Original Article

Institution of MECHANICAL ENGINEERS

Journal of RAIL AND RAPID TRANSIT

Proc IMechE Part F:
J Rail and Rapid Transit
2024, Vol. 238(10) 1338–1350

DOI: 10.1177/09544097241280848
journals.sagepub.com/home/pif
Sage

Estimating the occurrence of broken rails in commuter railroads with machine learning algorithms

Di Kang[1], Junyan Dai[1], Xiang Liu[1], Zheyong Bian[2], Asim Zaman[1] and Xin Wang[1]

Abstract

Broken rail prevention is critical for ensuring track infrastructure safety. With the increasing availability of rail data, the opportunity for data-driven analyses emerges as a promising avenue for enhancing railroad safety. While previous research has predominantly concentrated on predicting broken rails within the context of freight railroads, the attention afforded to commuter railroads has been limited. To address this research gap, this paper presents an analytical modeling framework based on machine learning (ML) algorithms (including LightGBM, XGBoost, Random Forests, and Logistic Regression) to investigate the occurrence of broken rails on commuter rail segments. It leverages various features such as gradient, curvature, annual traffic, operational speed, and the history of prior rail defects. We use oversampling techniques, including ADASYN, random oversampling, and SMOTE, to address the issue of imbalanced data. This challenge arises due to the majority of commuter rail segments not experiencing any broken rails during the study period, resulting in a small sample size of broken rail instances. The findings indicate that, for the dataset employed in this study, LightGBM, in conjunction with random oversampling, exhibits superior performance. Based on the feature importance results, the critical factors influencing the prediction of broken rail occurrences on this commuter railroad are gradient, operational speed, and prior rail defects.

Keywords

Broken rails, machine learning, commuter railroad, rail defects

Date received: 20 February 2023; accepted: 13 August 2024

Introduction

Train derailments frequently occur due to broken rails.[1–5] Based on historical derailment records,[1] broken rails stand out as a primary contributor to train derailments across various types of trains, including unit trains and manifest trains. These records indicate that broken rails constituted 17.9 percent of the mainline derailment causes on Class I railroads from 1996 to 2018. A rail segment is classified as "broken" when it exhibits one or more gaps. The emergence of broken rails is a result of repetitive traffic loading over time, especially on heavily trafficked mainlines. The development of a broken rail is attributed to the presence of minor internal defects, referred to as "prior rail defects," which gradually evolve into significant fractures within the rail. If a prior rail defect exceeds a predefined threshold, rendering the track unsafe for any traffic, it is categorized as a service failure.

Currently, the majority of railroad companies conduct routine rail inspections and track the occurrence of prior rail defects through automated ultrasonic inspections. Subsequently, a walk-through process is implemented, wherein an inspector traverses the entire railroad system to validate the accuracy of broken rail detection by sensors. While sensor-based inspection techniques have gained prominence in recent years,[6–10] the majority of mandatory inspections aimed at identifying broken rails still rely on walk-through inspections.[11] These walk-through inspections, however, are recognized for being both time-consuming and costly.

While there have been prior studies on commuter railroads that concentrate on track geometry exceptions, they have not specifically addressed the potential locations of broken rails.[12] The majority of research regarding the prediction of broken rails has been conducted within the context of freight railroads.

Freight railroads often transport heavy loads, which can exert significant stress on the tracks over time. This can lead to wear and tear, increasing the risk of broken rails. Freight trains also tend to have longer and heavier cars, potentially amplifying the impact of a broken rail. Previous studies found that the occurrence of broken rails on freight railroad might be affected by traffic loadings (e.g., monthly or yearly gross tonnage or number of trains), track characteristics (the steel type, rail profile, and geometry), inspection information (depth of surface crack, the number of prior rail defects), and maintenance records (grinding and ballast

[1]Department of Civil and Environmental Engineering, Rutgers University, New Brunswick, NJ, USA
[2]Department of Supply Chain and Logistics Technology, University of Houston, Houston, TX, USA

Corresponding author:
Xiang Liu, Department of Civil and Environmental Engineering, Rutgers University New Brunswick, CoRE 606, 96 Frelinghuysen Road, Piscataway, NJ 08854-8018, USA.
Email: xiang.liu@rutgers.edu

cleaning).[13–15] In contrast, commuter railroads primarily carry passengers, and while their trains are generally lighter than freight trains, the frequency of train movements can still contribute to track degradation. Commuter trains often engage in more frequent stops and starts, imparting a distinctive impact on the tracks when contrasted with the relatively consistent pace of freight trains. Furthermore, in contrast to freight railroads, commuter railroads adhere to more stringent safety conditions, involving frequent maintenance and replacement of components. These conditions also have an impact on the occurrence of broken rails. While previous research has extensively examined the prediction of broken rails in the context of freight railroads, the applicability of existing research from the freight railroad context to the commuter railroad context may be constrained by the differences between the two types of rail operations. The objective of this paper is to create an analytical modeling framework based on machine learning for the estimation of the likelihood of broken rail occurrences within the context of commuter railroads. By conducting a comprehensive comparative analysis of several machine learning algorithms, this study aims to pinpoint the optimal prediction framework customized to the dataset obtained from a commuter railroad. Using the given data, this paper seeks to identify the most influential factors contributing to the occurrence of broken rails based on the results of feature importance analysis.

This paper presents an ML-based framework for predicting broken rail occurrences on commuter railroads, integrating track design characteristics, operational features, inspection data, and prior rail defect records. Due to limited commuter railroad-specific broken rail records, accurately assessing the likelihood of occurrences is challenging. To address data imbalance issues, three resampling techniques (ADASYN, random oversampling, and SMOTE) are employed. ML algorithms include LightGBM, XGBoost, Random Forests, and Logistic Regression, with a comparative analysis determining the optimal framework. Based on feature importance results, this paper identifies the most influential factors in estimating the probability of broken rails for commuter railroads.

The remainder of this paper is organized as follows: the literature review section summarizes previous work related to broken rail prediction. The methodology section delineates the machine learning-based modeling framework and elucidates the performance metrics used to assess the outputs. The experimental results section implements the proposed modeling framework with the data from a commuter railroad. This is followed by the paper's discussion and conclusions.

Literature review

Various types of rail failures and damage can occur on railway tracks, and Machine Learning algorithms are widely adopted for monitoring such issues.[16] Rail corrugation, also referred to as "rail surface defect", is an example type of rail damage. This type of damage is primarily caused by the interaction between train wheels and the rail, leading to a zigzag-shaped appearance. Rail corrugation can generate high-frequency vibrations, causing discomfort for passengers on commuter trains and exacerbating defects between vehicles and tracks. Kaewunruen et al.[17] conducted a study where they applied the Artificial Neural Network. They used acceleration data from D-track simulations as input and the size of corrugation as the output variable to analyze and predict rail corrugation. Moreover, track geometry irregularities represent another category of rail damage that has undergone extensive study. These irregularities are crucial as track systems form the foundation of the rail and directly bear dynamic loads. In a study by Li et al.,[18] the unsupervised learning mechanism of an autoencoder was employed to estimate track longitudinal irregularity, with in-service train acceleration data as input. Damage to rail fasteners, frequently leading to heightened vibration and track misalignment, has also been the subject of significant research attention. Chen et al.[19] devised an innovative approach using a fully convolutional network (FCN) to detect imperceptible fastener damage. Their method leveraged data such as axle box accelerations, track irregularity, and vehicle speed to identify these subtle but impactful issues. In the context of ballasted railway tracks, Ngamkhanong and Kaewunruen[20] employed Artificial Neural Networks to forecast buckling failures resulting from extreme temperature, with a range of ballasted track conditions serving as the input features.

Among the various types of rail damage and failures, broken rails are considered one of the most critical.[21] This type of failure leads to the rail splitting into two separate pieces. Previous research[1,4] has highlighted that broken rails are a primary cause of train derailments across different types of trains. Factors affecting the occurrence of broken rails include rail characteristics (e.g., rail age, rail weight, and rail quality), track layout (grade, turnout, track, and curvature), track maintenance, operational information (speed and traffic loadings), and defect inspection history (track geometry exceptions, service failures, and prior rail defects).[13] Previous studies to predict broken rails on freight railroads are summarized by the data resources, input variables to predict broken rails, and models used (Table 1). Although the prediction of broken rails on freight railroads has been well studied in previous work, the correlation between influencing factors and broken rails on commuter railroads is still considerably understudied. The main differences between this paper and previous studies include: (1) this paper leverages data from a commuter rail agency, marking the pioneering effort in analyzing and estimating the probability of broken rail occurrences in this context. The input variables that might affect the formation of broken rails are utilized in this paper, including track design characteristics (gradient and curvature), operational features (annual traffic and operational speed), as well as inspection data (historical records of prior rail defects and occurrences of broken rails). (2) It constructs an ML-based modeling framework that incorporates multiple oversampling techniques to resample and accommodate the unbalanced dataset.

Previous studies on the commuter railroads focused more on economic policy[22–24] and operation evolutions.[25–27] In pursuit of enhanced commuter railway safety and the reduction of human labor in post-disaster recovery, researchers

Table 1. Summary and comparison of previous work on predicting the occurrence of broken rails on freight railroads.

Authors	Data description	Features used in the model	Models used
Dick et al.[31]	3,676 records with complete service failure and descriptive parameter information over 2 years.	Service failures, track and traffic data, rail age, rail weight, degree of curve, speed, average tons per car, average dynamic tons per car, percent grade, annual gross tonnage, annual wheel passes, insulated joints, and mainline turnouts.	Multivariate statistical model
Schafer and Barkan[32]	24,000 route miles of mainline trackage for a major North American railroad covering the 4-year period, 2003–2006	Rail characteristics, infrastructure data, maintenance activity, operational information, and track testing results.	Logistic regression, artificial neural network (ANN), and the hybrid ANN/logistical regression
Wang et al.[33]	Class I freight railroad data covers 20,000 miles for 2013–2019	Infrastructure data, operation data, inspection related data, climatological data	Logistic regression, Random Forests, XGBoost
Zhang et al. [14]	Class I freight railroad data covers 20,000 miles for 2013–2016.	Broken rail data, prior rail defect data, track geometry exception data, exception, tonnage data, grinding data, ballast cleaning data, track type data, geometry data, signal data, GIS data.	Linear regression, Logistic Regression model, XGBoost, cox proportional hazard regression
Ghofrani et al.[34]	21,000 miles from a Class-I railway during the period from 2011 to 2016.	Tonnage, ultrasonic rail inspection schedules, prior rail defects, service failures, geometry defects, grinding schedules, rail age, curvature, grade and turnouts data	Accelerated failure time model
Ghofrani et al.[35]	Class I U.S. freight railroads covering 93.2 miles for 6 years from 2011 to 2016	Track prefix, track type, date of the previous and the following inspections, interval between two inspections, tonnage, number of prior rail defects, grinding, grade, curvature, age of rail, number of turnouts, and temperature.	Logistic regression, decision tree, multilayer perceptron, and gradient boosting classifier

have explored innovative approaches. To alleviate labor-intensive inspection tasks, Onodera et al.[28] proposed a sensor-based method to detect seismic damage to bearings. Rungskunroch et al.[29] used decision tree (DT) and Petri-nets (PT) models to design a posterior probability model to benchmark balanced safety performance among railway networks. However, the only related paper on commuter railroads estimating the occurrence of broken rails was conducted in 2010 using linear regression,[30] which lacked the sophistication required for accurate estimation.

Methodology

Figure 1 details the ML-based analytical modeling framework to predict the probability of broken rail occurrences by utilizing track design characteristics, operational features, and inspection data.

Data preparation and machine learning algorithms

Prior to feeding the data into the machine learning algorithms, the railroad is initially condensed into rail segments, with each of these segments containing track design attributes, operational characteristics, and inspection data. The commuter railroad under investigation uses jointed rail, with each rail unit measuring 39 feet in length. Consequently, the approach we employed in this paper involves dividing the examined railroad network into distinct segments, each of which spans precisely 39 feet in length. This segmentation strategy is implemented with the intention of streamlining the maintenance process, as it ensures that whenever a 39-foot rail segment experiences a break, the entire segment is replaced. This approach aligns closely with the real-world maintenance procedures in use.

This study employs two tree-based learners, namely the Light Gradient Boosting Machine (LightGBM) and eXtreme Gradient Boosting (XGBoost), to estimate the probability of broken rails on rail segments. Both LightGBM and XGBoost are renowned for their high-performance gradient boosting capabilities. XGBoost, in particular, utilizes the output from the previous tree as input for the next one, incrementally adding new trees to enhance the existing estimation. Before moving on to the next level, XGBoost grows trees level-wise. Since the XGBoost algorithm includes a regulation parameter when building the decision trees, it decreases the bias in fitting XGBoost to the training data set by reducing the amount that each single data point contributes to the new estimation. The primary objective of the XGBoost algorithm is to minimize both the loss function and the regularization term. The ultimate result is a weighted average of all the decision trees.

Regarding LightGBM, it, too, constructs an ensemble of decision trees in a sequential manner to make predictions. However, in contrast to XGBoost's level-wise tree building approach, LightGBM adopts a leaf-wise tree growth strategy. This means that LightGBM selects the leaf node that offers the maximum reduction in loss. Consequently, this strategy typically yields shallower trees and significantly enhances computational efficiency. For comparison purposes, this research paper incorporates two additional machine learning algorithms alongside the primary focus on LightGBM and XGBoost: Random Forests and Logistic

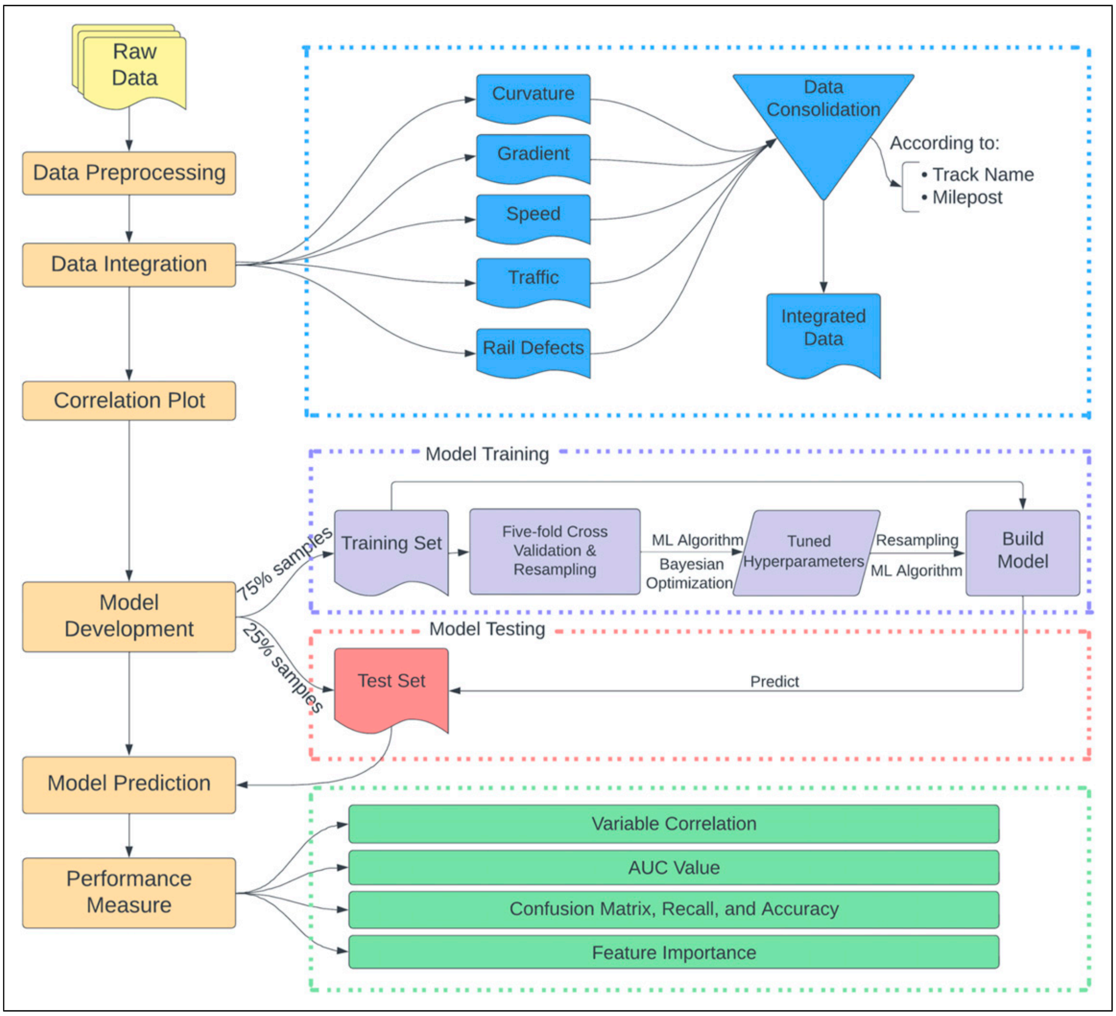

Figure 1. The ML-based analytical modeling framework to estimate the probability of broken rail occurrences.

Regression. These supplementary models are chosen to encompass a broader spectrum of machine learning algorithms.

Hyperparameter tuning

Hyperparameter tuning is a critical step in the application of tree-based machine learning algorithms. The complexity of a decision tree can enhance its capacity to accurately capture the nuances of the training data, thereby minimizing bias. Nonetheless, the improved complexity may result in overfitting, wherein the model exhibits diminished accuracy when confronted with unseen testing data. The process of hyperparameter tuning serves to achieve optimal model performance by balancing the delicate equilibrium between model complexity and simplicity. Thus, in this study, which encompasses the utilization of tree-based algorithms including Random Forest, XGBoost, and LightGBM, hyperparameter tuning is employed.

Training set oversampling

The volume of data collected for commuter railroad analysis is significantly smaller than that used in freight railroad analysis. While undersampling is a potential approach to mitigate the impact of this data imbalance, it reduces the amount of data available for model training, potentially resulting in lower model accuracy. Consequently, we implemented oversampling techniques to effectively address the imbalanced classification issue. Specifically, this study utilizes and compares three oversampling techniques: Adaptive Synthetic (ADASYN), Synthetic Minority Over Sampling Technique (SMOTE), and random oversampling. These techniques share the common objective of mitigating class imbalance by artificially augmenting the number of instances in the minority class. The distinctions among them arise from their manipulation of data using different rules and strategies to generate synthetic data. Random oversampling duplicates existing minority class samples randomly, while SMOTE generates synthetic samples by interpolating between neighboring data points.[36] ADASYN is an adaptive oversampling technique that balances the class distribution based on the classifier performance. For both ADASYN and SMOTE, we set the number of nearest neighbors (k) to $k = 5$. k is set to 5 because the authors of the paper introducing ADASYN[37] utilized $k = 5$ in their simulation analysis for both ADASYN and SMOTE.

As described in Figure 1, the ML-based analytical modeling framework incorporates a five-fold cross-validation approach. In the training sets, resampling techniques (including ADASYN, random oversampling, or SMOTE) are applied to tackle the challenge posed by imbalanced datasets. Additionally, Bayesian optimization is employed to systematically explore and pinpoint the optimal hyperparameter values for the tree-based machine learning algorithms. This comprehensive procedure is carried out within the context of a five-fold cross-validation setup, ensuring both robust model evaluation and effective hyperparameter tuning.

Performance metrics and model evaluation

The ROC (Receiver Operating Characteristics) curve is a commonly used visualization tool to show the performance of a binary classifier. Since the proposed ML algorithms estimate probabilities for each segment of having a broken rail, the ROC curve is applicable, and thus it is used as a performance measure. The ROC curve considers all possible thresholds to distinguish between true or false predictions based on the estimated probabilities, while each confusion matrix (as a traditional and straightforward metric) only pertains to a single threshold. The ROC curve shows the true positive rate (y-axis) versus the false positive rate (x-axis). The AUC (Area Under the Curve) is the area under the ROC curve. Ita calculates an aggregate value across all possible classification thresholds. It reflects the model performance of an ML algorithm by one value: the larger the AUC value, the better the model performance. In addition to the AUC value, various classification metrics, including the confusion matrix, accuracy, and recall, are employed to assess the performance of the proposed modeling framework.

Experimental results for broken rail estimation

To implement the ML-based modeling algorithm described in Figure 1, we have utilized various libraries, including scikit-learn version 1.3.0, xgboost version 1.7.6, lightgbm version 4.0.0, and imbalanced-learn version 0.11.0, all within the Python environment, specifically Python version 3.10.0.

Data description

The raw data contains rail information for 28 miles of commuter rail from a commuter rail agency. During data preprocessing, rail features are recorded and stored in separate Excel files foot by foot. All records are compared with the track chart and corrected accordingly. Records with missing data or apparent errors are discarded. The preprocessed data set covers approximately 28 miles of commuter rail tracks across seven segments. Then, six features are extracted from the preprocessed data and mapped into a new file based on the unique track names and milepost positions. Specifically, they are "curvature," "gradient," "traffic" (train schedule is assumed to be fixed over multiple years), "speed," "prior rail defect," and "broken rail." Table 2 provides basic information for each variable. Track Inspection Vehicles (TIV) have been conducting ultrasonic inspections for prior rail defects since 2020. Prior to this, Sperry ultrasonic testing was used for detection. However, the prior rail defect records from the years 2018 and 2020 are missing in the case study of the commuter rail agency. Whether a segment has broken rails, which is regarded as the output, is verified manually. Since broken rails rarely occur, we use all broken rail records from 2017 to 2021.

During the feature value extraction process, if at least one broken rail occurred over a 5-year period, the "broken rail" value for the 1-foot segment is set to one; otherwise, it is set to zero. The "prior rail defect" value for a 1-foot segment is computed as the sum of prior rail defects detected on that segment across all inspections.

Due to the infrequent occurrence of broken rails, the use of the original data resolution from the raw dataset leads to highly imbalanced data. Many one-foot segments do not experience broken rails, while only a few segments have reported broken rails. Consequently, the raw data is aggregated at a new data resolution, consolidating it into segments of uniform length measuring 39 feet, which corresponds to the physical length of a rail. When merging a 39-foot segment, this paper tries various aggregation methods to consolidate "gradient," "curvature," "speed," and "traffic" values. Aggregation options are considered such as computing the maximum, minimum, or average values. Given the computational efficiency of LightGBM, it is used as a benchmark to assess the effects of these diverse aggregation methods on model performance. Table 3 displays the AUC values for various combinations of aggregation methods and resampling techniques. Our analysis of the results reveals that the optimal aggregation method for generating feature values for each rail segment involves selecting the maximum value among all available values. To illustrate, when considering a feature like "gradient," if

Table 2. Data dictionary for input and output variables on 1-foot segment.

	Variable name	Date range	Variable description
Input variable	Curvature	2021	A negative or positive numerical value represents the curvature of a segment.
	Gradient	2021	A negative or positive numerical value represents the gradient of a segment.
	Speed	2021	A positive integer represents the operational speed of this segment.
	Traffic	2021	Cumulative counts denote the total number of trains that traversed this segment in 2021, with the assumption of constant yearly traffic.
	Prior rail defect	2016, 2017, 2019, and 2021	An integer value represents the cumulative count of prior rail defects detected across all inspections.
Output variable	Broken rail	2017–2021	A binary variable, taking the value 0 or 1, indicates the presence or absence of at least one broken rail occurrence within a segment during a 5-year period.

multiple gradient values appear on a 39-foot rail segment, we determine the feature value by taking the maximum among these values. Thus, in this paper, when merging a 39-foot segment, "gradient," "curvature," "speed," and "traffic" take the maximum values among 39 pieces of the 1-foot segment. According to feature interpretation, the "prior rail defect" value for the 39-foot segment signifies the accumulation of "prior rail defect" values from each of the 1-foot segments contained within that larger 39-foot segment.

After preprocessing the raw data (excluding missing data and errors), the dataset is narrowed down to 3756 segments, each measuring 39 feet in length. Historical records indicate that approximately 15 instances of broken rails occurred annually across the entire network. Over the past 5 years, 3682 of the 39-foot segments were devoid of broken rails (representing the majority class), while 74 segments recorded incidents of broken rails (forming the minority class).

Correlation among input variables

Prior to implementing the proposed ML-based modeling framework, a correlation plot offers an initial insight into

Table 3. AUC values for various aggregation methods using LightGBM.

Aggregation method	No resampling	Random Oversampling	SMOTE	ADASYN
Min	0.53	0.58	0.59	0.61
Avg	0.57	0.60	0.58	0.60
Max	0.64	**0.74**	0.68	0.70

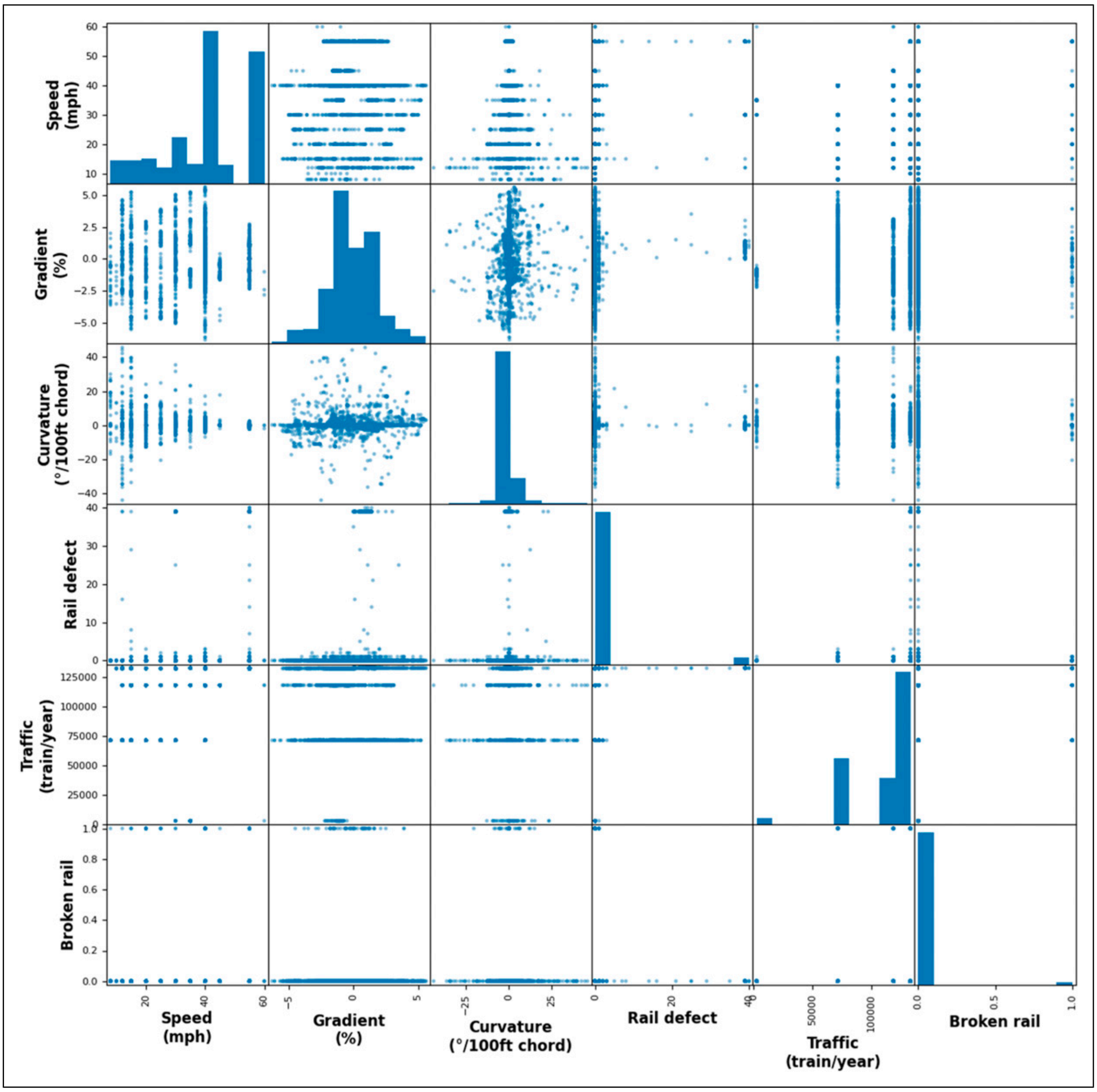

Figure 2. Correlation plot displaying all pairs of input features.

the relationships between each pair of features (Figure 2). The correlation plot reveals two significant insights: (1) When the segment has an exceptionally sharp curvature, the operational speed tends to be relatively low. As anticipated, this observation is consistent with the reasonable expectation that trains decrease their speed when navigating curves to ensure safety. (2) Broken rails are more commonly associated with moderate curvatures and gradients rather than sharp ones. Segments featuring substantial curvature and gradient might be expected to carry a higher probability of experiencing broken rails. However, akin to the practices of most commuter rail agencies, this particular commuter rail agency also follows a proactive approach to maintenance. Upon detecting a prior rail defect, they opt to replace the entire rail segment. This maintenance policy effectively curtails the progression of minor defects into full-blown broken rails.

As shown in Figure 2, there is no significant correlation observed among any pair of features. Consequently, all features are retained and included in the analytical modeling process.

Hyperparameter tuning for the tree-based algorithms

As shown in Figure 1, the preprocessed data is split into two parts: 75 percent is used as the training data set, and the remaining 25 percent is regarded as the testing data set. As mentioned in the methodology section, the hyperparameter tuning process is executed on the training dataset using a combination of five-fold cross-validation and the resampling technique. After optimizing the hyperparameters, the tuned values of each hyperparameter for different combinations of machine learning algorithms and resampling techniques are presented in Appendix A, Table A.1.

AUC values for various machine learning algorithms

Figure 3 displays the AUC values obtained after applying the proposed ML-based modeling framework to the testing data. We employed four different machine learning algorithms: LightGBM, XGBoost, Random Forest, and Logistic Regression, each in combination with various resampling techniques. The results are particularly noteworthy, highlighting that the use of the LightGBM algorithm in conjunction with the random oversampling technique achieves the highest AUC value. In contrast, among these four ML algorithms, the combination of Random Forest with the random oversampling technique yields the poorest model performance, falling even below the performance of random guessing. These findings underscore the significant improvement gained through the integration of LightGBM and random oversampling for estimating broken rails in commuter railroads. Consequently, we proceed to assess the effectiveness of this optimal strategy using additional performance measures in the subsequent subsection.

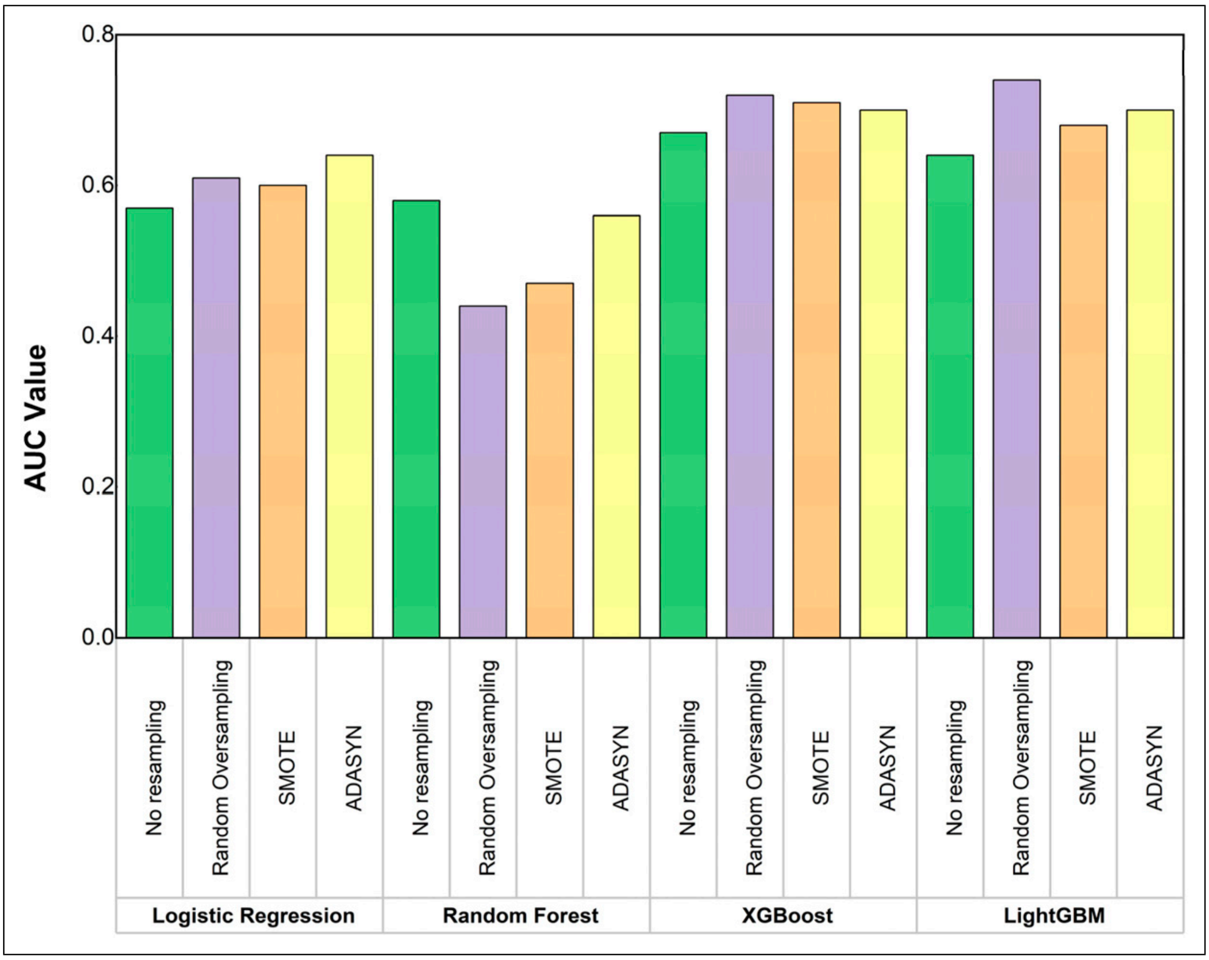

Figure 3. AUC values for various ML algorithms with various resampling techniques.

Estimation results for likelihood of broken rails using LightGBM with random oversampling

Segments are categorized as having a broken rail if the estimated probability from LightGBM surpasses the threshold of 0.5, a commonly employed criterion in classification trees.[33] In Figure 4, each red (or green) point signifies whether a 39-foot segment is estimated to have a broken rail (red) or not (green) based on the LightGBM estimation. Each line represents a track, with multiple subdivisions along it, commonly referred to as "tunnel sections" within this commuter railroad. Based on the estimation, both Track A and Track D exhibit a widespread presence of red points scattered along their entire length. Consequently, it is imperative to conduct comprehensive inspections along the entirety of Tracks A and D, with a primary focus on the high-risk positions indicated by the red points. In contrast, Tracks B, C, E, and F display clusters of red points primarily concentrated in specific tunnel sections. In these cases, a targeted inspection approach can be adopted, concentrating efforts on the tunnel sections where red points are prevalent. Notably, Track G stands out as it demonstrates a notably healthy rail condition, with minimal to no red points detected. Given this, if the inspection team faces time constraints, prioritizing Tracks A and D for inspection should take precedence, with Track G scheduled for inspection last due to its relatively low risk profile.

Classification metrics

In order to assess the performance of the proposed modeling framework, we utilize classification metrics such as the confusion matrix, accuracy (equation (1)), and recall (equation (2)). The primary objective of this paper is to aid commuter railroad companies in estimating the likelihood of broken rails based on historical data, with a particular focus on identifying the

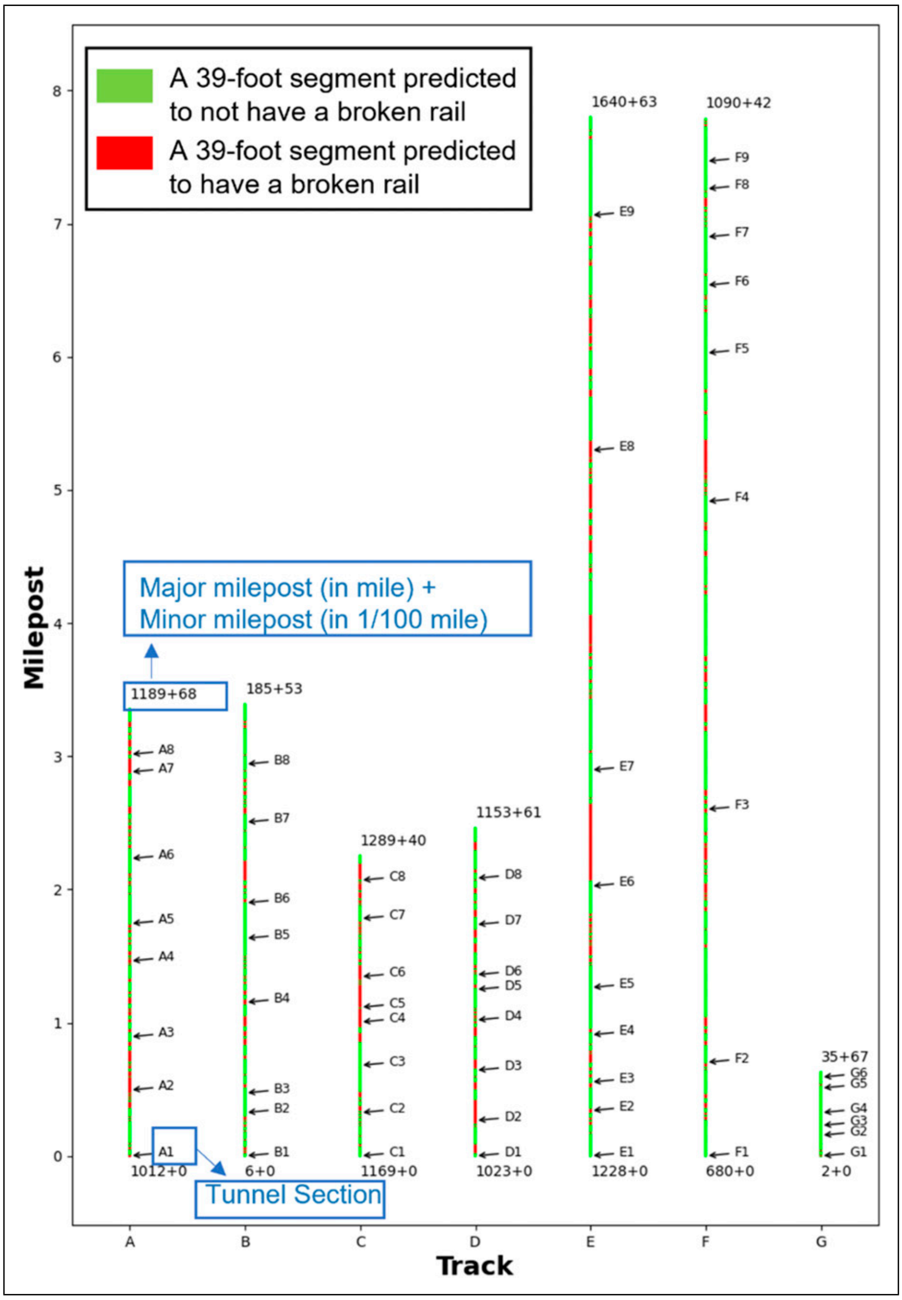

Figure 4. The broken rail estimation results for each segment.

locations where broken rails are most likely to occur. The ideal model should detect as many instances of broken rails as possible, thus aiming for a high recall value. Tolerating false positives (cases where the model predicts broken rails, but they are not present) is acceptable since commuter railroads conduct inspections on every rail segment by default.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Sample Size}} \tag{1}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{2}$$

Figure 5 displays the confusion matrix generated by the model employing LightGBM and random oversampling. According to the confusion matrix, the overall accuracy stands at 0.7114, signifying that the model achieves a 71.14% accuracy rate, which is quite commendable. The recall value, at 0.7113, accomplishes our goal effectively, as it indicates that our model successfully identifies 71.13% of the total broken rails within the testing dataset.

Feature importance

Figure 6 illustrates the feature importance based on the “F score” derived from the LightGBM algorithm. The “F score” aggregates the frequency of each input variable’s utilization in splitting nodes and provides insight into the significance of each variable in the final estimation. Notably, when estimating

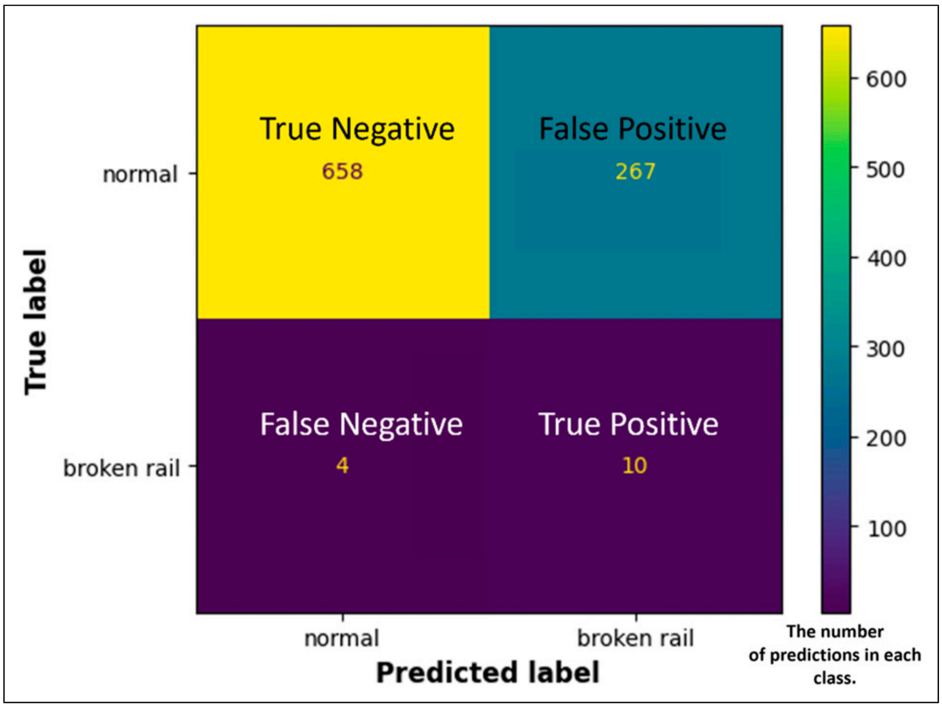

Figure 5. Confusion matrix of the ground-truth labels and model predictions.

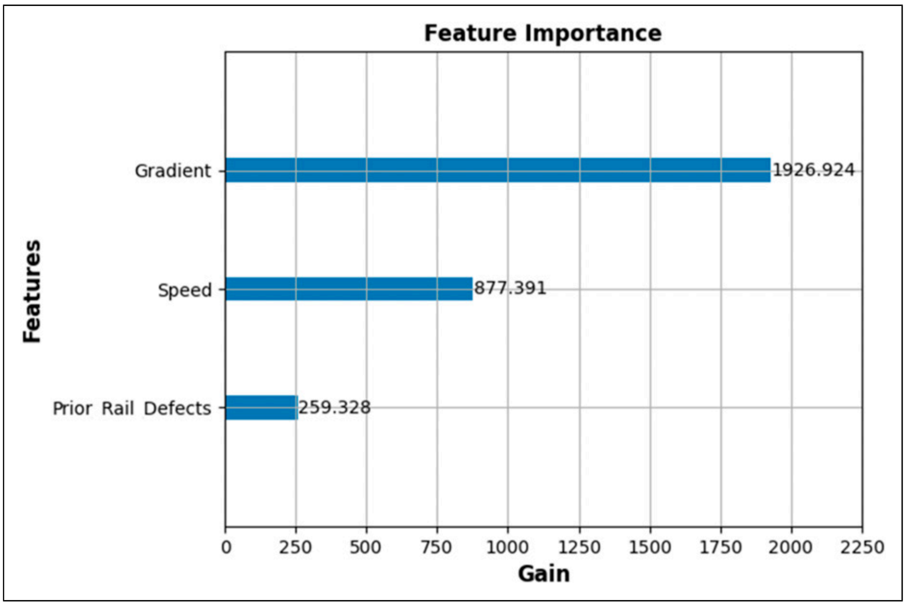

Figure 6. “F score” for each feature generated by the LightGBM algorithm.

the probability of a broken rail, the track design characteristic "gradient" emerges as the most influential factor, followed by "speed" and "prior rail defect." As previously mentioned in the introduction section, a key distinction between freight railroads and commuter railroads lies in the weight of fully loaded trains, with commuter trains generally being lighter than their freight counterparts. Consequently, the variable of "traffic" does not carry substantial weight in contributing to the estimation of the likelihood of broken rails for commuter railroads. It is worth noting that Wang et al.[33] conducted an analysis focused on freight railroads and found that, in that context, traffic loading outweighs the importance of track design characteristics like gradient and curvature when estimating the probability of a broken rail. Nonetheless, these findings suggest potential variations in the context of passenger rail systems, underscoring the significance of conducting comprehensive studies tailored to the commuter railroad. Unlike gradient, curvature does not appear to play a significant role in the model. This could be attributed to the relatively stable nature of curvature, as indicated by the correlation plot in Figure 2, in contrast to the more variable gradient, which has a more pronounced impact on broken rail occurrences.

Discussion and conclusions

Broken rail prevention is of utmost importance to the rail industry. Knowing the locations with a high probability of broken rail occurrence would help the walk-through process to prioritize inspection concentration. When the commuter rail agency deploys a walk-through inspection, they can prioritize segments with a high estimated risk of broken rail and then focus on other low-risk segments. However, due to the infrequency of broken rail occurrences in commuter rail systems, the scarcity of data related to such incidents and the resulting imbalance between positive (having broken rails) and negative (not having broken rails) cases pose significant challenges when attempting to estimate the probability of a broken rail event. To narrow this research gap, this paper proposes an ML-based analytical modeling framework with multiple oversampling techniques to estimate the likelihood of the occurrence of a broken rail on commuter railroads.

As described in Figure 1, this paper employs five-fold cross-validation, and augmenting the training set using various oversampling techniques, including ADASYN, random sampling, and SMOTE. In this paper, the primary machine learning algorithms are LightGBM and XGBoost. Additionally, we have implemented Logistic Regression and Random Forests for comparative analysis. Prior to inputting the data into machine learning algorithms, this paper performs hyperparameter tuning for tree-based algorithms, namely LightGBM, XGBoost, and Random Forest, using Bayesian optimization. The overall model performance is initially compared using AUC values, with the combination of random oversampling and LightGBM producing the largest AUC values, outperforming other selected machine learning algorithms in this study. Consequently, with the utilization of this combination, we proceed to present more comprehensive classification metrics, encompassing the broken rail estimation outcomes for each rail segment, the confusion matrix, recall, and accuracy. The ML-based modeling framework we selected demonstrated a successful classification rate of 71.14% on an unseen testing dataset, indicating a commendable level of performance. Given that this paper is designed based on a practical need to prioritize inspection efforts on segments with a higher probability of having a broken rail, the primary aim is to specify as many broken rails as possible, with less concern for false positives. Consequently, the focus is on achieving a high recall value, as indicated by the provided recall value of 0.7113 in the paper. According to the F-score, among the five input variables, "gradient," "speed," and "prior rail defect" are the top three most important features to estimate a broken rail, with almost no impact of "traffic" and "curvature." The observations for the studied commuter railroad significantly differ from those for freight railroads. This difference can be attributed to the substantial operational distinctions between freight railroads and commuter railroads described in the Introduction section. In a prior study focusing on freight railroads,[33] the three most crucial features were identified as "number of defects," "minimum temperature," and "days from last failure." First, on freight rail segments, the presence of multiple defects is a common occurrence, making the number of defects a crucial indicator for predicting a broken rail. Conversely, in commuter railroads, a significant majority of rail segments have no prior defects, diminishing the efficacy of the prior rail defects as a strong predictor for broken rails on commuter railroads. Secondly, the extensive length of the studied freight railroad (over 20,000 miles) compared to the commuter railroad in our study (28 miles) results in less apparent temperature variance on commuter railroads but significant variation on freight railroads. Lastly, the implementation of half-year inspections on the studied commuter railroad, coupled with the practice of replacing the entire segment upon defect detection, renders the feature "days from last defect" inapplicable to commuter railroads. These operational disparities elucidate the distinct characteristics influencing feature importance in the two contexts.

It is worth noting that various other factors can influence the occurrence of broken rails. These factors encompass maintenance practices, environmental conditions (such as temperature, humidity, and precipitation), geographic factors (including soil type, elevation changes, and geological shifts), and track age. Notably, the study corridor in this paper spans just 28 miles. When examined historical environmental data across this 28-mile track, we observed minimal variation over the studied period. Consequently, environmental factors were not included in this analysis. Similarly, geographic factors demonstrated limited variation within this relatively short corridor. As a result, this paper did not include geographic factors as influential variables in the predictive model. Unlike the considerable variability in track age often observed in freight railroads, the track age of the commuter railroad we studied remained relatively consistent. Therefore, track age was excluded as a feature. Turning to maintenance data, it is essential to acknowledge that this type of data was primarily recorded and maintained by railroad maintenance crews. However, there were instances where some minor maintenance activities were not consistently documented, raising concerns about data reliability and quality. Consequently, despite obtaining handwritten maintenance records, this data was excluded from the predictive model. Instead, this paper prioritized machine-inspected data, which offered a more reliable and consistent information source.

A potential source of estimation errors could be attributed to a misalignment in the data period between "prior rail defect" and "broken rail" records. While broken rails have been well-documented over the past 5 years, the availability of consecutive prior rail defect records is not consistent. Specifically, inspection records were traditionally documented on paper (non-digital), and for certain years (specifically, 2018 and 2020), they are challenging to locate. Furthermore, it is important to note that the frequency of prior rail defect inspections has undergone changes over time. Before 2021, inspections were conducted twice a year, but from 2022 onwards, they were increased to four times per year. For future studies, the increased frequency of inspections offers the potential for conducting time series analyses to predict broken rails more accurately in the future. Access to these datasets would be invaluable for advancing this paper.

The railway industry is characterized by its traditional nature and a tendency to adopt new technologies gradually. Challenges such as manual data recording by railway workers can diminish model accuracy due to their lack of reliability. This article, serving as an initial investigation into commuter railroad broken rail occurrences, offers opportunities for enhancement. For instance, delving into historical maintenance data represents a vital avenue for future exploration. Furthermore, the current dataset does not adequately support a time series analysis due to the shortage of broken rail samples. Even if maintenance staff begins recording more detailed information, it would still take several years to amass a sufficient number of broken rail samples for predicting future conditions. In the future, once enough data has been collected to enable a time series analysis, the proposed framework can be adjusted and tailored for new prediction tasks. This paper serves as a foundation for future studies investigating broken rail locations and their associated theoretical root causes. By establishing a solid groundwork, it opens the door to a more in-depth exploration of the features surrounding broken rail occurrence.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded through a contract by the Federal Railroad Administration (693JJ620C000017).

ORCID iDs

Di Kang https://orcid.org/0000-0003-3337-2815
Junyan Dai https://orcid.org/0000-0003-2647-4178
Xiang Liu https://orcid.org/0000-0002-4348-7432
Xin Wang https://orcid.org/0000-0002-7433-6768

References

1. Kang D, Zhao J, Tyler Dick C, et al. Probabilistic risk analysis of unit trains versus manifest trains for transporting hazardous materials. *Accid Anal Prev* 2023; 181: 106950.
2. Zhao J, Dick CT and Kang D. Analysis of derailment severity comparing unit trains at transload terminals and manifest trains at railroad switching and hump classification yards. *Transport Res Rec* 2023;2677(5):793–811. .
3. Liu X, Kang D, Dick TC, et al. *Risk analysis methodology for unit and manifest trains transporting hazardous materia* [Internet]. United States. Department of Transportation. Federal Railroad Administration; 2022 [cited 2024 Aug 16]. Available from: https://rosap.ntl.bts.gov/view/dot/76417
4. Kang D, Kirkpatrick SW, Zhang Z, et al. Freight train derailment severity prediction: a physics-informed one-dimensional model. *Smart and Resilient Transportation* [Internet]. 2024 [cited 2024 Mar 17]; Available from: https://www.emerald.com/insight/content/doi/10.1108/SRT-10-2023-0008/full/html
5. Kang D. Risk analysis of unit trains versus manifest trains for transporting hazardous materials [Internet] [PhD Thesis]. *Rutgers The State University of New Jersey, School of Graduate Studies*; 2024 [cited 2024 Aug 16]. Available from: https://search.proquest.com/openview/82edd2c7143f2cae805203f6b9b49f49/1?pq-origsite=gscholar&cbl=18750&diss=y
6. Sireesha R, Ajay Kumar B, Mallikarjunaiah Get al. Broken rail detection system using RF technology. *SSRG International Journal of Electronics and Communication Engineering*. 2015; 2(4): 11.
7. Luthfi KM, Sugiana A and Suratman FY. Broken rail detection system using laser. *IOP Conf Ser Mater Sci Eng* 2021; 1098(4): 042045.
8. Liu X, Lovett A, Dick T, et al. Optimization of ultrasonic rail-defect inspection for improving railway transportation safety and efficiency. *J Transp Eng* 2014; 140(10): 04014048.
9. Liu X. Optimizing rail defect inspection frequency to reduce the risk of hazardous materials transportation by rail. *J Loss Prev Process Ind* 2017; 48: 151–161.
10. Zaman A, Huang Z, Li W, et al. *Development of railroad trespassing database using artificial intelligence* [internet]. United States. Department of Transportation. Federal Railroad Administration, 2024. [cited 2024 Mar 17]. Available from: https://rosap.ntl.bts.gov/view/dot/73444
11. Latvala J, Nurmikolu A and Luomala H. Problems with railway track drainage in Finland. *Procedia Eng* 2016; 143: 1051–1058.
12. Altay A and Baykal-Gürsoy M. Imperfect rail-track inspection scheduling with zero-inflated miss rates. *Transport Res C Emerg Technol* 2022; 138: 103608.
13. Zhou K. *Artificial-intelligence-aided prediction of broken rail-caused derailment risk* [PhD Thesis]. New Brunswick, NJ: Rutgers The State University of New Jersey, School of Graduate Studies; 2019.
14. Zhang Z, Zhou K and Liu X. Broken rail prediction with machine learning-based approach. In: ASME/IEEE Joint Rail Conference. American Society of Mechanical Engineers, 2020, p. V001T08A014.
15. Ekberg A and Kabo E. *Surface fatigue initiated transverse defects and broken rails–an International Review*. Vienna: International Collaborative Research Initiative, 2014. Report No.: ISSN 1652-8549.
16. Dai J and Liu X. Machine learning based prediction of rail transit signal failure: a case study in the United States. *Proc Inst Mech Eng F J Rail Rapid Transit* 2023; 237(5): 680–689.
17. Kaewunruen S, Sresakoolchai J and Zhu G. Machine learning aided rail corrugation monitoring for railway track maintenance. *Struct Monit Maint* 2021; 8(2): 151–166.
18. Li C, He Q and Wang P. Estimation of railway track longitudinal irregularity using vehicle response with information compression and Bayesian deep learning. *Computer aided Civil Eng* 2022; 37(10): 1260–1276.

19. Chen M, Zhai W, Zhu S, et al. Vibration-based damage detection of rail fastener using fully convolutional networks. *Veh Syst Dyn* 2022; 60(7): 2191–2210.
20. Ngamkhanong C and Kaewunruen S. Prediction of thermal-induced buckling failures of ballasted railway tracks using artificial neural network (ANN). *Int J Str Stab Dyn* 2022; 22(5)): 2250049.
21. Zaman A, Huang Z, Li W, et al. Artificial intelligence-aided grade crossing safety violation detection methodology and a case study in New Jersey. *Transport Res Rec: J Transport Res Board* 2023; 2677(10): 688–706.
22. Andrews JRC. *The demise of the lone star rail district: a case study in the failure to expand passenger rail service in Texas* [PhD Thesis]. Philadelphia, PA: Temple University, 2022.
23. Sun Y, Guo Q, Schonfeld P, et al. Evolution of public transit modes in a commuter corridor. *Transport Res C Emerg Technol* 2017; 75: 84–102.
24. D Knowles R and Ferbrache F. Evaluation of wider economic impacts of light rail investment on cities. *J Transport Geogr* 2016; 54: 430–439.
25. Allen JG and Newmark GL. Commuter rail electrifications that never were and what they teach us. *Transport Res Rec*. 2022; 2677(1): 03611981221101028.
26. Allen JG. Reasons for commuter rail electrification: early 20th century and since 2000. *Transport Res Rec* 2019; 2673(7): 227–238.
27. Khashe Y and Meshkati N. Human and organizational factors of positive train control safety system the application of high reliability organizing in railroad. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. Los Angeles, CA: Sage Publications Sage CA, 2019, pp. 1824–1828.
28. Onodera M, Toyooka A, Yoshida Y, et al. Estimation method of seismic damage of bearings at railway structure by displacement sensor. *Q rep RTRI* 2022; 63(1): 64–69.
29. Rungskunroch P, Jack A and Kaewunruen S. Benchmarking on railway safety performance using Bayesian inference, decision tree and petri-net techniques based on long-term accidental data sets. *Reliab Eng Syst Saf* 2021; 213: 107684.
30. Zarembski AM. Management of broken rail risk for high speed passenger rail. In: Joint Rail Conference, April 27, 2010, Urbana, IL, pp. 15–22.
31. Dick CT, Barkan CP, Chapman ER, et al. Multivariate statistical model for predicting occurrence and location of broken rails. *Transport Res Rec* 2003; 1825(1): 48–55.
32. Schafer DH and Barkan CP. A prediction model for broken rails and an analysis of their economic impact. In: Proc, American Railway Engineering and Maintenance of Way Association (AREMA) Annual Conference, September 21, 2008, Salt Lake City, UT.
33. Wang X, Liu X and Bian Z. A machine learning based methodology for broken rail prediction on freight railroads: a case study in the United States. *Construct Build Mater* 2022; 346: 128353.
34. Ghofrani F, Chava NK and He Q. Forecasting risk of service failures between successive rail inspections: a data-driven approach. *J Big Data Anal Transp* 2020; 2(1): 17–31.
35. Ghofrani F, Sun H and He Q. Analyzing risk of service failures in heavy haul rail lines: a hybrid approach for imbalanced data. *Risk Anal* 2022; 42(8): 1852–1871.
36. Chawla NV, Bowyer KW, Hall LO, et al. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 2002; 16: 321–357.
37. He H, Bai Y, Garcia EA, et al. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), Hong Kong, 01–08 June 2008. IEEE, 2008, pp. 1322–1328.

Appendix A

Table A.1. Hyperparameter tuning results for tree-based machine learning algorithms.

(a) Using Random Forest.

			Tuned hyperparameter values			
Hyperparameter	Type	Range	No resampling	Random oversampling	SMOTE	ADASYN
Number of trees	Int	[100, 1000]	104	663	273	180
Maximum tree depth	Int	[2,15]	14	6	7	8
Minimum samples required to split	Int	[2,10]	2	2	6	7

(b) Using XGBoost

			Tuned hyperparameter values			
Hyperparameter	Type	Range	No resampling	Random oversampling	SMOTE	ADASYN
Learning rate	Float	[0, 0.1]	0.09	0.09	0.05	0.1
Number of boosting rounds	Int	[10, 1000]	938	842	419	882
Maximum tree depth	Int	[2, 15]	13	4	10	11
Minimum sum of instance weight required in a child	Int	[0, 5]	2	3	1	2
Minimum loss reduction required to split	Float	[0, 0.5]	0.33	0.38	0.37	0
Subsample ratio of the training instance	Float	[0.5, 1]	0.6	0.83	0.72	0.5
Subsample ratio of columns when constructing a tree	Float	[0.5, 1]	0.81	0.55	0.95	0.84

(continued)

Table A.1. (continued)

(c) Using LightGBM

Hyperparameter	Type	Range	Tuned hyperparameter values			
			No resampling	Random oversampling	SMOTE	ADASYN
Learning rate	Float	[0, 0.1]	0.1	0.02	0.07	0.05
Number of boosting rounds	Int	[10, 1000]	775	191	876	726
Maximum tree leaves	Int	[2, 50]	11	48	35	34
Maximum tree depth	Int	[2, 15]	10	13	14	13
Minimum number of data needed in a child	Int	[2, 50]	12	28	3	43
Minimum loss reduction required to split	Float	[0, 0.5]	0.13	0.17	0.35	0.26
Subsample ratio of the training instance	Float	[0.5, 1]	0.65	0.89	0.76	0.69
Subsample ratio of columns when constructing a tree	Float	[0.5, 1]	0.91	0.55	0.93	1