Enhancing Freight Rail Arrival Time Predictions with Light GBM and XG Boost: A Comparative Analysis of Algorithms and Feature Importance Using Data Driven Models

Ms.Valapala Prabhavathi [1], Aare Samatha [2], Boorgula Manusha [3], D Laxmi [4]

[1] Assistant Professor, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad

[2], [3], [4] Student, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad

ABSTRACT: It is still very difficult to maximize the use of rail infrastructure in every way, even if rail is becoming more and more popular as a freight transportation mode owing to its position in multimodal transportation and many environmental and economic advantages. Several approaches to train interruption prediction model development have been used to tackle this problem. Still, these models don't do a very good job of estimating how long it will be until an arrival, how long it will take to determine what caused the delay, and how it will affect operations. Operators are unable to adequately lessen the impact of interruptions due to a lack of information at their disposal. This study aims to examine a collection of data-driven models that can predict arrival delays in the short term using data from transport train operations among Battenburg (Luxembourg) and nine other terminal stations in the European Union (Caccini et al., 2010; McMahon et al., 2020). Then, it will investigate the effects of the features linked to arrival delays. It seems that departing delay a period of time trip the distance, and train makeup are the most important characteristics for short-term arrival delay time prediction in freight rail activities, with the bright GBM model outperforming other models for our dataset. The National Railway Corporation of Luxembourg may use the decision-support system that this research developed—a short-term prediction model. One example is the ability to forecast future operating time based on a train's arrival delay time. This web service provides further help in reducing interruptions and operational delays. Search Terms: Data-driven models, freight transport, delays forecasting, gradient boosting, and delays in rail operations.

INTRODUCTION:

As a result of its many benefits in terms of operating costs, efficiency, dependability,

emissions, and safety, rail transportation is rapidly gaining popularity in the freight transportation business (Zhao et al., 2015; Bigi et al., 2023). As a consequence of this tendency, governmental agencies are pushing for a change away from other options, and rail is becoming further integrated into multimodal transportation. It is crucial to optimize all parts of the rail infrastructure since rail intermodal operations are becoming increasingly vital for the reliability and efficiency of the freight transport business. Part of this is making sure the infrastructure can handle the needs of the transportation system as it is, which means keeping it in good repair and under good management. In addition, the rail transport industry as a whole must optimize its infrastructure and performance via the use of data analytics and technology. Railway delays are a major problem that needs fixing since rail networks are complicated and have a lot of rolling equipment working on them. The first kind of delay is caused by changes in the train's performance while traveling; the second kind is caused by the uncertain period it takes to get the train ready to go. Rail risk management relies heavily on arrival delay forecasting, which is the process of determining when a train will arrive at two stations relative to their planned arrival times. Train dispatchers have the important task of analyzing how interruptions will affect the whole schedule and making adjustments to operations in order to minimize losses. This is done to prevent a domino effect of delays that might affect the overall system functioning. To predict interruptions and the consequent delays in railway operations, event-based models are often used. These models include processes with departure, transit, and arrival events. When it comes to dealing with and identifying connections in time-based, multidimensional, nonlinear data, data-driven models have shown some potential. Previous research has used these models to predict disturbances in rail operations and their resulting delays, and they have proved effective in revealing the interrelationships between different characteristics of rail operations. On the other hand, these models haven't been able to nail down not just the projected effect on operations but also the fundamental causes of short-term arrival delays and their timings.

RELATED WORK

Rail network scheduling of additional freight trains

Train networks that carry both passengers and freight are the focus of our research

(Corman & Meng, the year 2015). The schedules of passenger trains cannot be amended, but the infrastructure manager might receive requests to insert additional freight trains from the train operators. An ideal schedule is specified by the corresponding train operator for each freight train; however, this schedule may be changed by the infrastructure management as long as safety operating requirements are satisfied. Specifically, this change can cause the train to follow a different course than the one specified in the optimal schedule. To put it simply, the goal is to increase the number of freight trains by giving them schedules that are almost perfect**.**

An Optimization of Multiple Train Trajectories for Delay and Energy Saving

In railroad operations, the time interval between trains may drop under the required line headway distance if the previous train's travel is interrupted. This might lead to train encounters, which in turn could cause more energy consumption, knock-on delays, and fines for the operators. Taking into account the trade-off between reducing train energy use and increasing the delay penalty in a delay scenario with a fixed blocks signalling system, this article offers an optimization analysis of a train path (driving speed curve). Recalculating the behavior of the second and following trains based on the efficiency of all cars in the network, save the leading train, allows us to evaluate the interactions between the trains. The research required the creation of a multigrain simulator. In order to discover the best results as fast as possible, we use three search techniques: genetic algorithm, ant colony optimizing, and improved brute force. The outcome demonstrates that trains may improve their efficiency and decrease energy consumption by using optimum trajectories or driving styles, which in turn limit interactions between them. Both security and passenger comfort are enhanced as a result of this.

Concepts, Technologies, and Challenges in Security at the Physical Layer for Future Wireless Networks

To ensure the privacy and authenticity of communications, security at the physical layer (PHY-security) makes use of the inherent randomization of transmission channels. We anticipate that this new safety system will rely heavily on wiretap coding and processing of signals technology. Due to its distinctive characteristics and the fact that wireless communications are essential to our everyday lives for the transfer of sensitive and private information, PHY-security has

garnered considerable interest. In contrast to traditional cryptography, which considers the execution of security protocols while performing security functions, PHY-security solutions focus only on ensuring that all entities involved load authenticated cryptographic information. Put another way, it doesn't need the use of any additional security measures or algorithms on levels beyond the physical one. This survey presents the core ideas of physical layer security (PHY-security), including authentication and confidentiality, and gives a rundown of the latest research on PHY-security methods that can secure wireless communications, including discussions of problems and possible solutions. Additionally, we outline the outstanding problems that will guide our future study at the conclusion of the publication.

ALGORITHM:

Light GBM Algorithm

According to Bacanovic et al. (2016), Light GBM is a distributed, high-performance, open-source gradient boosting framework that was created by Microsoft. Its design prioritizes accuracy, scalability, and efficiency. It relies on decision trees that are made to make models more efficient and use less memory. Among its innovative features is Gradient-based One-Side Sample (GOSS), a method for optimizing memory utilization and training time by preferentially retaining instances with significant gradients. Light GBM also uses techniques that are based on histograms to build trees efficiently. The efficiency of Light GBM is enhanced by these approaches and improvements like as leaf-wise tree development and effective data storage formats, which set it apart from other gradient boost frameworks.

objective: Chooses the loss function that will be optimized when training. Light GBM is compatible with a wide range of goals, including multiclass classification, binary classification, and regression.

task: Whether we want to train or make predictions, that's what it says. "Train" is the default value. Prediction is an additional value that might be assigned to this option.

Num leaves: Sets the upper limit for how many leaves a tree may have. While larger values make the model more capable of capturing intricate patterns, they also increase the risk of overfitting.

learning rate: This parameter controls the step size used in gradient descent. Slower

learning and maybe better generalization are the outcomes of lower values.

modest: Determines the highest possible tree depth. More complex interactions can be captured by the model with higher values, although overfitting is a risk.

Min data in leaf: The smallest possible set of data points needed to create a leaf node is specified. Increasing the value helps avoid underfitting, but it also increases the risk of overfitting.

Num iterations: It defines the maximum number of iterations that must be executed. A number of 100 is used by default.

feature fraction: Sets the percentage of characteristics used to construct each tree. To decrease overfitting and increase model variety, randomly picking certain characteristics is useful.

bagging fraction: Determines what percentage of the data will be utilized for bagging (replacing data points in a sample) when training. The model's robustness and variance may be enhanced with its support.

lambda_l1 and lambda_l2: Default values for L1 and L2 regularization parameters, correspondingly. To avoid overfitting, they punish big coefficients.

Min split gain: Determines the bare minimum gain needed to further partition a node. As a result, the tree's development may be more carefully managed, and splits won't be as common.

Categorical feature: It specifies the categorical feature used for training model.

Linear Regression:

Many issues, even those that are inherently non-linear, can be simply handled using linear models, making them the simplest parametric approaches that always merit the correct attention. Since regression is a kind of continuous target prediction with several potential uses, it's useful to be familiar with the inner workings of a linear model, including how it fits the data, its advantages and disadvantages, and when to choose an alternative. An intriguing approach to effectively working with non-linear information utilizing the same models will be covered in the latter section of the chapter. As a supervised machine learning approach, linear regression fits observed data to a linear equation to determine the dependent variable's linear relationship with one or more independent characteristics. Simple Linear Regression is used when there is a single independent variable, while several Linear

Regression is used when there are several independent variables. Just as Multivariate Regression accounts for several dependent variables, Univariate Linear Regression does the same for a single dependent variable.

Types of Linear Regression

There are two main types of linear regression:

Simple Linear Regression

Since there is merely one dependent variable along with an independent variable in this version of linear regression, it is the most basic kind. A basic linear regression equation looks like this:

Here, Y is the variable that is dependent and X is the variable that is independent, and the equation

$y=\beta 0+\beta 1X$y=β0+β1X reads as follows: ϋ0 represents the intercept, whereas ω1 stands for the slope.

Multiple Linear Regression

There is more than one set of variables at play here, both independent and dependent. When using multiple linear regression, the equation becomes:

This is the equation: $y=\beta 0+\beta 1X+\beta 2X+$…….The equation y=β0+β1X+β2X+…子… can be paraphrased asIn cases where:

• The dependent variable is Y.

The independent variables are X1, X2,..., Up.

There are n slopes, denoted as β1, β2,..., βn, and the intercept is β0.

K-Nearest Neighbours (KNN):

The **K-Nearest Neighbors (KNN) algorithm** is a a supervised machine learning approach used to address regression and classification issues. Originally created in 1951 by Edith Fix and Joseph Hodges, Thomas Cover later refined and improved upon this method. Learn everything about the KNN algorithm—its inner workings, its implementation, and more—in this comprehensive essay.

Could you please explain the K-Nearest Neighbors Algorithm?

When it comes to machine learning classification algorithms, KNN is among the most fundamental and fundamentally important. Many fields rely on it, including recognition of patterns, mining information, and intrusion detection, all of which fall within the supervised learning area.

Distance Metrics Used in KNN Algorithm

The KNN method is well-known for its ability to locate nearby points or groups in relation to a query point. However, a measure is required in order to ascertain the closest clusters or points to a query point. Here are some distance measurements that we utilize for this purpose:
Proximity to the Equator
For two locations in the plane or hyperplane, this is just the distance in cartesian coordinates. You may alternatively think of the width of the straight path connecting the two places as the Euclidean distance. Using this measure, we can determine the total amount of movement that an item has undergone between its two states.

Manhattan Radius
When we are more concerned with the object's total distance traveled rather than its displacement, its Manhattan Length metric is often used. The total disparity between each point's n-dimensional coordinates is added together to get this metric.
The Minkowski Distance
The Minkowski distance is a specific case of the Euclidean and the Manhattan distances, as we might state.

By extrapolating from the above formula, we see that the formula for the distance from Manhattan is obtained when p = 1, and that it is identical to the equation for the Euclidean distance when p = 2.
When solving a Machine Learning problem, the most popular metrics are those we've already covered. However, other distance metrics, such as Hamming Distance, can be useful when comparing two vectors that contain both Boolean and string values.

Random Forest:

One effective machine learning tree learning approach is the Random Forest algorithm. In order for it to function, many Decision Trees are constructed during the training process. A different portion of the dataset is used to measure a different collection of characteristics in each partition when each tree is created. By introducing variation among individual trees, this randomization improves prediction performance and decreases the danger of overfitting. In forecasting, the algorithm takes into account the output of all trees and averages them (in regression tasks) or votes on them (in classification tasks). A good example of consistent and accurate findings is this collaborative decision-making method that uses the insights of several trees. Because of their robustness against overfitting, flexibility in dealing with complicated data,

and capacity to provide accurate predictions in a variety of contexts, random forests find extensive use in regression and classification tasks.

Key Features of Random Forest

Some of the Key Features of Random Forest are discussed below–>

High Predictive Accuracy: Picture random forest learning as a group of expert decision-makers. Together, the insights of each wizards (decision tree) into a robust prediction tapestry are woven. When wizards work together, they can frequently create more precise models than any one of them could on their own.

Resistance to Overfitting Decision trees are the mentees of Random Forest, which acts as a level-headed teacher. It promotes a more holistic understanding rather than allowing each trainee to commit every element of their instruction to memory. This method makes the model less likely to overfit by avoiding becoming too entangled with the training data**.**

Large Datasets Handling: Faced with an overwhelming amount of data? Using its decision trees as a crew, Random Forest takes on the problem like an experienced explorer. With everyone chipping in, the expedition is able to cover a lot of ground in a surprising amount of time.

Variable Importance Assessment: Like a police investigator sifting through evidence, Random Forest prioritizes characteristics based on their significance. It prioritizes the most important factors that influence predictions by evaluating the significance of every clue to solving the case.

Built-in Cross-Validation random forest learning is like getting a coach at your side, making sure you stay on track. It uses a hidden set of cases (out-of-bag) to evaluate while it educates each decision tree. Your model will not only succeed during training, but it will also thrive when faced with novel problems, thanks to its built-in validation.

Handling Missing Values: Similar to datasets that include missing values, life is filled with uncertainty. Random Forest is like a reliable companion that adjusts to new circumstances by generating predictions based on the data at hand. It remains composed in the face of incomplete information and concentrates on providing us with the information it is certain of.

Parallelization for Speed: Random Forest may help you save time. To further understand this concept, think of every tree of decisions as an employee working on a

separate puzzle piece. To better manage large-scale projects, this parallel technique makes use of current technology, which speeds up the whole process.

XGBOOST Algorithm:

Machine learning model training has never been easier than with XG Boost, an enhanced distributed gradient boosting library. In order to make a more robust prediction, it integrates the results of several less robust models, a process known as ensemble learning. "XG Boost" stands for "Extreme Gradient Boosting," and it is one of the most famous and extensively used ML algorithms because of how well it performs on big datasets and how many ML tasks, like regression and classification, it can handle.
Since XG Boost efficiently handles missing values, it is capable of handling real-world data with values that are missing without needing considerable pre-processing, which is an important advantage. Plus, XG Boost comes with its own built-in parallel processing capabilities, so you can train models on massive datasets quickly.
Many different kinds of applications may make use of XG Boost. Some examples include recommendation systems, click-through rate prediction, and Kaggle contests. It also lets you tweak different model settings to get the most performance out of it, and it's quite configurable.
XG Boost is an abbreviation for extreme gradient boost, a method developed by UW researchers. An optimization library for Gradient Boosting training, built in C++. XG Boost is a class of decision trees that use Gradient Boosting. XG Boost models are the clear winners of several Kaggle tournaments.

METHODOLOGY: To implement this project, we have designed following modules

Upload Attack Dataset: By using this module, the dataset may be uploaded to the program, which will then read the dataset.

Preprocess & split Dataset: In this module, we will implement processing methods such as data shuffling, normalization, and cleaning. We will then divide the dataset into two halves, one for training and the other for testing, with a ratio of 80:20.

Run Light GBM Algorithm: To train a model, we will feed the Light GBM method 80% of the training dataset. Then, we will apply the trained model to 20% of the test data in order to determine the prediction accuracy.

Run Linear Regression Algorithm: The Linear Regression method will be fed an 80% training dataset in order to train a model, and

then the model will be tested on a 20% test dataset in order to determine the accuracy of its predictions.

Run KNN Algorithm In order to train a model, we will feed the KNN algorithm 80% of the training dataset. Then, we will apply the trained model to 20% of the test data in order to determine the prediction accuracy.

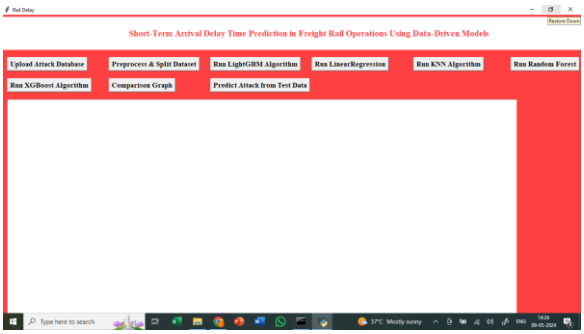
Run Random Forest Algorithm: The Random Forest technique will be used to train a model using 80% of the training dataset, and then it will be applied to 20% of the test data to determine the prediction accuracy.

Run XG Boost Algorithm: We will train a model using 80% of the training dataset and then apply it to 20% of the test data to determine the prediction accuracy using the XG Boost technique.

Comparison Graph: The use of this module entails evaluating each algorithm's efficiency. In this graph, the x-axis shows the name of the algorithm, while the y-axis shows its accuracy.

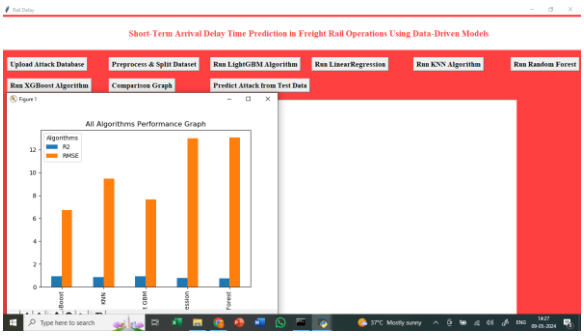
Predict Attack from Test data: Forecasts on test data are made using the trained models in this last lesson. Additional analysis or decision-making may be informed by the findings.

RESULT AND DISCUSSION



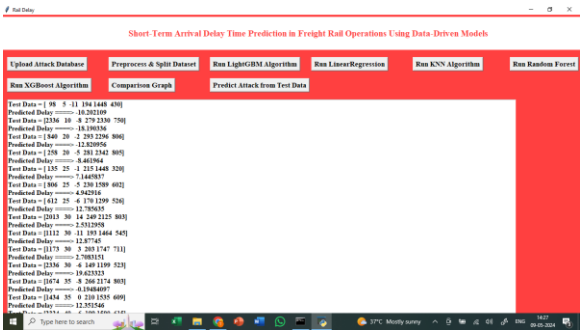
In the above result we got Tk inter Output Window.

In above result training Random Forest algorithm and its R2 values is 81%.



In above graph 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.



In above result we are reading test data from test file and then predicting delay using XGBOOST extension object and in output we can see test data in square bracket and after arrow =➔ symbol we can see predicted

CONCLUSION

A reliable approach to forecasting freight train arrival delays and comprehending their root causes and consequences is presented in this paper. The goal is to improve railway dependability and reduce interruptions via the development of an actual time decision-support system (STDSS). This study used five regression models and indicators such as R^2, RMSE, MAPE, and MAE to examine short-term freight train delays, as opposed to the majority of prior research that has concentrated on train passengers or network delays. Most effectively, the Light GBM models demonstrated how departing delays, journey the distance, and railway composition substantially impact arrival delays. The results indicate that if train routes were optimized, operating and arrival delays might be significantly reduced. Incorporating data from many organizations and perhaps other aspects, such as weather, into future study is essential. Zhao et al. (2015) and Caccini et al. (2010) found that real-time delay predictions might be even better with computerized data analysis and model implementation.

REFERENCES

[1] V. Caccini, A. Caprara, and P. Toth, ‘‘Scheduling extra freight trains on railway networks,’’ Transp. Res. B, Methodios., 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. Perrineau, I. Dayoub, F. Gallie, 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. Hillmans, and G. Nicholson, ‘‘A multiple train trajectory optimization to minimize energy consumption and delay,’’ IEEE Trans. Inntel. 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. Inntel. 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. Darian. (2023). Addressing the Impact of Maintenance in Shunting Operations Through Shunt-In Policies for Freight Trains Operations. [Online].

[7] N. Bacanovic, R. M. P. Govere, E. Quaglietta, and R. Roberti, ‘‘An integrated micro–macro approach to robust railway timetabling,’’ Transp. Res. B, Methodology., 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. Intel. Transp. Syst., vol. 16, no. 3, pp. 1274–1284, Jun. 2015, Doi: 10.1109/TITS.2014.2358392.