International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Optimizing Freight Rail Arrival Times with Machine Learning

Duggirala Kanakasatya

Assistant Professor

Abstract

This project focuses on enhancing the accuracy of freight rail arrival time predictions using machine learning. Traditional estimation methods relying on distance and travel time were found to be unreliable. Various modern machine learning models, including LIGHTGBM, Random Forest, KNN, and Linear Regression, were assessed, with LIGHTGBM initially demonstrating the highest predictive accuracy (R2). The study also integrates SHAPLEY values to interpret model performance and emphasizes data preprocessing techniques such as feature selection and normalization. Furthermore, the advanced XGBOOST algorithm was tested, achieving an improved R2 score of 91%. The findings indicate that these models significantly improve short-term arrival delay forecasting, which is essential for optimizing freight rail operations.

Keywords: LIGHTGBM, XGBOOST

INTRODUCTION:

The growing importance of rail transport due to its cost-effectiveness, reliability, lower emissions, and safety, which has led to its integration into intermodal transport systems. Public agencies are actively promoting rail over other transport alternatives. However, optimizing rail infrastructure is crucial for maintaining efficiency and dependability, which requires effective management, maintenance, and the application of technology and data analytics.

A key challenge in rail operations is train delays, which can arise from preparation time issues and in-transit performance variations. Predicting arrival delays—defined as the difference between actual and scheduled arrival times—is essential for risk management in rail transport. The paragraph contrasts event-based models, which forecast disruptions, with data-driven models that are better at recognizing complex, multidimensional data relationships.

The research is focuses on evaluating and comparing data-driven models for predicting short-term arrival delays. Additionally, it aims to identify key delay factors and develop a **Short-term Decision Support System (STDSS)** to help mitigate real-time disruptions in freight operations. The emphasis on data-driven models suggests that the study is leveraging advanced analytics and machine learning techniques to enhance rail system reliability.

LITERATURE SURVEY:

P. Mcmahon et al The advent of big data has transformed railway asset management by enabling the collection and analysis of extensive operational datasets. Advanced data analytics, including machine learning and multiple regression analysis, play a crucial role in improving decision-making processes.

These technologies help identify critical insights, such as early indicators of potential failures. This paper examines the requirements and challenges associated with big data analytics in railway asset management, with a focus on data acquisition, processing, and practical applications, including the role of Blockchain technology. It highlights the significance of leveraging big data and outlines future research directions to address existing gaps in the field.

Railway operations are often impacted by disturbances affecting a preceding train, which can reduce service intervals below the required headway distance. This results in increased train interactions, leading to higher energy consumption, delays, and financial penalties. This study investigates optimizing train trajectories to achieve a balance between energy efficiency and delay mitigation within a fixed block signaling system. A custom-built multitrain simulator is used to analyze the impact of a leading train's movements on following trains. Optimization techniques such as enhanced brute force search, ant colony optimization, and genetic algorithms are applied to determine optimal train trajectories. Findings suggest that well-planned train movements can minimize disruptions, improve energy efficiency, enhance operational performance, and contribute to safer and more comfortable travel experiences.

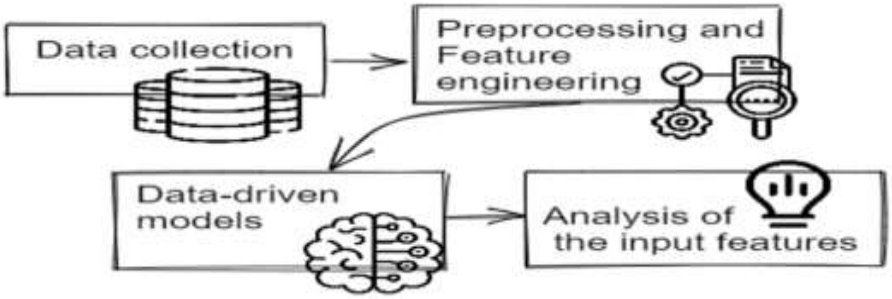
PROBLEM STATEMENT:

Freight rail is a crucial mode of transportation for moving goods between locations, ensuring the efficient delivery of everyday items produced in different regions. As one of the most extensive transport networks, rail plays a vital role in supply chain logistics. To optimize freight movement, operators need accurate departure and travel time estimates to predict arrival times with precision. Historically, arrival times were determined using basic distance and travel time calculations. However, this approach often led to inaccuracies, highlighting the need for more advanced and reliable prediction methods.

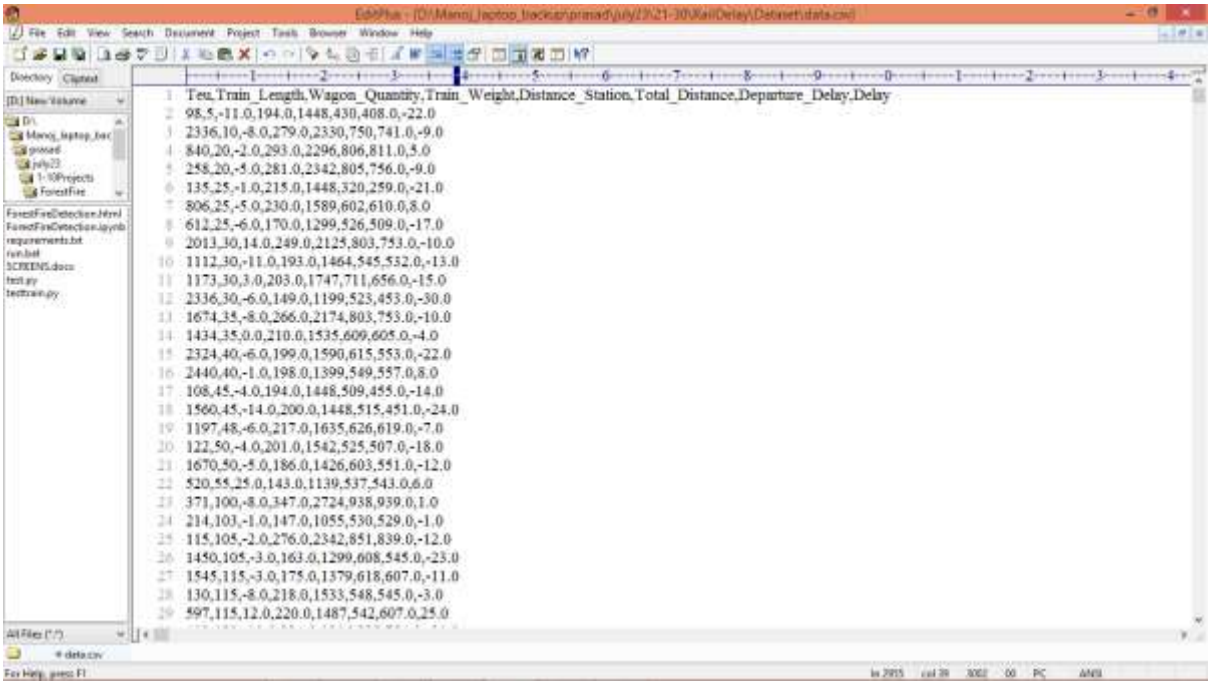
PROPOSED METHOD:

Machine learning models have been implemented to forecast train arrival times, yet they often struggle with accuracy in short-term delay predictions. Short-term arrival delay estimation refines predictions by calculating delays at each station, allowing for more precise forecasts at subsequent stops. This delay is factored into the actual arrival, scheduled arrival, scheduled departure, and departure delay. In this study, a real-time dataset from the National Rail Company of Luxembourg was used to train multiple machine learning algorithms, including LIGHTGBM, Random Forest, KNN, and Linear Regression. The models were assessed using performance metrics such as RMSE, MAPE, MAE, and R^2. Lower RMSE, MAPE, and MAE values indicate better predictive performance, while a higher R^2 signifies greater accuracy. Among the tested models, LIGHTGBM demonstrated the highest R^2 value, confirming its superior ability to predict train arrival times accurately.

ARCHITECTURE:



SHORT TIME DELAY DATASET:



```
Teu,Train_Length,Wagon_Quantity,Train_Weight,Distance_Station,Total_Distance,Departure_Delay,Delay
98,5,-11.0,194.0,1448,430,408.0,-22.0
2336,10,-8.0,279.0,2330,750,741.0,-9.0
840,20,-2.0,293.0,2296,806,811.0,5.0
258,20,-5.0,281.0,2342,805,756.0,-9.0
135,25,-1.0,215.0,1448,320,259.0,-21.0
806,25,-5.0,230.0,1589,602,610.0,8.0
612,25,-6.0,170.0,1299,526,509.0,-17.0
2013,30,14.0,249.0,2125,803,753.0,-10.0
1112,30,-11.0,193.0,1464,545,532.0,-13.0
1173,30,3.0,203.0,1747,711,656.0,-15.0
2336,30,-6.0,149.0,1199,523,453.0,-30.0
1674,35,-8.0,266.0,2174,803,753.0,-10.0
1434,35,0.0,210.0,1535,609,605.0,-4.0
2324,40,-6.0,199.0,1590,615,553.0,-22.0
2440,40,-1.0,198.0,1399,549,557.0,8.0
108,45,-4.0,194.0,1448,509,455.0,-14.0
1560,45,-14.0,200.0,1448,515,451.0,-24.0
1197,48,-6.0,217.0,1635,626,619.0,-7.0
122,50,-4.0,201.0,1542,525,507.0,-18.0
1670,50,-5.0,186.0,1426,603,551.0,-12.0
520,55,25.0,143.0,1139,537,543.0,6.0
371,100,-8.0,347.0,2724,938,939.0,1.0
214,103,-1.0,147.0,1055,530,529.0,-1.0
115,105,-2.0,276.0,2342,851,839.0,-12.0
1450,105,-3.0,163.0,1299,608,545.0,-23.0
1545,115,-3.0,175.0,1379,618,607.0,-11.0
130,115,-8.0,218.0,1533,548,545.0,-3.0
597,115,12.0,220.0,1487,542,607.0,25.0
```

In above dataset screen first row contains dataset column names and remaining rows contains dataset values and in last column we have Short Time Delay as target value which is in minutes from junction to junction.

METHODOLOGY:

Data Loading and Preprocessing:

Import the necessary Python libraries and packages.

Load the dataset from a CSV file into a Pandas DataFrame.

Handle missing values by replacing them with zeros.

Detect and remove highly correlated features to prevent multicollinearity.

Feature Importance Analysis:

The LightGBM (LGBM) algorithm is employed to assess the impact of different features on delay prediction.

Feature importance scores are computed and visualized to highlight the most influential factors.

Data Normalization and Train-Test Split:

MinMaxScaler is used to normalize both the features and the target variable (delay).

The dataset is divided into training (80%) and testing (20%) subsets.

The total count of records and features, as well as the sizes of the training and testing sets, are displayed.

Visualizing Train-Test Similarity:

The distribution of departure delay values in the training and testing datasets is visualized using box plots to ensure similarity.

Model Training and Evaluation:

Five machine learning algorithms are trained and evaluated:

LightGBM Regressor

Linear Regression

Random Forest Regressor
K-Nearest Neighbors (KNN) Regressor
XGBoost Regressor (Extension)
Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R2) values are calculated for each algorithm.
The original and predicted delay values are plotted for visual comparison.

Interpreting XGBoost Predictions:

SHAP (SHapley Additive exPlanations) values are used to interpret the predictions made by the XGBoost algorithm.
Feature importance is explained using a summary plot.

Comparing Algorithm Performance:

A bar graph is plotted to compare the performance of all algorithms based on R2 and RMSE values.

Summary of Algorithm Performance:

A summary table is presented, showing the R2, MAE, RMSE, and MAPE values for each algorithm.

Predicting Delay for Test Data:

Load test data from a separate CSV file.
Handle missing values by replacing them with zeros.
Normalize the test data using the same MinMaxScaler applied to the training set.
Use the XGBoost algorithm to predict delay values.
Denormalize the predicted delay values and display them alongside the corresponding test data.

EVOLUTION:

Mean Squared Error (MSE):

MSE is calculated by taking the average of the squared differences between the predicted values and the actual values.

$$\text{MSE} = \frac{1}{n} \quad \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

#Calculate Mean Squared Error (MSE)

```
def mean_squared_error(actual, predicted):
mse = np.mean((actual - predicted)**2)
return mse
```

Root Mean Squared Error (RMSE):

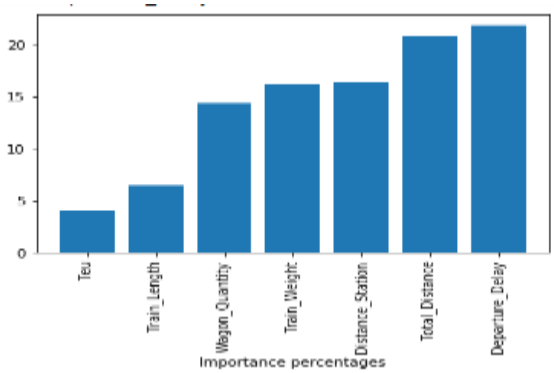
RMSE is the square root of the MSE and provides a measure of the average magnitude of the errors in the predictions.

$$\text{RMSE} = \sqrt{\text{MSE}}$$

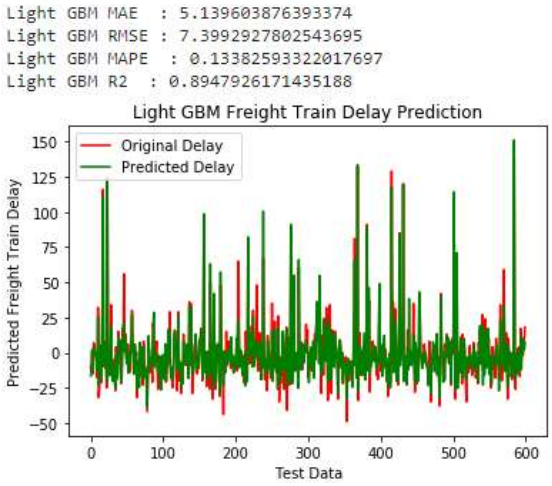
#Calculate Root Mean Squared Error (RMSE)

```
def root_mean_squared_error(actual, predicted):
mse = mean_squared_error(actual, predicted)
rmse = np.sqrt(mse)
return rmse
```

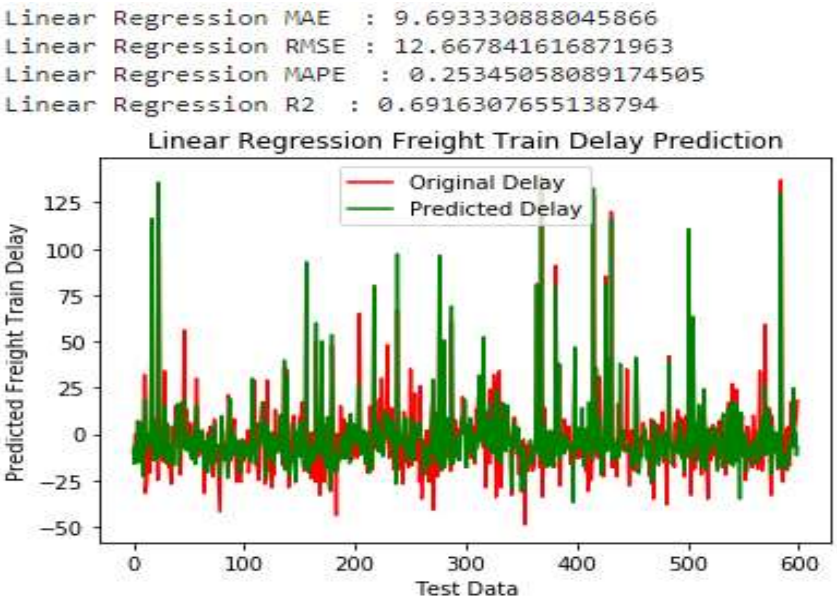
RESULTS:



X-axis represents feature name and y-axis represents importance values

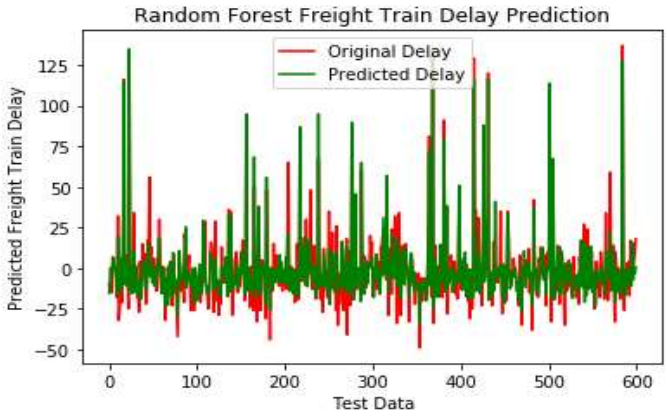


Training LIGHTGBM algorithm on training data and then performing prediction on test data and then in blue colour line we can see R2 value as 89%



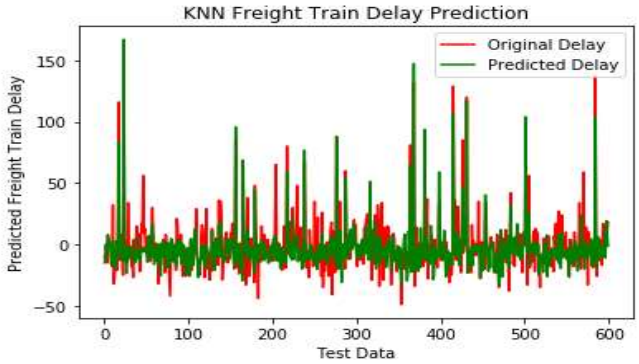
Linear Regression and its R2 values is 69%

```
Random Forest MAE   : 7.118916666666667
Random Forest RMSE  : 9.807943506838386
Random Forest MAPE  : 0.18396082938523373
Random Forest R2    : 0.8151490425798404
```



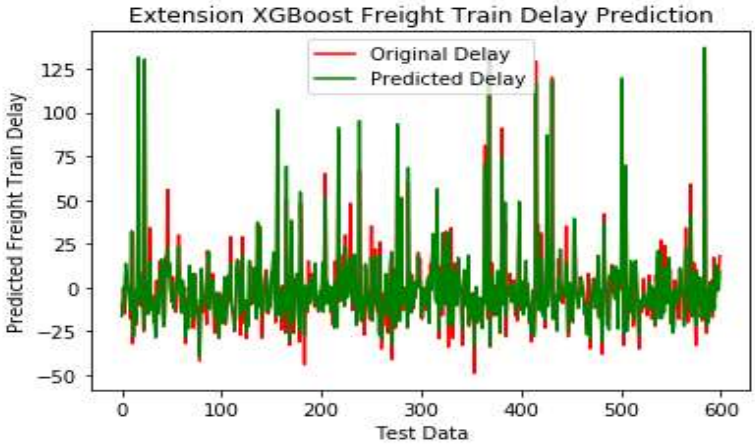
Random Forest algorithm and its R2 values is 81%

```
KNN MAE   : 10.256111111111114
KNN RMSE  : 14.055386472412948
KNN MAPE  : 0.2545302246028417
KNN R2    : 0.6203779970661356
```

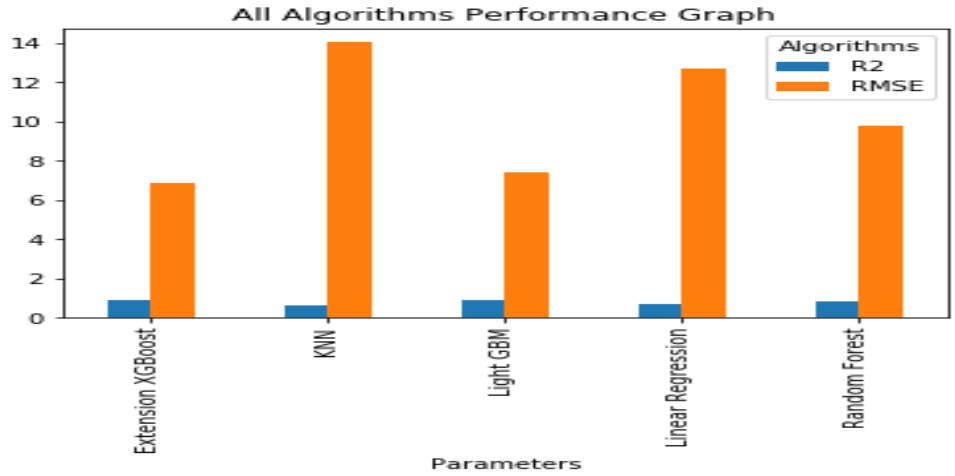


KNN and its R2 value is 62%

```
Extension XGBoost MAE   : 4.865689219682633
Extension XGBoost RMSE  : 6.828325684326598
Extension XGBoost MAPE  : 0.12226352326118042
Extension XGBoost R2    : 0.9104028393272756
```



Extension XGBOOST and its R2 value is 91%



X-axis represents algorithm names and y-axis represents R2 and RMSE error and in all algorithms Extension XGBOOST and Light GBM got high R2 and less RMSE error compare to other algorithms

Prediction:

```
Test Data = [1500.  510.    4.  182.  925.  805.  816.]
Predicted Delay ====> 10.05116

Test Data = [ 399.  700.   -5.  392. 2475. 1025. 1027.]
Predicted Delay ====> 2.9311402

Test Data = [1133.  740.   -3.  131.  647. 1025. 1048.]
Predicted Delay ====> 20.413258

Test Data = [742. 740.  -1. 108. 406. 910. 927.]
Predicted Delay ====> 11.702452

Test Data = [1104.  758.   18.  143.  862. 1115. 1139.]
Predicted Delay ====> 24.17175

Test Data = [1033.  840.   54.  243. 1671. 1445. 1537.]
Predicted Delay ====> 52.51954

Test Data = [ 835.  935.    9.  335. 2384. 1310. 1319.]
Predicted Delay ====> 9.217901
```

Reading test data from test file and then predicting delay using XGBOOST extension object and predicted delay

CONCLUSION

This project focuses on enhancing the accuracy of freight train arrival time predictions by addressing short-term arrival delays. Traditional estimation methods relying on distance and travel time often resulted in inaccuracies, prompting the adoption of machine learning models. Utilizing real-time data from the National Rail Company of Luxembourg, the study trained various models, including LIGHTGBM, Random Forest, KNN, and Linear Regression, assessing their performance using RMSE, MAPE, MAE, and R2 metrics. Among them, LIGHTGBM achieved the highest R^2 score, indicating superior predictive accuracy. To improve model interpretability, the SHAPLEY technique was employed. Additionally, the

project explored the XGBOOST algorithm, which further enhanced prediction performance, achieving an impressive R^2 score of 91%.

REFERENCES:

1. V. Cacchiani, A. Caprara, and P. Toth, ‘‘Scheduling extra freight trains on railway networks,’’ Transp. Res. B, Methodol., vol. 44, no. 2, pp. 215–231, Feb. 2010, doi: 10.1016/j.trb.2009.07.007.
2. P. Mcmahon, T. Zhang, and R. Dwight, ‘‘Requirements for big data adoption for railway asset management,’’ IEEE Access, vol. 8, pp. 15543–15564, 2020, doi: 10.1109/ACCESS.2020.2967436.
3. Q. Li, J.-C. Sibel, M. Berbineau, I. Dayoub, F. Gallee, and H. Bonneville, ‘‘Physical layer enhancement for next-generation railway communication systems,’’ IEEE Access, vol. 10, pp. 83152–83175, 2022, doi: 10.1109/ACCESS.2022.3192971
4. N. Zhao, C. Roberts, S. Hillmansen, and G. Nicholson, ‘‘A multiple train trajectory optimization to minimize energy consumption and delay,’’ IEEE Trans. Intell. Transp. Syst., vol. 16, no. 5, pp. 2363–2372, Oct. 2015, doi: 10.1109/TITS.2014.2388356.
5. M. S. Artan and I. Sahin, ‘‘Exploring patterns of train delay evolution and timetable robustness,’’ IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 11205–11214, Aug. 2022, doi: 10.1109/TITS.2021.3101530.
6. F. Bigi, T. Bosi, J. Pineda-Jaramillo, F. Viti, and A. D’Ariano. (2023). Addressing the Impact of Maintenance in Shunting Operations Through Shunt-In Policies for Freight Trains Operations. [Online]. Available: http://hdl.handle.net/10993/53485
7. N. Bešinović, R. M. P. Goverde, E. Quaglietta, and R. Roberti, ‘‘An integrated micro–macro approach to robust railway timetabling,’’ Transp. Res. B, Methodol., vol. 87, pp. 14–32, May 2016, doi: 10.1016/j.trb.2016.02.004.
8. 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, doi: 10.1109/TITS.2014.2358392.
9. P. Bhavsar, I. Safro, N. Bouaynaya, R. Polikar, and D. Dera, ‘‘Machine learning in transportation data analytics,’’ in Data Analytics for Intelligent Transportation Systems, M. Chowdhury, A. Apon, and K. Dey, Eds. Amsterdam, The Netherlands: Elsevier, 2017, pp. 283–307, doi: 10.1016/B978- 0-12-809715-1.00012-2.
10. J. D. Pineda-Jaramillo, R. Insa, and P. Martínez, ‘‘Modeling the energy consumption of trains by applying neural networks,’’ Proc. Inst. Mech. Engineers, F, J. Rail Rapid Transit, vol. 232, no. 3, pp. 816–823, Mar. 2018, doi: 10.1177/0954409717694522.
11. S. Milinković, M. Marković, S. Vesković, M. Ivić, and N. Pavlović, ‘‘A fuzzy Petri net model to estimate train delays,’’ Simul. Model.Pract. Theory, vol. 33, pp. 144–157, Apr. 2013, doi: 10.1016/j.simpat.2012.12.005.
12. V. De Martinis and F. Corman, ‘‘Data-driven perspectives for energy efficient operations in railway systems: Current practices and future opportunities,’’ Transp. Res. C, Emerg. Technol., vol. 95, pp. 679–697, Oct. 2018, doi: 10.1016/j.trc.2018.08.008.
13. J. Pineda-Jaramillo, P. Martínez-Fernández, I. Villalba-Sanchis, P. Salvador-Zuriaga, and R. Insa-Franco, ‘‘Predicting the traction power of metropolitan railway lines using different machine learning models,’’ Int. J. Rail Transp., vol. 9, no. 5, pp. 461–478, Sep. 2021, doi:

10.1080/23248378.2020.1829513.
14. H. Alawad, S. Kaewunruen, and M. An, ‘‘A deep learning approach towards railway safety risk assessment,’’ IEEE Access, vol. 8, pp. 102811–102832, 2020, doi: 10.1109/ACCESS.2020.2997946.
15. P. Kecman and R. M. P. Goverde, ‘‘Predictive modelling of running and dwell times in railway traffic,’’ Public Transp., vol. 7, no. 3, pp. 295–319, Dec. 2015, doi: 10.1007/s12469-015-0106-7.
16. D. Li, W. Daamen, and R. M. P. Goverde, ‘‘Estimation of train dwell time at short stops based on track occupation event data: A study at a Dutch railway station,’’ J. Adv. Transp., vol. 50, no. 5, pp. 877–896, Aug. 2016, doi: 10.1002/atr.1380.
17. P. Huang, C. Wen, Q. Peng, C. Jiang, Y. Yang, and Z. Fu, ‘‘Modeling the influence of disturbances in high-speed railway systems,’’ J. Adv. Transp., vol. 2019, pp. 1–13, Mar. 2019, doi: 10.1155/2019/8639589.