information 15 minutes before a train arrives at a station, using the predictive power of XGBoost.The immediate relevance of these forecasts is critical for operators, giving them a critical time advantage in managing delays and mitigating their impact on timetables and operational efficiency. This proactive approach provides rail operators with a valuable opportunity to respond effectively to potential delays, thereby optimizing service.

This paper is structured as follows: In the ‘‘Related work’’ section, we review previous studies on rail delay prediction. The ‘‘Methodology’’ section describes the delay problems and identifies the factors that influence train departure delays with our approach to the influence of past delays on future train journeys using the three rail datasets presented in this section.’’Use case’’ presents the real data set extracted from a railway control center for training our model in the ‘‘XGBOOST’’ section, which presents the training of our approach to the influence of past delays on future train paths, using BO to optimize the XGBOST parameters. In addition, we provide performance metrics from our experimental study on real data, including results shared with the OCC. Finally, we summarise our results, draw conclusions, and suggest possible extensions of this research.

II. RELATED WORK

The field of railway delay prediction has witnessed significant advancements in recent years, with a particular emphasis on leveraging machine learning algorithms, especially deep learning models, for precise forecasting. Several studies have explored the application of these sophisticated techniques to predict delays in real-time scenarios. However, a common limitation observed in existing works is the instantaneous nature of predictions, presenting practical challenges for railway operators.

Predicting delays in metro or tram systems is generally a significant challenge due to the complexity of collecting data and availability from various sources, encompassing multiple formats, including static and temporal data. The task becomes intricate as it involves understanding and integrating heterogeneous information to develop robust prediction models. The diversity in data formats, coupled with the dynamic nature of traffic in public transportation systems, makes delay modeling an exciting yet complex endeavor. This complexity necessitates innovative approaches and advanced algorithms

to fully leverage the potential of available data and provide accurate real-time predictions.

Railway data learning and modeling are constantly evolving fields that have practical applications in rail traffic management, predictive infrastructure maintenance, and passenger safety [6]. Numerous studies have been conducted in the literature to improve railway systems performance and efficiency by exploiting data generated by various equipment such as sensors, and signaling systems [2], [8]. Machine learning techniques, especially neural networks, are widely used to model and predict railway outputs (delays). Linear models have been mostly superseded by complex models [2], [8], [9], including deep neural networks to predict train arrival delay at the next station using Extreme Learning Machine (ELM) with nine characteristics plus the Particle Swarm Optimization (PSO) algorithm to optimize the hyper-parameters of ELM [5], that have greater accuracy and performance, and have the ability to extract valuable insights from unprocessed and unstructured data using gradient boosting (XGBoost) prediction model that captures the relation between the train arrival delays and various railway system characteristics [10]. The latest technological advances allow large volumes of data to be processed and analyzed in real time, thus allowing operators to make knowledgeable rapid decisions. To summarize, the current state of rail data learning and modeling involves the increased utilization of machine learning techniques, and the integration of multimodal techniques using three different methods to define inputs including normalized real number, binary coding, and binary set encoding inputs [11].

In [12], the authors successfully demonstrated the use of LSTM for predicting train delays, taking into account specific temporal patterns in train movements. These works highlighted LSTM's capability to handle long-term time sequences, making them promising candidates for modeling delays in dynamic environments like public transportation networks [13]. Contrasting with our XGBoost-based approach, our proposal also incorporates the past history of each train using the LSTM principle, resulting in a lighter and faster model compared to LSTM. On the other hand, boosting-based approaches, including XGBoost, have also been explored successfully in train arrival delay prediction. The study [3] proposes a data-driven method that combines Extreme Gradient Boosting (XGBoost) and a Bayesian Optimization (BO) algorithm to predict train arrival delays at the next station by handling complex and heterogeneous datasets, making it a relevant choice for modeling delays in constantly evolving transport systems. While LSTM is a type of Recurrent Neural Network (RNN) [7], other members of the RNN family have also been explored for delay prediction. The applied model accurately predicts flight delays, addressing challenges posed by massive data and dependencies. In comparison to traditional methods, the Deep Learning (DL) model demonstrates superior precision, accuracy, sensitivity, recall, and F-measure. Evaluation of imbalanced and balanced datasets highlights the model's effectiveness, surpassing both traditional methods and a previous RNN model in forecasting flight delays. The innovative approach presented here represents a promising advancement in enhancing accuracy and reliability in flight delay prediction.

The authors in [5] introduce the integration of ELM PSO into the railway delay prediction landscape representing a novel and promising direction, offering both accuracy and practicality concerning urban railway operations. This methodology, as well as focusing on time-advantaged predictions, distinguishes our approach from conventional models and contributes to the ongoing development of effective delay prediction strategies in the transport sector.

In addition, various other models have been explored in the literature for the prediction of delays in transportation systems [7], [14]. The diversity of these approaches reflects the ongoing effort to find the most effective methods for capturing the complexities inherent in real-time transportation data.SVM (Support Vector Machine) is a machine learning algorithm [15], that has shown promise in modeling and predicting delays. Studies such as [16] have explored the use of SVM for classification and regression tasks related to transportation delays. SVM's ability to handle non-linear relationships and high-dimensional data makes it a noteworthy candidate for delay prediction, but its performance may depend on the tuning of hyperparameters and the nature of the data.

The method we propose in the following section differs from the work described by the authors on a crucial point. All the predictions described cover the next station, whereas we propose predictions for a horizon of 15 minutes, corresponding to at least the fifth station after the current station. This approach gives the OCC the necessary time to react and use these forecasts in the field. The choice of a 15-minute horizon can be adapted to the needs of each OCC. For example, it is possible to extend the forecast horizon to 20 minutes, although this could reduce the accuracy of the model. Conversely, the horizon can be reduced to 10 or 5 minutes for greater accuracy, but this may not give operators enough time to react. Egis-rail chose a 15-minute horizon based on the expertise of the PCC.

III. METHODOLOGY

Under normal circumstances, the OCC is responsible for managing the traffic of each train-station pair throughout the day. Their main objective is to achieve the targets set by minimizing delays for each train-station pair. They also try to optimize the flow of passengers on the platforms. Delays in particular are an increasing challenge over time. A significant correlation has been observed between the increase in delays and the number of passengers on platforms. The aim is to help OCC anticipate delays through a real-time forecasting program. The obstacle they face is the lack of time to anticipate delays immediately. The problem is that delays increase progressively over time, disrupting the whole line.

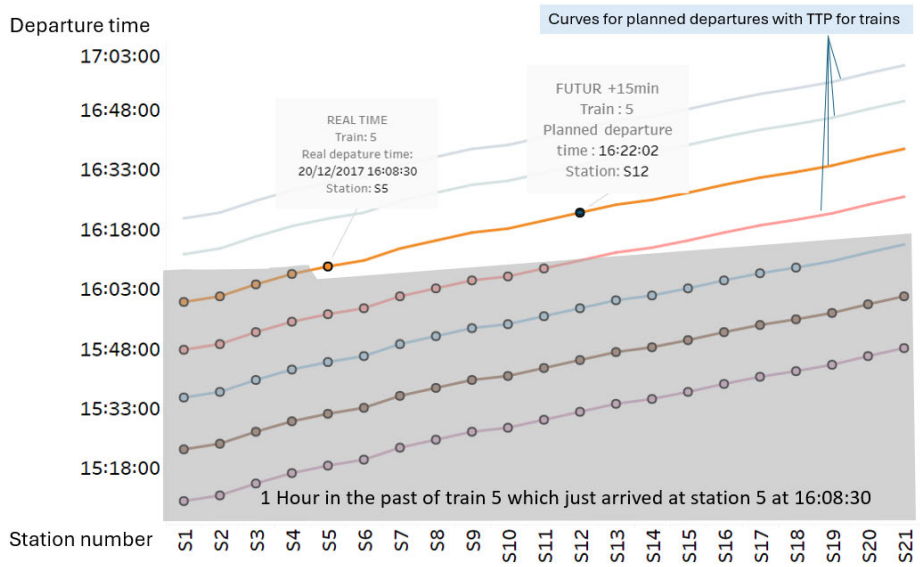

FIGURE 1. The fundamental concept of a 15-minute prediction window: consider t_5, departing from s_5 in real-time, and estimate its arrival within the next 15 minutes at s_{12}.

Our specific objective is to develop a method of delay prediction that would allow OCC to have information on the expected delays for each train-station pair 15 minutes in advance. This would make it possible to effectively anticipate, according to predefined procedures, the addition, deletion, delay, or advance of trains. The ultimate goal is to ensure constant intervals between two trains at a given station. The distinguishing feature of our approach is its ability to predict delays over a 15-minute horizon, giving the OCC a sufficient window of opportunity to implement proactive measures.

Figure 1 shows the basic concept of a 15-minute prediction window for a given train, denoted t_i, where i belongs to the set of trains running during the day. The idea is to use the history of the trains that precede and follow t_i, as well as the times at which the train passes through the preceding stations (shown in grey). The aim is to study the past behavior of trains and its potential impact on the future behavior of train t_i over the next 15 minutes. By adopting this approach, the model aims to provide the railway operations manager with predictive information 15 minutes before the arrival of train t_i at the next station s_{j+e}, where e is an estimate of the next station in the future of t_i for the next 15 minutes. By integrating historical data and relying on complex time patterns, this approach aims to improve operational visibility. This allows operators to obtain critical information in advance, facilitating more efficient traffic management. Let's look at the present moment, marked by the departure of t_5 from station S_5 at 04:07 PM. We examine the actual departure times of the trains that preceded and followed t_5 over the past hour. By comparing the actual time t=04:07 PM with the planned time (PTT) of t_5, we estimate that the arrival time of t_5 at the station in the next 15 minutes is t+15=04:22 PM, which means that our t_5 should arrive at station S_{12} within 15 minutes. Using an XGBOOST prediction model we predict whether t_5 will be on time, late, or early at S_{12} in the next 15 minutes, based on the estimated arrival time. This comprehensive process involves analyzing historical departure patterns and future forecasts to make real-time predictions for the specific train in question.

Accurately predicting train delays 15 minutes before they occur requires a thorough understanding of traffic patterns, train movement dynamics, and environmental factors affecting the system. Figure 2 presents our methodology for building a 15-minute prediction model using a dataset from an OCC in a French city. This methodology required meticulous preparation phases to prepare the data for our learning model(4). This rigorous preparation ensures that the data fed into our models is accurate, reliable, and suitable for training. Our data preparation phases are outlined in Figure 2. The first phase of pre-processing, Data Cleaning (1), involved resolving issues in the raw data, such as removing duplicate lines and correcting outliers. Additionally, string data, such as station names, were encoded to be usable for model training. Once this initial cleaning phase was validated, the second phase, Data Enrichment (2), began. This phase enhanced the dataset with inter-station data by calculating the standard deviation and average time between consecutive stations. Key information, such as predicted delays, actual intervals between trains, and predicted travel times between stations, was also calculated. These enriched features aimed to improve the model's understanding of the temporal and spatial dynamics within the train network. Additionally, a scraping method was employed to extract weather data for 2017, providing crucial information for effective model training. Data engineering (3) details how we utilized the data in our learning model (4), with three data sets explained in the following sections.

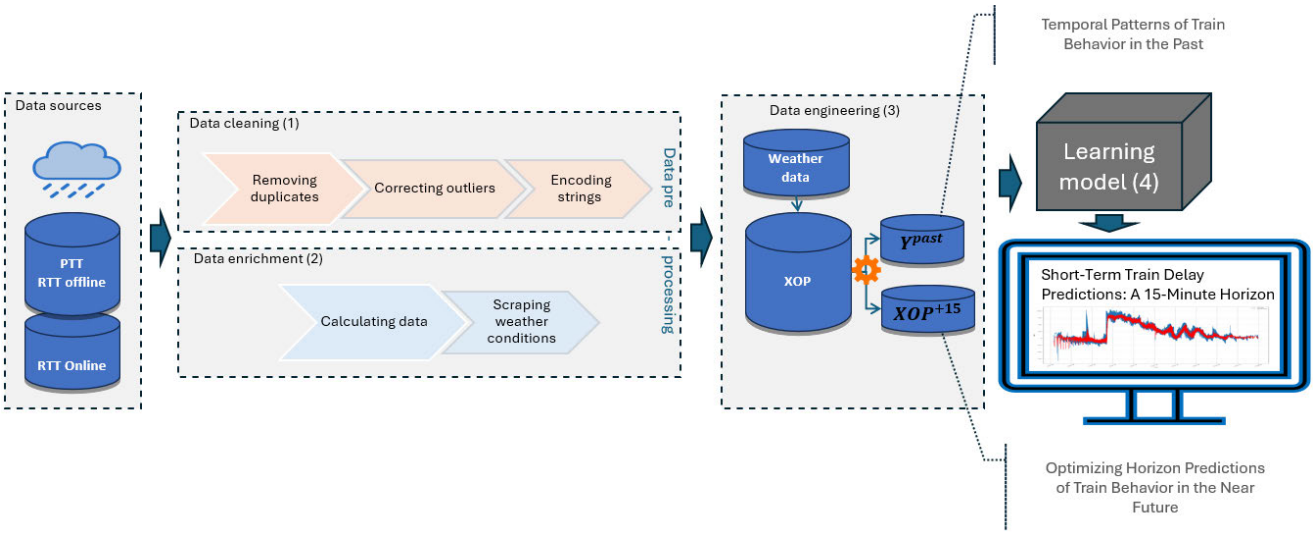

FIGURE 2. Data processing methodology, sources and engineering.

The choice of XGBoost is explained by its remarkable ability to process heterogeneous data and capture complex temporal dependencies [17]. Our approach leverages the power of XGBoost to provide a comprehensive solution, perfectly tailored to the specific operational needs of the OCC. XGBoost stands out for its excellence in processing various types of data based on feature selection and error correction [18], offering essential flexibility to manipulate the variety of railway information [3]. Additionally, XGBoost excels at optimizing model performance. Its ability to compile quickly compared to other models, such as neural networks like LSTM, represents a significant advantage. This efficiency in the compilation process allows for faster implementation and increased responsiveness, thus meeting the operational requirements of the OCC. A notable feature of XGBoost is its ability to retain context over long periods, making it particularly suited to capturing the complex temporal patterns inherent in streetcar movement data [19]. By integrating these benefits, our approach aims to improve the quality of forecasts and strengthen the OCC's ability to make informed decisions in real time for more efficient rail traffic management.

Through this meticulous data pre-processing and the utilization of advanced machine learning techniques, our research seeks to empower the OCC with a predictive tool capable of not only anticipating delays with a 15-minute lead time but also ensuring that the predictions are based on a refined and high-quality dataset. This combination of robust data pre-processing and advanced modeling techniques is integral to the success of our approach in enhancing the operational efficiency and reliability of train services managed by the OCC.

A. INPUT DATA: OPERATION DATA XOP

After completing the data engineering (3) outlined in Figure 2, we have aggregated all pertinent *XOP* data to initiate the analysis and model testing for delay prediction. These carefully selected variables serve as predictors in our delay prediction model, each making a unique contribution to the modeling of delays. This selection enables a comprehensive understanding of the studied system's various temporal and contextual aspects. To ensure clarity regarding their respective scales, we have specified the associated units for each variable.

Let $x_{i,j} \in XOP$ where $XOP = (S, R, PTT, RTT, AI, A, D, PI, P, AT, Y)$, the overall set of data grouping different categories of operation data, each identified by a specific letter. Using this notation, each element of the set *XOP* is associated with a specific category of information related to trains, stations, planning, and departure times as shown in Table 1, Where:

- t_i: Train number throughout each day, $T = t_1, t_2, \ldots, t_{40}$.
- s_j: Station number, $S = s_1, s_2, \ldots, s_{21}$.
- pt_{t_i,s_j}: Planned departure time of t_i at the current station s_j ($pt_{t_i,s_j} \in PTT$).
- rt_{t_i,s_j}: Real departure time of t_i at the current station s_j ($rt_{t_i,s_j} \in RTT$).
- ai_{s_j}: Average time of the planned timetable between the current s_j and the previous station s_{j-1} (Seconds).
- a_{s_j}: Average time spent at the current station s_j (Seconds).
- d: Day of the week.
- pi_{t_i,s_j}: Planned time interval between two consecutive trains t_i and t_{i+1} (seconds).

$$pi_{t_i,s_j} \in PI \,|\, pi_{t_i,s_j} = pt_{t_i,s_j} - pt_{t_{i+1},s_j} \tag{1}$$

- p_{t_i,s_j}: Planned travel time for a t_i between s_j and s_{j+1}.

$$p_{t_i,s_j} \in P \,|\, p_{t_i,s_j} = pt_{t_i,s_{j+1}} - pt_{t_i,s_j} \tag{2}$$

- at: Average temperature for a day using *WebScraping* methods.

- y_{t_i,s_j}: The actual departure delay time of the train t_i at the current station s_j indicates the difference between the real departure time and planned departure time (seconds).

$$y_{t_i,s_j} \in Y \mid y_{t_i,s_j} = rt_{t_i,s_j} - pt_{t_i,s_j} \tag{3}$$

B. INPUT DATA : PREVIOUS TRAIN DELAYS Y^{PAST}

The principle of previous delays relies on collecting and utilizing historical delay data from XOP as illustrated in Figure 2 phase (3), regarding a train to anticipate its future delays. The central idea is to create a specific dataset for each train t_i referred to as previous delays of the train (Y^{past}), which consolidates the historical delays of train t_i across all stations, considering the preceding trains. The Figure 2 illustrates this concept, depicting the comprehensive dataset, denoted as $y^{past}_{t_i,s_j,rt_{t_i,s_j}}, t_i \in T, s_j \in S$ and $rt_{t_i,s_j} \in RTT$, capturing the historical delays of train t_i in all stations as a list. This data set is formed by the union of the delays of all previous trains of t_i on the entire network of stations, as presented in Table 2, where y^{past}_1 represents the first delay of a previous train in a station and y^{past}_h represents the last delay of a previous train before the current time. In the context of the data set $XOP = (S, R, PTT, RTT, AI, A, D, PI, P, AT, Y)$, it is noted that there exists a subset Y^{past} representing all previous train delays for a train t_i at a station s_j with a real departure at rt_{t_i,s_j} This subset is such that the previous delays fall within a one-hour window in the past, as shown in the grey window in Figure 2. For each train t_i, where y_{t_i,s_j} represents the delay of this train t_i at station s_j, we define the set of previous delays, denoted as $y^{past}_{t_i,s_j,rt_{t_i,s_j}}$, as:

$$\begin{aligned} y^{\text{past}}_{t_i,s_j,rt_{t_i,s_j}} = \{&y_{t_k,s_l} \mid t_k \in T, s_l \in S, \\ &rt_{(t_k,s_l)} < rt_{(t_i,s_j)} \text{ and } rt_{(t_k,s_l)} > rt_{(t_i,s_j)} - 3600\} \end{aligned} \tag{4}$$

where:

- $Y^{past}_{t_i,s_j,rt_{t_i,s_j}}$ is the subset of previous train delays for the train at the station at the real departure time rt_{t_i,s_j}.
- y_{t_k,s_l} is the delay of train t_k at station s_l.
- rt_{t_k,s_l} is the real departure time of train t_k at station s_l.
- The condition $rt_{t_k,s_l} < rt_{t_i,s_j}$ ensures that we only consider delays of trains that have already left the stations before the real departure time rd_{t_i,s_j}.
- The condition $rt_{t_k,s_l} > rt_{t_i,s_j} - 3600$ ensures that we only consider delays of trains that have left the stations within the one-hour window before the real departure time rt_{t_i,s_j}.

This approach allows the construction of a specific dataset for each train, encompassing historical delays across all stations while considering the delays of trains preceding the train in question. By examining the cumulative delay history of the train, considering station-by-station variations and interactions with preceding trains on the same track, the predictive model can capture specific temporal trends and relationships for that particular train. Thus, this methodology highlights the importance, in our particular context, of applying a unified approach to temporal data using simple machine learning models such as XGBoost, rather than opting for complex alternatives such as LSTM. The primary goal is to simultaneously optimize computation time and memory usage. The integration of XGBoost as a machine learning model aims to maximize computational efficiency and resource management. Simple models such as XGBoost are characterized by their speed in both training and deployment, resulting in significant time savings. This approach underlines the importance of preferring simpler and more efficient methods for processing temporal data while retaining the flexibility to opt for more complex models when the complexity of the temporal relationships justifies it.

C. INPUT DATA: FUTURE OPERATING STATUS 15 MINUTES AHEAD XOP^{+15}

The data set $XOP = (S, R, PTT, RTT, AI, A, D, PI, P, AT, Y)$, there exists a subset XOP^{+15} designed to encompass future information about train t_i at a station within a 15-minute time window as shown in Figure 1,2 phase (3). The detailed process is as follows:

- Current operating data of train t_i at station s_j is considered, including information such as the real departure time rt_{t_i,s_j}.
- An estimation is made for the arrival of train t_i at station $s_{(j+e)}$ within a 15-minute window in the future.
- The anticipated position of train t_i in the next 15 minutes is determined based on the PTT as shown in Figure 2.
- The subset XOP^{+15} is formed, encompassing both current and future data for train t_i. It integrates information related to the real departure, the estimated future arrival, and the predicted position within the next 15 minutes as shown in Figure 2 for train 5 in station 12.

Let $x^{+15}_{(t_i,j_{j+e},rt_{t_i,s_j})} \in XOP^{+15}$ such that $x^{+15}_{(t_i,j_{j+e},rt_{t_i,s_j})} = \{t_i, s_j, rt_{t_i,s_j}, s_{(j+e)}, pt_{(t_i,s_{j+e})}\}$, where:

- $s_{(j+e)}$ represents the estimated station for a train t_i within the next 15 minutes.
- $pt_{(t_i,s_{j+e})}$ denotes the planned departure time of train t_i at station $s_{(j+e)}$.

In summary, this subset XOP^{+15} is used to represent the future data of the train and is used in conjunction with the sets XOP and Y^{past} which represent the operational variables (timetables) of the trains and the set of past delays of a train in a station respectively, note that the global data set for our training model is $X = XOP \cap Y^{\text{past}} \cap XOP^{+15}$, as shown in Table 1. The problem of predicting delays Y is formulated under the mathematical representation:

$$\hat{y}_{(t_i,s_j)} = f(x_{(t_i,s_j,rt_{(t_i,s_j)})}), \quad x_{(t_i,s_j,rt_{(t_i,s_j)})} \in X, \quad \hat{y}_{(t_i,s_j)} \in \hat{Y}^{+15} \tag{5}$$

where X is the global dataset resulting from the fusion of current features XOP and previous delays Y^{past} and the future elements XOP^{+15}. This formulation expresses the functional

TABLE 1. Feature exploration for delay prediction y^{+15}.

Inputs X																Output
XOP											XOP^{+15}		Y^{past}			Delays
t	s	rt	ai	a	d	pi	p	at	y		pt^{+15}	s^{+15}	y_1^{past}	...	y_h^{past}	y^{+15}
t_{14}	s_{10}	2017-09-01 07:33:00	135	34	5	300	120	22	50		07:47:00	s_{17}	-185	...	68	147
t_6	s_{18}	2017-09-01 07:33:49	96	23	5	360	96	22	110		07:48:30	s_3	209	...	-28	96
..	..	..	..	..	..	..	..	..	..		..	..	...	..	..	..
t_m	s_{15}	2017-12-01 23:47:20	130	31	7	120	120	9	-170		00:10:00	s_{21}	-591	...	141	-524

dependence between the combined features and the target variable. The approach, XGBoost, is employed as a prediction function f to model this complex relationship.

IV. USE CASE

Due to the sensitivity and confidentiality requirements of the data provided by our industrial partner, specific details of lines, timetables, and other operational information cannot be disclosed. This is a precautionary measure to protect confidential information relating to the operation of the transport network concerned. We are fully committed to ensuring the security and confidentiality of the data while allowing in-depth analysis of operational performance based on the general information available in our dataset.

The utilization of real-world data in the process of data modeling is of paramount importance as it ensures the accuracy and dependability of the outcomes. In this context, the actual operational data employed in the study was sourced from a collaborative effort with a railway control center responsible for managing 21 stations and 40 trains operating during the day. This raw dataset encompasses critical information such as PTT (Planned Time Table), RTT (Real Time Table), train numbers, station names, and platform details [20].

Our database, spanning three months from September 1, 2017, to December 31, 2017, provides a comprehensive snapshot of train operations, encompassing railway lines. Specifically focusing on 21 stations, with two distinct tracks labeled 0 from S_1 to S_{21} and 1 from S_{21} to S_1, this dataset was selected to capture a representative and diverse range of scenarios and operating conditions, thereby enhancing the robustness of our study.

The utilization of real-world data, despite the necessary confidentiality constraints, significantly contributes to the validity of the data modeling process. This authenticity ensures that the models developed are more adaptable to real-world situations, fostering a better understanding of the phenomena under investigation. Moreover, the use of genuine data results in more relevant recommendations for policymakers and practitioners, as the models are grounded in the intricacies of actual operational scenarios. The train line connects a bustling metropolis to a distant suburb, passing through 21 stations. Each day, multiple trains travel between these two points, carrying passengers for their daily commutes as shown in Figure 2.

In the departure planning phase of PTT, each day begins with the scheduled departure times for each train. Throughout the day, the trains move according to their PTT, stopping at different stations. Each train remains on schedule PTT until it reaches its final destination at station 21. On arrival at each station, the system evaluates several factors, including the scheduled departure time and the actual departure time recorded in the RTT database, any difference between these times gives rise to delay assessments, calculated based on the difference between the actual departure time and the scheduled departure time. The railway day ends when the last train has completed its journey and arrived at the last station 21. This scenario highlights the intricacies of the daily management of a train line. The inclusion of real and planned departure time for each train allows for an assessment of punctuality and the handling of potential delays.

V. XGBOOST

This integrated methodology aims to improve train delay prediction using a comprehensive approach that exploits the specific benefits of XGBoost while incorporating Bayesian Optimisation. To achieve this, we have taken a careful approach to data pre-processing, including cleaning, normalization, and variable encoding. An important innovation in our methodology is the enrichment of the model by incorporating the delays of previous trains. To do this, we used a sliding time window to capture the temporal relationships between past, present, and future. This allowed us to incorporate the temporal dynamics of the data, recognizing that past delays can have a significant impact on current and future delays, as shown in Figure 1.

Our approach to pre-processing the data by keeping past information as input showed that the model converged significantly better in the end. On the other hand, if we exclude historical data and restrict ourselves to operational data, the model fails to accurately predict reality. This underlines the importance of including historical data to obtain more reliable forecasts. Our methodology is based on the concept of capturing the complex relationships between the past, present, and future. We aim to model temporal wealth and leverage sequential information to its fullest extent, enhancing the model's predictive performance. This choice of architecture improves the model's ability to learn from long-term dependencies, which is crucial in the context of train delays, where complex temporal patterns can significantly influence future outcomes.

In our study aimed at predicting train delays, we delved into the utilization of the XGBoost model, a particularly robust and popular machine-learning technique. XGBoost, short for eXtreme Gradient Boosting, is an ensemble method based on decision trees. XGBoost excels in handling complex

data with non-linear relationships. It amalgamates multiple weak models (weak decision trees) to form a robust model. Its ability to handle heterogeneous datasets makes it an ideal choice for our problem, where train delays are influenced by a variety of factors. We can express the prediction of the XGBoost model $F(X)$, as follows:

$$F(X) = \sum_{k=1}^{K} f_k(X), \tag{6}$$

where K is the number of trees in the ensemble, f_k represents the prediction of the k-th tree, and X is the vector of explanatory variables [17].

A. BAYESIAN OPTIMIZATION FOR HYPERPARAMETER

A critical aspect of optimizing the performance of an XGBoost model lies in judiciously tuning its hyper-parameters. To achieve this effectively, we opted for a Bayesian optimization approach. This iterative method intelligently explores the optimal values for hyper-parameter θ, which would maximize the accuracy of our model. These hyper-parameters are essential for tuning the XGBoost model to achieve the right balance between complexity and generalization, thereby improving its predictive performance on unseen data. In the context of the XGBoost code, let B represent the objective function mathematically defined as follows according to [21].

$$B(\theta) = -\text{RMSE} \tag{7}$$

where: $\theta = \{$`max_depth`, `learning_rate`, `subsample`, `colsample_bytree`$\}$, are the hyper-parameters to be optimized.

- `max_depth`: the maximum depth of a tree. Deeper trees can capture more complex patterns in the data. A higher `max_depth` allows the trees to have more nodes, potentially capturing intricate patterns but can lead to over-fitting.
- `learning_rate`: the step size at each iteration while moving toward a minimum of the loss function. A smaller `learning_rate` makes the optimization process more robust by taking smaller steps, but it may require more boosting rounds.
- `subsample`: the fraction of observations that are randomly sampled to grow trees during the training process. A value less than 1.0 means that not all data is used for training each tree.
- `colsample_bytree`: the fraction of observations that are randomly sampled to grow each tree. Similar to `subsample`, it helps to prevent over-fitting by using a random subset of features for each tree.

$$\text{RMSE} = \sqrt{\frac{1}{p}\sum_{l=1}^{p}(y_l - \hat{y}_l)^2}, \tag{8}$$

$$\text{MAE} = \frac{1}{p}\sum_{l=1}^{p}|y_l - \hat{y}_l| \tag{9}$$

where:

- $\mathbf{p}$ is the total number of observations.
- $\mathbf{y_l}$ is the actual value of the target variable for observation l.
- $\hat{\mathbf{y}}_l$ is the predicted value for observation l.

In the Bayesian optimization process, these hyper-parameters are optimized to find the combination that minimizes the root mean squared error (RMSE) [22] in the cross-validated results [23]. The objective function B returns the negative of the mean of the root mean squared errors because the optimizer seeks to maximize the objective function. This mathematical representation captures the essence of the objective function as used in the code for Bayesian optimization with XGBoost.

The method described in the Algorithm 1 aims to explain the details of optimizing the hyper-parameters of an XGBoost. The parameter defines the potential ranges for each hyper-parameter, including ''maximum depth'', ''learning rate'', ''sub-sample ratio'', and ''column sub-sample ratio''. The objective is to minimize the Root Mean Squared Error (RMSE). Initially, θ_{best} is initialized as a zero mean function and is iterative updated by optimizing the acquisition $\alpha(\theta)$ to determine which hyper-parameters to evaluate. The optimization process balances exploration and exploitation by calculating the expected improvement and then evaluating the objective function with the negation of this improvement. The optimal hyper-parameters are updated based on the optimization results. The Algorithm 1 adapts to explore and exploit the hyper-parameter space θ_{best}, aiming to minimize the negative RMSE for the XGBoost model.

Using Bayesian Optimisation (BO), we can identify the hyper-parameters that significantly optimize the results of our predictions. We refine the model configuration by placing the optimal combination of parameters to achieve more accurate and reliable predictions. In summary, using BO proves to be an effective strategy for attaining well-tuned hyper-parameters, leading to a noticeable improvement in prediction performance.

$$\begin{aligned}\theta_{\text{best}} = \{&\\ \texttt{max\_depth} &= 6,\\ \texttt{learning\_rate} &= 0.01,\\ \texttt{subsample} &= 0.98,\\ \texttt{colsample\_bytree} &= 0.97\}\end{aligned}$$

Figure 3 illustrates the fundamental principle of our methodology, which employs a combined approach of Bayesian Optimization (BO) and the XGBoost algorithm to build a predictive model for predicting train arrival delays. This process begins with a training dataset representing 80% of the overall data. After this partitioning, we specifically apply Bayesian optimization to the training data to determine the optimal hyper-parameters for XGBoost, represented by θ, which maximizes model performance to achieve optimal predictive accuracy. We use the beta parameters in

Algorithm 1 Bayesian Optimization Algorithm for Hyper-Parameter Tuning

Data: θ^{space}, X^{all}
Result: θ^{best}
θ^{space}.`max_depth` $\leftarrow$ (3.30);
θ^{space}.`learning_rate` $\leftarrow$ (0.01, 1);
θ^{space}.`subsample` $\leftarrow$ (0.1, 1);
θ^{space}.`colsample_bytree` $\leftarrow$ (0.1, 1);
$B(\theta) \leftarrow -\text{RMSE}$;
$\theta^{\text{best}} \leftarrow 0$;
while θ^{best} *is not converged* **do**
  $\alpha(\theta) \leftarrow \mathbb{E}[\max(B(\theta^{\text{best}}) - B(\theta), 0)]$; /* Optimize acquisition function */
  $ov \leftarrow -B(\alpha(\theta))$; /* Sample the objective function */
  $\theta^{\text{best}} \leftarrow$ `update`(ov) ; /* Update Parameters based on optimization results */
  $X \leftarrow$ no ; /* Update Data with new observations *no* */
end

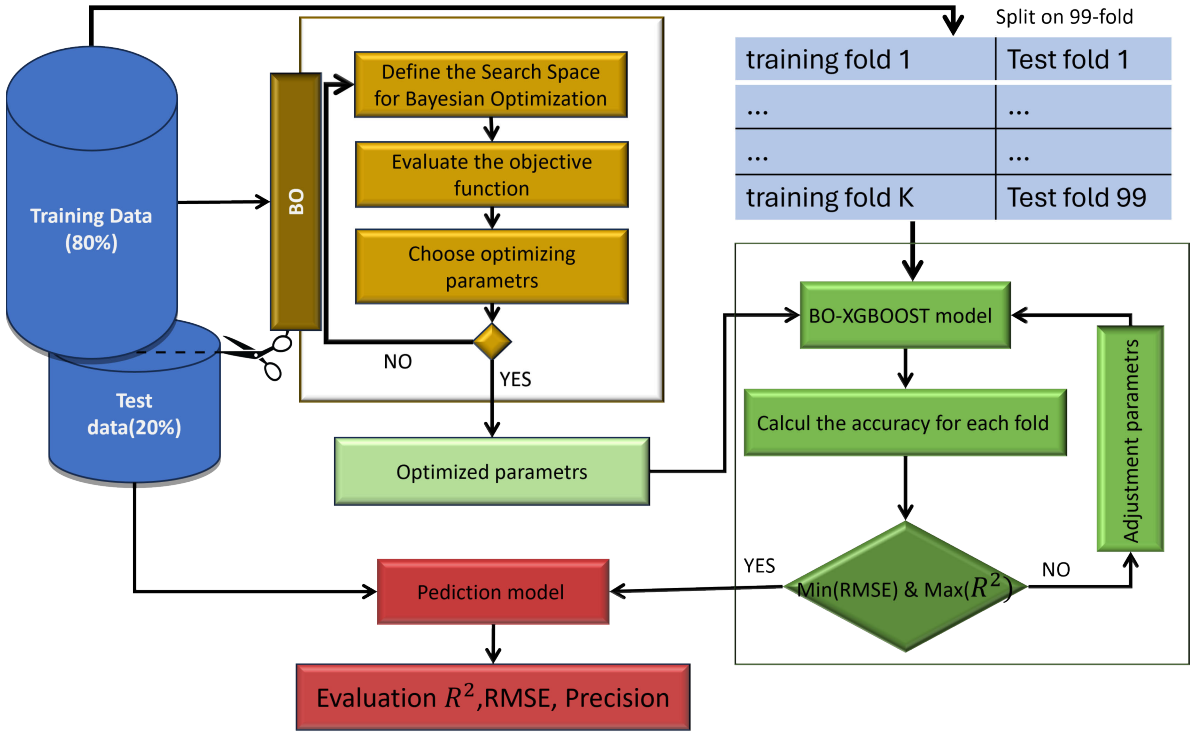

FIGURE 3. BO-XGBoost architecture for predicting train arrival delays.

our XGBOOST model, with the training data spread over 99 folds for cross-validation, where this choice is made according to a recursive function that tests divisions from 1 to 1000 and returns the best division by minimizing RMSE and maximizing R^2. This approach allows us to adapt the XGBoost algorithm to the specific characteristics of the dataset, improving its ability to capture complex relationships within the data. The model was trained using 1400 decision trees as shown in Figure V-A, and the selection of the 1400 trees, as explained for the function to optimize the number of folds, was determined by a recursive function that returns the optimal number of trees, as shown in Figure 3.

Once the model has been meticulously optimized, it undergoes rigorous testing on the test dataset, representing 20% of the overall data set spanning from December 12, 2017, to December 31, 2017. This evaluation phase is crucial for measuring the model's real-world performance on unseen data. Evaluation metrics, including accuracy, RMSE, and mean absolute error (MAE), are then scrutinized to validate the model's precision, and ensure its effective generalization to new data. The model's accuracy on the test dataset serves as a vital indicator of its ability to generalize effectively, thereby guaranteeing its relevance and robustness when faced with new, real-world data.

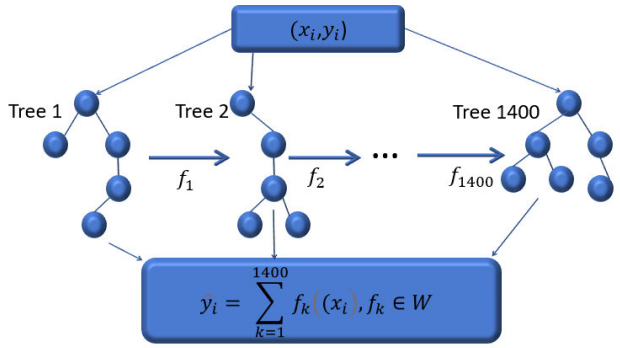

FIGURE 4. General architecture of XGBOST with the hyper-parameters used.

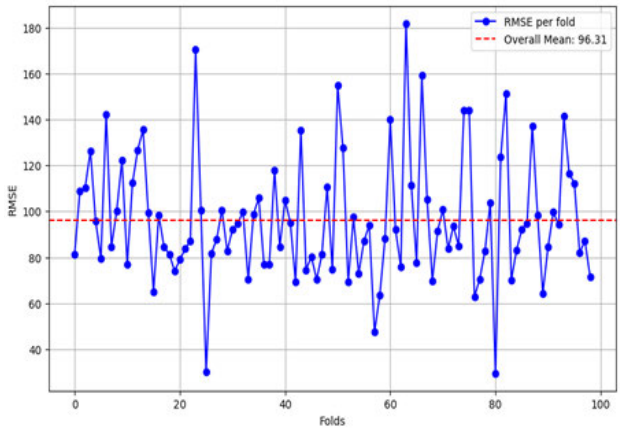

FIGURE 5. RMSE evaluation during XGBOOST training for 99-fold.

B. RESULT

We present the results of the evaluation of the test dataset, including performance measures and an analysis of the importance of variables in the XGBoost model. This analysis provides crucial information on the factors that confirm the accuracy of predictions on the data with our methodology, which is based on the importance of the past in predicting the future.

We begin our analysis of the results with the RMSE to validate the performance of the XGBoost model using 90 training trees. During the training phase, the model demonstrated remarkable accuracy with an RMSE of 96 seconds as shown in Figure 6, indicating a close match between predicted and actual values in the training set. This RMSE value underscores the model's efficiency in minimizing prediction errors while capturing the subtleties of relationships within the training data. However, during evaluation on the test set, the RMSE slightly increased to 107 seconds. While this variation may be attributed to the presence of unseen features in the test data, the model retains solid competence, displaying a reasonable gap between predicted and actual values. This observation highlights the need to balance the model's complexity during training while ensuring its ability to generalize effectively to new data.

After validating our model with RMSE, indicating good performance, Figure 6 presents two delay curves: one in blue for predictions and the other in orange for actual data. This

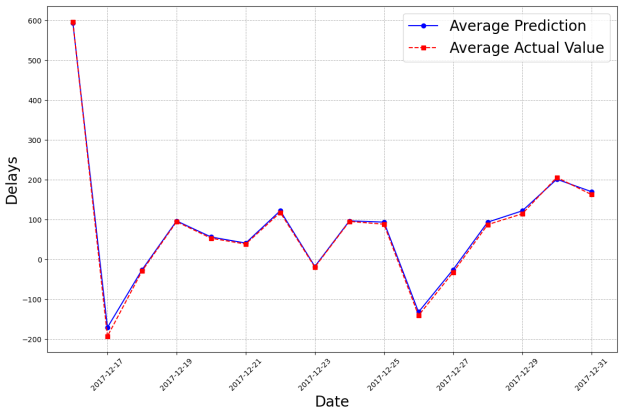

FIGURE 6. Average of predicted and real delays for each day of the test data.

figure demonstrates an almost perfect alignment between the two curves for the test data. For example, on December 17, 2027, there was a delay of -200 seconds in reality and −185 seconds in prediction, resulting in a very small error that increases at most to 10 seconds per day. However, for most days, predictions accurately match the actual data.

Figure 7, shown for each hour of the day over the testing period, illustrating delays as a function of time for each day from December 17, 2017, to December 31, 2017. These graphs highlight an increase in error between actual and predicted data, reaching up to 30 seconds for each hour. Nevertheless, there is a convergence in the rhythm of delays, as shown in Figure 6. Figure 8 represents plots of actual delays in blue, delays for predictions on training data in green, and in red for predictions on test data. Observing this graph suggests that both predictions follow a similar trend and remain close to reality. The performance metrics for the training data indicate an RMSE of approximately 100 seconds and an MAE of 60 seconds. Similarly, for the test data, the RMSE is around 107 seconds, and the MAE is approximately 65 seconds, as well as an R^2 (Coefficient of Determination) of about 0.94 for the training set and 0.92 for the test set, provides valuable insights into the model's accuracy and generalization capabilities. The RMSE values indicate the average magnitude of prediction errors, with lower values suggesting better precision. In this context, the model exhibits reasonable accuracy, as evidenced by the relatively low RMSE values. Furthermore, the R^2 values reflect the proportion of variance explained by the model, with higher values indicating a better fit. The high R^2 values for both training and test sets suggest that the model effectively captures the underlying patterns in the data and generalizes well to new observations. While the model performs slightly better on the training set, the overall metrics portray a robust and effective predictive model.

1) MODEL PERFORMANCE (SECOND VALIDATION)

The discussion of the results from the XGBoost-based model for predicting delays in trains and metros played a crucial role in validating the model's effectiveness. Experts

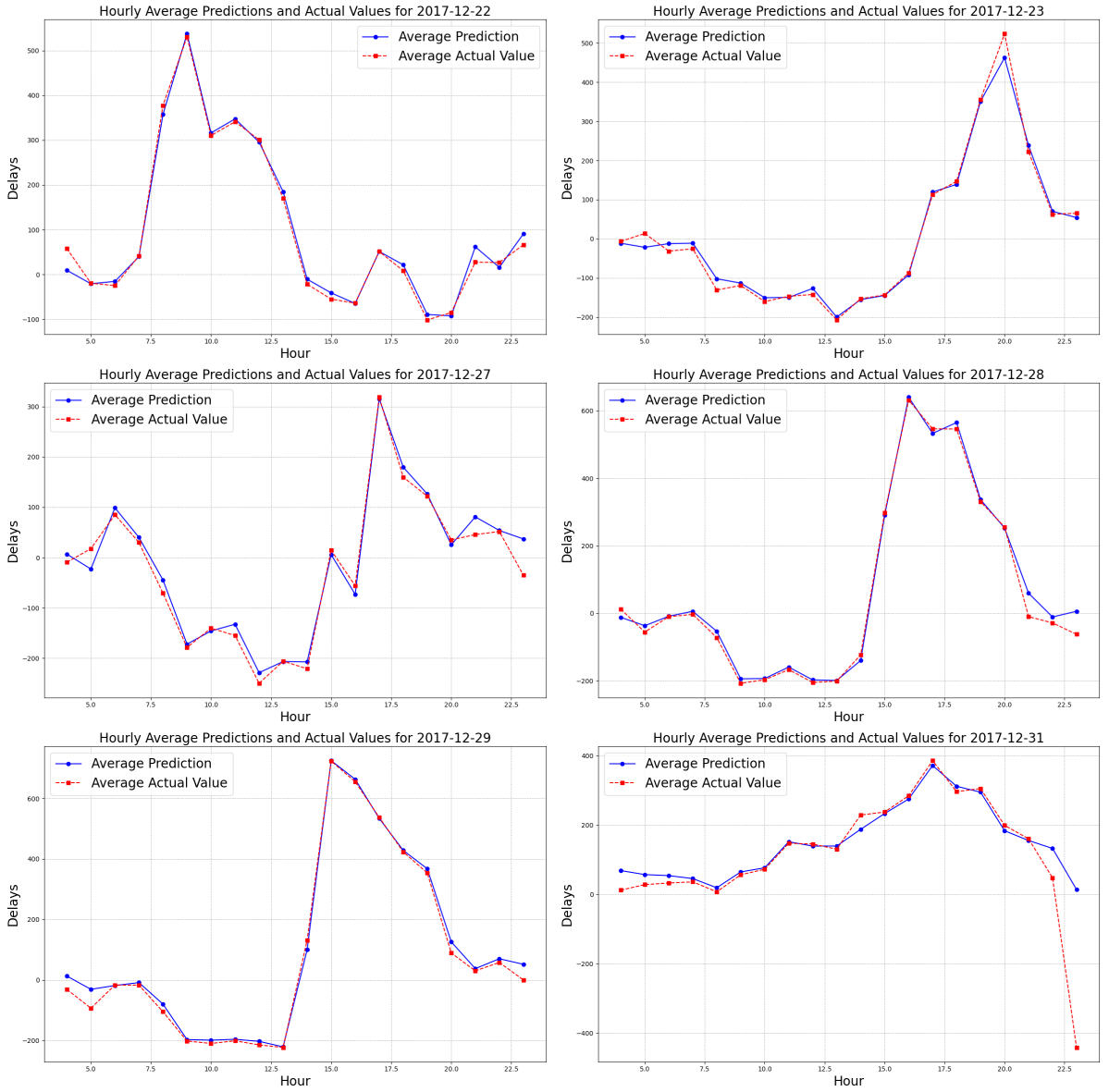

FIGURE 7. Average of predicted and real delays for each hour of the day on test data: 22-12-2017, 23-12-2017, 27-12-2017, 28-12-2017, 29-12-2017, 31-12-2017.

from the control center were pivotal in this assessment, aiming to determine whether the model's performance was acceptable for operational use. During the discussion, experts closely scrutinized the model's results in comparison to actual train delay data. A specific approach was adopted to assess the model's ability to anticipate delays, using the default margin of error defined in the control centers for each arrival. The experts decided to set a 5-minute error margin for arrivals, with a total tolerance of 150 seconds for delays and −150 seconds for advances. This approach has enabled the establishment of precise criteria for evaluating the model's accuracy in predicting both delays and advances. Consequently, it offers a comprehensive overview of the forecasting performance, outlined as follows:

$$E = \{e \in E \mid e = y - \hat{y}\} \tag{10}$$

Next, the set of admissible errors E is divided into two subsets TF (True Inside) and FI (False Inside) as shown in Figure 7, utilizing set notation:

$$\begin{aligned} E &= TI \cup TF \\ TI &= \{ti \in TI \mid -150 \leq ti \leq 150\} \\ TF &= \{tf \in TF \mid tf < -150 \text{ or } tf > 150\} \end{aligned}$$

Furthermore, we can express the precision P in terms of set cardinality, capturing the proportion of correct predictions within the total number of elements in the E set:

$$P = \frac{|TI|}{|TI| + |TF|} \tag{11}$$

This set-theoretic approach provides a comprehensive evaluation of the model's precision by distinguishing between correct predictions TI and incorrect predictions FI.

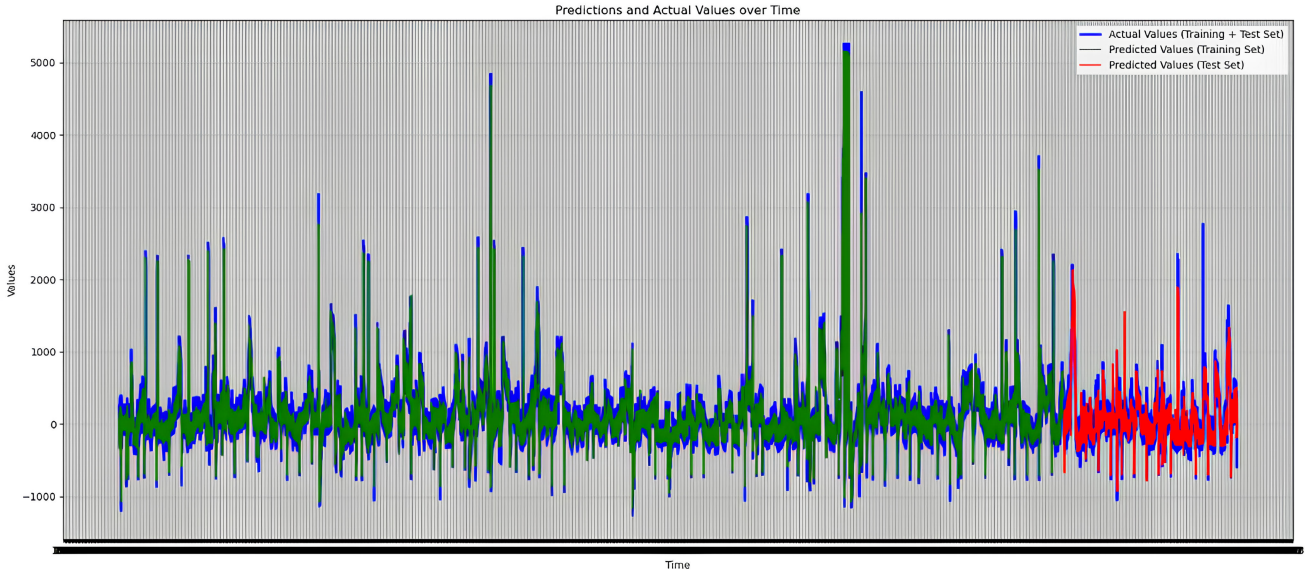

FIGURE 8. Visualisation of actual and predicted delays, the blue curve for actual delays, green for training data predictions, and red for test data predictions.

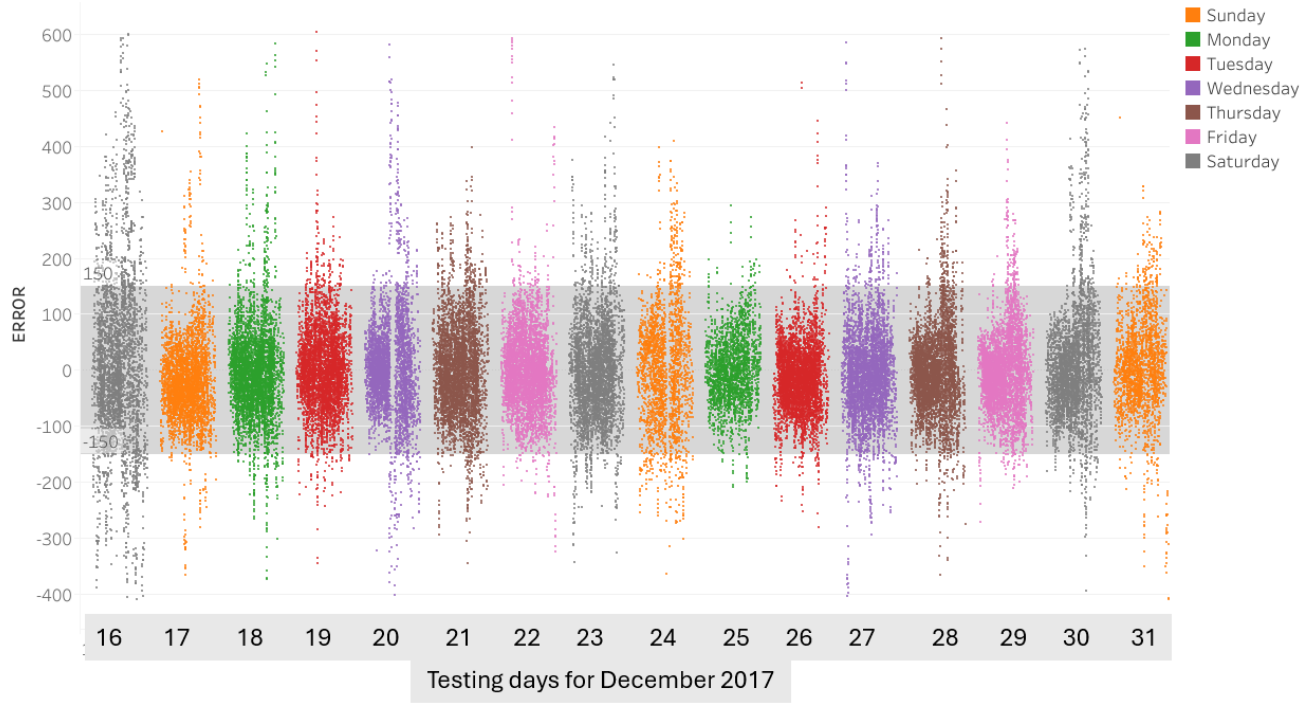

FIGURE 9. Distribution of prediction errors: Visualisation of the cloud of error points on the test data.

TABLE 2. Precision of margins above and below the expert's margins. Green is the margin of error determined by the rail experts.

Margin for Arrivals	-50 <TI	-100 <TI	-150 <TI	-200 <TI	-300 <TI	-350 <TI
Precision	62%	86%	95 %	98%	99.2%	99.8%

Subsequent analysis of the results demonstrated that the XGBoost model accurately predicted delays in trams and metros. Evaluating each arrival revealed that 95% of the test data fell within the defined error margin using the P(%) parameter. This observation indicates that the model can predict train delays with an impressive accuracy of 95% up to 15 minutes into the future, as depicted in Figure 9, which represents the error cloud set E. Each color in the figure 9 corresponds to the test data days, spanning from 16/12/2017 to 31/12/2017. Out of the 63029 data lines (departure times of trains for each station over the 16 days), 63 029 predictions of departure times were accurate

TABLE 3. Evaluation of XGBoost model performance at stations: Assessing RMSE, R^2, and accuracy within (−150, 150) Margin.

Stations	RMSE (seconds)	R^2	P (%)
1	88.59938	0.930428	96.89744
2	86.6763	0.943308	96.16667
3	101.4909	0.895058	96.63926
5	81.35647	0.946669	97.4022
6	90.487	0.937085	95.12742
7	96.01692	0.913406	96.30514
8	100.6197	0.922556	95.72506
9	97.66491	0.933273	92.23332
10	85.63531	0.944048	97.50649
11	104.579	0.888717	95.77991
12	91.38462	0.926697	96.23851
13	86.20662	0.944647	96.13595
14	87.44911	0.940571	95.6028
15	104.4072	0.917676	94.92426
16	102.5715	0.92644	92.18212
17	106.6195	0.917661	94.42376
18	109.6118	0.914776	94.18364
19	107.5393	0.915107	94.92426
20	94.20407	0.918784	96.23881
21	105.2779	0.918367	95.42762
22	107.619	0.917039	93.88652
Mean	96.953166	0.924396	95.426245

with a 95% precision, comprising 95% true positives *TI* and 5% false positives *FI*. This performance underscores the model's proficiency in forecasting train delays. This remarkable performance confirms the effectiveness of the XGBoost model in predicting delays, providing a robust foundation for its operational application in the realm of public transportation control.

The method of error margin proposed by railway control operators demonstrates remarkable precision. The initial margin, set at ±150 based on data with a 95% confidence level, provides reliable stability. A thorough analysis reveals that 86% of predictions are exceptionally accurate, with a narrow margin of ±100. This margin, defined as twice the standard deviation, encompasses nearly all data at 99%, within a range of ±300 as presented in Table 2, allowing for the identification of outliers that deviate from this range. It is noteworthy that the remaining 1%, associated with peak hours between 3:00 PM and 5:00 PM, appears to exceed the ±300 margin, suggesting temporary variations linked to the specific conditions of the French tourist city in question. These findings attest to the robustness of the predictive method while emphasizing the importance of considering peak-hour peculiarities in a specific urban context.

The analysis of train delay prediction results for a 15-minute future time window, using a one-hour interval from the past, reveals the overall satisfactory performance of the model. The results are presented in Tables 3, 4 and 5, evaluating key parameters such as R^2 (coefficient of determination), *RMSE* (root mean square error), and *P* (precision). In Table 3, each station displays notable values, with a maximum of 109 seconds for *RMSE*. Minimal values for R^2 and *P*, namely 0.89 and 93%, signify significant adequacy of the model for each station. Table 4 highlights the model's daily performance, showcasing an overall precision

TABLE 4. Daily performance of XGBoost model evaluation based on RMSE and R^2, with precision within the (−150, 150) margin blue color represents weekends (Saturday, Sunday).

Days (Year-Month-Day)	RMSE (seconds)	R^2	P (%)
2017-12-16	162.1656	0.51845	70.5778
2017-12-17	77.63573	0.953062	98.73544
2017-12-18	79.83413	0.932814	97.32024
2017-12-19	71.66906	0.93078	99.11815
2017-12-20	122.9476	0.745928	93.39921
2017-12-21	77.81046	0.541751	96.64347
2017-12-22	72.86071	0.887855	100
2017-12-23	87.91664	0.82518	94.50618
2017-12-24	113.8041	0.582692	88.30155
2017-12-25	55.70328	0.981923	100
2017-12-26	67.20878	0.970411	100
2017-12-27	79.46986	0.799008	87.26607
2017-12-28	134.4719	0.851053	97.30822
2017-12-29	89.39027	0.948621	99.5531
2017-12-30	111.833	0.9675	93.17691
2017-12-31	96.20438	0.643203	93.4526

TABLE 5. Performance per hour of XGBoost model evaluation based on RMSE and R^2, with precision within the margin (−150, 150).

Hours	RMSE (seconds)	R^2	P (%)
04:00:00	64.8314	0.450005	97.00748
05:00:00	89.07691	0.462186	91.69899
06:00:00	91.19509	0.69429	92.31683
07:00:00	61.0138	0.796814	97.23935
08:00:00	80.58421	0.928725	95.6621
09:00:00	74.10977	0.937086	95.19088
10:00:00	75.59375	0.877857	94.66862
11:00:00	83.5865	0.922545	94.37761
12:00:00	76.2212	0.944589	95.17225
13:00:00	87.95249	0.900181	93.95484
14:00:00	194.7441	0.902503	85.09327
15:00:00	108.8969	0.970915	98.3568
16:00:00	126.3103	0.949927	91.47162
17:00:00	109.2745	0.992199	99.3857
18:00:00	112.0794	0.988981	87.95407
19:00:00	102.547	0.998178	97.73494
20:00:00	107.6341	0.973451	96.13629
21:00:00	96.94439	0.985267	99.7193
22:00:00	106.2292	0.978576	95.55411
23:00:00	135.8437	0.932795	90.77157

of 100% for a quarter of the test data days. However, Table 5 underscores specific issues, particularly at the beginning of the day, at 4 a.m., where the model faces challenges with R^2 due to reliance on previous delays. Although the *RMSE* increases from 64 to 194 seconds at 2 p.m. due to real delay peaks, precision and R^2 remain high despite these challenges, demonstrating the model's robustness against variations in real-world data. Furthermore, it is crucial to note the relationship between precision (*P*) and *RMSE*, as a significant inverse correlation becomes apparent. When an increase in *RMSE* is observed at a specific hour, a simultaneous trend of decreasing precision occurs, as illustrated in the attached figure 8. This observation suggests a sensitivity of the model's precision to fluctuations in prediction errors, emphasizing the importance of closely monitoring these parameters together for a more thorough evaluation of the model's performance. Thus, any notable deviation in *RMSE* could directly impact the reliability of predictions, warranting particular attention in the management and optimization of the model.

TABLE 6. Comparative analysis of model performance with variable deletion.

XGBOOST Performance	All variables	Less variables
RMSE (seconds)	94	310
R^2	0.92	0.62

Correlation	T	S	PT	AI	A	D	P	PT	AT	Y^{past}	Y
Y	-0,01	0,02	0,20	0,05	0,01	-0,06	-0,06	-0,02	-0,09	0,84	1,00
Y^{past}	0,01	0,01	0,17	0,00	0,01	0,01	0,02	0,01	-0,04	1,00	0,84
AT	0,00	0,00	0,20	0,00	0,00	-0,02	0,00	0,00	1,00	-0,04	-0,09
PT	0,00	-0,10	0,00	-0,60	-0,59	0,00	0,00	1,00	0,00	0,01	-0,02
PI	0,00	0,00	0,14	0,02	0,02	0,19	1,00	0,00	0,00	0,02	-0,06
D	0,00	0,02	-0,07	0,00	0,00	1,00	0,19	0,00	-0,02	0,01	-0,06
A	0,00	0,25	0,02	0,55	1,00	0,00	0,02	-0,59	0,00	0,01	0,01
AI	0,00	0,16	0,02	1,00	0,55	0,00	0,02	-0,60	0,00	0,00	0,05
PD	0,02	0,02	1,00	0,02	0,02	-0,07	0,14	0,00	0,20	0,17	0,20
S	0,01	1,00	0,02	0,16	0,25	0,02	0,00	-0,10	0,00	0,01	0,02
T	1,00	0,01	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,01	-0,01

FIGURE 10. X Correlation table: Red for strong correlation, yellow for moderate, and blue or white for no correlation.

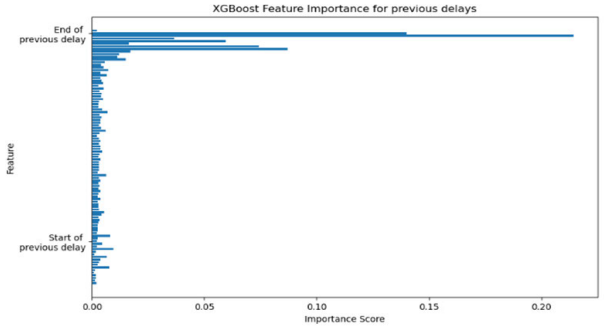

FIGURE 11. XGBoost feature importance for previous delays over a one-hour interval (Y^{past}).

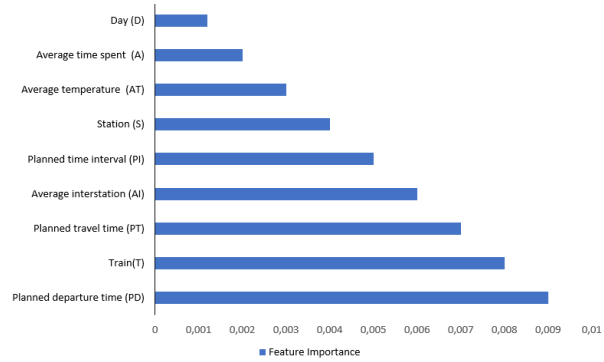

FIGURE 12. XGBoost feature importance plot for operational data (*XOP*).

The XGBoost-based model has demonstrated exceptional accuracy and robustness in predicting train arrival delays within 15 minutes of future arrivals, providing a reliable tool for public transport control. Accuracy analysis, operational validation, and a detailed robustness assessment of the model demonstrate its effectiveness and its ability to predict the delay for every train leaving a station in order to use these results in the control center. During the XGBoost model training, all variables used proved to be important for model convergence, even in the absence of clear correlations between the input variables X and our output variable Y, representing the delay, as illustrated in Figure 10. Examining the last column depicting the correlation between the delay and other variables does not reveal significant correlations, with values ranging between -0.06 and 0.14, except for the average of previous delays, which exhibits a strong correlation of 0.84 with the delay shown in Figures 11 and 12, depicting variable importance after model training, show that all variables have increasing importance up to 0.008. In contrast, the importance of previous delays in our model reaches a significant value of 0.20. These results emphasize that even in the absence of clear linear correlations, previous delays play a crucial role in predicting the current delay, and their substantial importance in the model reflects their significant contribution to the overall model performance. During model experimentation, we conducted tests by eliminating variables with low correlation with the output, aiming to simplify the model without compromising performance. However, the results indicated that the model did not converge satisfactorily in this configuration. The *RMSE* increased significantly, reaching 4 minutes, while the coefficient of determination R^2 decreased to 0.63 shown in Figure 10. This observation can be explained by the fact that, even if certain variables do not exhibit clear individual correlations with the output, they might collectively contribute to delayed predictions. The removal of these variables resulted in a loss of essential information, hindering the model's ability to capture the complexity of relationships between input and output data. Therefore, while the intention to simplify the model is understandable, it is crucial to consider the overall impact of variables on model performance, as arbitrary removal may lead to a degradation of the model's predictive capacity.

VI. CONCLUSION

The successful deployment of XGBoost in real-world railway operations to predict delays has proven to be a significant breakthrough in public transportation. Close collaboration with control center experts has ensured the robustness and reliability of the predictive model, allowing railway operators to make informed decisions quickly and efficiently. By providing a 15-minute prediction window with an accuracy rate of 95%, the model has empowered control center operators to manage traffic, mitigate disruptions, and take timely actions to minimize the impact of delays on schedules and passenger experience. This strategic approach has not only optimized operational efficiency but has also elevated the overall quality of service for passengers, setting a new standard for real-time delay prediction in railway systems.

Furthermore, it's important to note that while our study is based on real train data, the methodology is equally applicable to metro systems. Whether the model is dynamically refined for each use case or kept static for daily or weekly predictions remains a question, as the current dataset does not allow for experimental determination of the need for retraining with the latest data, be it from the last hours, day, month, or year. These questions merit exploration in future articles with expanded datasets.

Looking ahead, there are exciting opportunities to further enhance the predictive capabilities of XGBoost and other machine learning algorithms in railway operations. Future research has the potential to delve deeper into the integration of diverse data sources, including events throughout the day, passenger flow patterns, passenger counting, and traveler behaviors. Additionally, exploring the inclusion of sources like maintenance logs and feedback from passengers could further enrich the dataset, contributing to the development of more comprehensive and precise prediction models. The seamless integration of predictive capabilities into real-time display systems at control centers has the potential to revolutionize how traffic flow is optimized and on-time performance is ensured in urban rail systems. By harnessing the power of advanced analytics and artificial intelligence, railway operators can proactively address challenges, anticipate disruptions, and deliver a more resilient and responsive transportation network.

Embracing innovation and continuous improvement will be key in shaping the future of delay prediction and management in railway operations, ultimately leading to a more efficient, and sustainable. As a critical next step, exploring how to integrate these predictive insights into operators' displays for maximum simplicity and usability will be crucial. This ongoing commitment to advancement ensures that our rail systems not only meet but exceed the expectations of efficiency and reliability in the years to come.

REFERENCES

[1] H. Alawad, S. Kaewunruen, and M. An, "Learning from accidents: Machine learning for safety at railway stations," *IEEE Access*, vol. 8, pp. 633–648, 2020.

[2] F. Corman and L. Meng, "A review of online dynamic models and algorithms for railway traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1274–1284, Jun. 2015. [Online]. Available: https://ieeexplore.ieee.org/document/6920082/

[3] R. Shi, X. Xu, J. Li, and Y. Li, "Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107538. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494621004610

[4] Y. Qu, Z. Lin, H. Li, and X. Zhang, "Feature recognition of urban road traffic accidents based on GA-XGBoost in the context of big data," *IEEE Access*, vol. 7, pp. 170106–170115, 2019, doi: 10.1109/ACCESS.2019.2952655.

[5] X. Bao, Y. Li, J. Li, R. Shi, and X. Ding, "Prediction of train arrival delay using hybrid ELM-PSO approach," *J. Adv. Transp.*, vol. 2021, pp. 1–15, Jun. 2021. [Online]. Available: https://www.hindawi.com/journals/jat/2021/7763126/

[6] N. Marković, S. Milinković, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transp. Res. C, Emerg. Technol.*, vol. 56, pp. 251–262, Jul. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X1500145X

[7] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167278919305974

[8] A. Berger, A. Gebhardt, M. Müller-Hannemann, and M. Ostrowski, "Stochastic delay prediction in large train networks," in *Proc. 11th Workshop Algorithmic Approaches Transp. Modelling, Optim., Syst.*, 2011, pp. 100–111, doi: 10.4230/OASIcs.ATMOS.2011.100.

[9] F. Corman and E. Quaglietta, "Closing the loop in real-time railway control: Framework design and impacts on operations," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 15–39, May 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X15000169

[10] T. Büker and B. Seybold, "Stochastic modelling of delay propagation in large networks," *J. Rail Transp. Planning Manage.*, vol. 2, nos. 1–2, pp. 34–50, Nov. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2210970612000182

[11] A. Núñez, J. Hendriks, Z. Li, B. De Schutter, and R. Dollevoet, "Facilitating maintenance decisions on the Dutch railways using big data: The ABA case study," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Washington, DC, USA, Oct. 2014, pp. 48–53, doi: 10.1109/BIGDATA.2014.7004431.

[12] S. Harrod, F. Cerreto, and O. A. Nielsen, "A closed form railway line delay propagation model," *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 189–209, May 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X18310878

[13] W. Mou, Z. Cheng, and C. Wen, "Predictive model of train delays in a railway system," in *Proc. 8th Int. Conf. Railway Oper. Model. Anal. RailNorrkoping*, 2019, pp. 1–17.

[14] P. Taleongpong, S. Hu, Z. Jiang, C. Wu, S. Popo-Ola, and K. Han, "Machine learning techniques to predict reactionary delays and other associated key performance indicators on British railway network," *J. Intell. Transp. Syst.*, vol. 26, no. 3, pp. 311–329, May 4, 2022. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/15472450.2020.1858822

[15] T. Xiahou, Y.-X. Zheng, Y. Liu, and H. Chen, "Reliability modeling of modular K-out-of-N systems with functional dependency: A case study of radar transmitter systems," *Rel. Eng. Syst. Saf.*, vol. 233, May 2023, Art. no. 109120. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0951832023000352

[16] A. Selvaraj, M. Madhumitha, J. Jeevajothi, and B. Monika, "Train delay prediction using support vector machine," *Int. Res. J. Modernization Eng. Technol. Sci.*, vol. 5, no. 3, pp. 1–7, 2023.

[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA, Aug. 13, 2016, pp. 785–794. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939785

[18] X. Yao, X. Fu, and C. Zong, "Short-term load forecasting method based on feature preference strategy and LightGBM-XGboost," *IEEE Access*, vol. 10, pp. 75257–75268, 2022, doi: 10.1109/ACCESS.2022.3192011.

[19] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, 2021, doi: 10.1109/ACCESS.2021.3107975.

[20] Z. Huang, X. Jia, L. Mi, Y. Cai, and J. Li, "Optimization of train timetables in high-speed railway corridors considering passenger departure time and price preferences," *IEEE Access*, vol. 12, pp. 14964–14986, 2024.

[21] T. T. Joy, S. Rana, S. Gupta, and S. Venkatesh, "Hyperparameter tuning for big data using Bayesian optimisation," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2574–2579, doi: 10.1109/ICPR.2016.7900023.

[22] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE, and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, p. e623, Jul. 2021. [Online]. Available: https://peerj.com/articles/cs-623

[23] G. Baron and U. Stańczyk, "Standard vs. non-standard cross-validation: Evaluation of performance in a space with structured distribution of datapoints," *Proc. Comput. Sci.*, vol. 192, pp. 1245–1254, May 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050921016173

SONDOSS CHTIOUI received the bachelor's degree in computer engineering from the Faculte Polydisciplinaire de Taza, Morocco, in 2019, the Master of Science (M2) degree in data science and artificial intelligence from Universite Sorbonne Paris Nord, Paris, France, in 2021, and the master's degree in web intelligence and data science from Sidi Mohammed Ben Abdellah University, Fez, Morocco, in 2021. She is currently pursuing the Ph.D. degree in engineering with ESTACA Laval, Paris-Saclay, Paris, France, where she will focus on a research project on the integration of a decision support system in a railway control center.

Her recent publication in the Proceedings of the Seventh International Conference (ICITT 2023), held in Madrid, Spain, is titled "Exploring the relational model of the Metro in decision supporting systems."

SEBTI MOUELHI received the M.S. degree in computer science from the University of Lorraine, Nancy, France, in 2007, and the Ph.D. degree in computer science from the University of Franche-Comte, Besancon, France, in 2011. He was a Postdoctoral Researcher with INRIA, Grenoble, France, in 2011. He was a Research and Development Engineer with SafeRiver, Montrouge, France, for approximately three years, in Fall 2012. In 2015, he was an Engineer in safety assurance with ALSTOM Transport, Saint-Ouen, France. He has been a Lecturer and a Researcher in embedded systems with ECE—Ecole d'ingenieurs, OMNES Education, Paris, France, for five years, since 2015. In September 2020, he joined ESTACA, Paris-Saclay Campus, as an Associate Professor. His current research interests include vehicular networks, formal methods, software engineering, embedded systems, real-time, automatic control, machine learning, and applied category theory.

SÉBASTIEN SAUDRAIS received the Ph.D. degree from the University of Rennes 1, in 2007. During his thesis, he worked on the integration of time issues into component-based Applications with IRISA/INRIA, Rennes, France. Then, he joined the Distributed System Group, Trinity College Dublin, on research works about using specification models for runtime adaptations for the European project ALIVE. He is currently an Associate Professor in computer science for embedded systems with the Engineering School (ESTACA), Laval, France. He is also working on human mobility.

TOUFIK AZIB (Member, IEEE) received the Diploma degree in electrotechnical engineering from the University of Setif (UFAS), Algeria, in 2006, the M.Sc. degree in electrical engineering from the National Superior School of Electrical and Mechanical Engineering of Nancy, ENSEM-INPL, France, in 2007, the Ph.D. degree in electrical engineering from the University of Paris South XI, France, in 2010, and the H.D.R. (Habilitation ã Diriger des Recherches) (H.D.R.: French Postdoctoral) degree from the University of Paris-Saclay, France, in 2021. Since 2011, he has been an Associate Professor with the Engineering Research Center ESTACA'Lab, ESTACA Engineering School, Saint-Quentin-en-Yvelines, France. He is currently a Full Professor and the Head of the S2ET Department (Energy and Embedded Systems for Transportation). His current research interests include optimal design of power electronics and control/energy management of new electrical devices (fuel cell, battery, ultracapacitor, and photovoltaic) for new mobility applications (green, sustainable, and smart mobility).

MARC ILLE has acquired and developed expertise in systems. His expertise initially focused on rail systems with Ansaldo STS, with a particular focus on signaling and supervision (PCC), metro, and LGV, and on-board driver assistance equipment (ERTMS) for rolling stock. Then, he developed his expertise in transport systems with Systra as the Deputy Director of the AMO-MOE Department in charge of rail equipment.

MELANIE MOREL received the Ph.D. degree in cognitive ergonomics and 24 years of working experience in human factors (HF) domain. She has a strong expertise in HF methods and process development for industrial projects ensuring the understanding and consideration of human behavior in the design and validation. In this context, she has handle a lot of HF studies in the development of complex and safety critical systems (for aircraft car cockpits and control rooms). After three years of working experience with Renault, with 18 years with EGIS, she is also an HF Expert, has contributed significantly to various domains, including designing operating control centers across nuclear, road, and rail sectors, providing HF support to the European Commission under the Single European Sky Initiative, developing HF methodologies for Airbus and EDF, training EASA experts in cockpit design and certification, and adeptly managing contracts.

FREDERIC ORU received the Doctor of Mathematics degree from the Ecole Normale Superieure, France. He is currently an Entrepreneur and a Research Engineer specialized in artificial intelligence. He is also a Graduate Engineer with the Ecole Polytechnique. He has international experience of over 20 years at the crossroads of the worlds of scientific research, large companies, and startups. He worked for ten years in the IT services and training sector where he led strategic and international projects for large companies. He spent the next ten years in the world of digital startups and is one of the founders of NUMA, the pioneer organization in the dissemination of innovation methods for companies and startups. He led the international development of the company in six countries and four continents. He has overseen international open innovation programs focused on "data," such as "DataCity" (AI solutions for the smart city), "Digital Industry" (AI solutions for industry), and "AI Hub" (artificial intelligence startup accelerator in Bengaluru, India). In 2019, he founded a new company, "AI4Better," with the ambition to bring cutting-edge AI technology to entrepreneurs who contribute to a better world. Since 2023 he is the AI scientific and technical director of Egis Group.