**Preprint Manuscript**
Bridgelall, R., and Tolliver, D. (2021). Railroad Accident Analysis Using Extreme Gradient Boosting. *Accident Analysis and Prevention*, 156(2021). DOI: 10.1016/j.aap.2021.106126, 2021(106126).

# Railroad Accident Analysis Using Extreme Gradient Boosting

**Raj Bridgelall, Ph.D**., Corresponding Author

Assistant Professor, Department of Transportation, Logistics & Finance

College of Business, North Dakota State University

Fargo, ND 58108; Email: raj@bridgelall.com, ORCID: 0000-0003-3743-6652

**Denver D. Tolliver, Ph.D.**

Director, Upper Great Plains Transportation Institute, North Dakota State University

Fargo, ND 58108; Email: denver.tolliver@ndsu.edu, ORCID: 0000-0002-8522-9394

Declarations of Interest: None.

# Railroad Accident Analysis Using Extreme Gradient Boosting

## Abstract

Railroads are critical to the economic health of a nation. Unfortunately, railroads lose hundreds of millions of dollars from accidents each year. Trends reveal that derailments consistently account for more than 70% of the U.S. railroad industry's average annual accident cost. Hence, knowledge of explanatory factors that distinguish derailments from other accident types can inform more cost-effective and impactful railroad risk management strategies. Five feature scoring methods, including ANOVA and Gini, agreed that the top four explanatory factors in accident type prediction were track class, type of movement authority, excess speed, and territory signalization. Among 11 different types of machine learning algorithms, the extreme gradient boosting method was most effective at predicting the accident type with an area under the receiver operating curve (AUC) metric of 89%. Principle component analysis revealed that relative to other accident types, derailments were more strongly associated with lower track classes, non-signalized territories, and movement authorizations within restricted limits. On average, derailments occurred at 16 kph below the speed limit for the track class whereas other accident types occurred at 32 kph below the speed limit. Railroads can use the integrated data preparation, machine learning, and feature ranking framework presented to gain additional insights for managing risk, based on their unique operating environments.

**Keywords**: data cleaning; feature engineering; financial loss; machine learning; principle component analysis; risk management

# 1 Introduction

U.S. railroads have been an important driver of economic progress for more than 150 years. Today, U.S. railroads carry approximately one-third of the nation’s exports [1]. Therefore, the safe and efficient operation of railroads is crucial to the nation’s economic health. Unfortunately, railroads lose hundreds of millions of dollars from accidents each year. Analysis of the Federal Railroad Administration (FRA) Rail Equipment Accident database revealed that human-factors was consistently the dominant cause of railroad accidents [2]. Hence, the federal government mandated that railroads deploy a positive train control (PTC) system by 2018 to help prevent accidents caused by human errors [3]. With PTC now in place, it is important for analysts to study other common causes of accidents.

The **goal** of this research is to identify factors associated with the most frequent and expensive types of accidents that are not attributable to human error. Data mining of FRA accident records from January 1, 2009, to June 30, 2020, revealed that derailment accidents accounted for 70.9% of the average annual financial loss (Figure 1). The trend showed that derailment accidents maintained a steady rate each year. Therefore, the ability to identify and rank features that increase the risk of derailments over other accident types can inform more cost-effective and impactful risk management strategies.

An **objective** of this research is to build a supervised machine learning (ML) model that can predict derailments from other accident types and to rank the importance of those features that contribute towards the classification accuracy. However, no single type of ML model performs best on all types of datasets. Therefore, another objective is to compare the classification performance of various types of ML models on the same dataset.

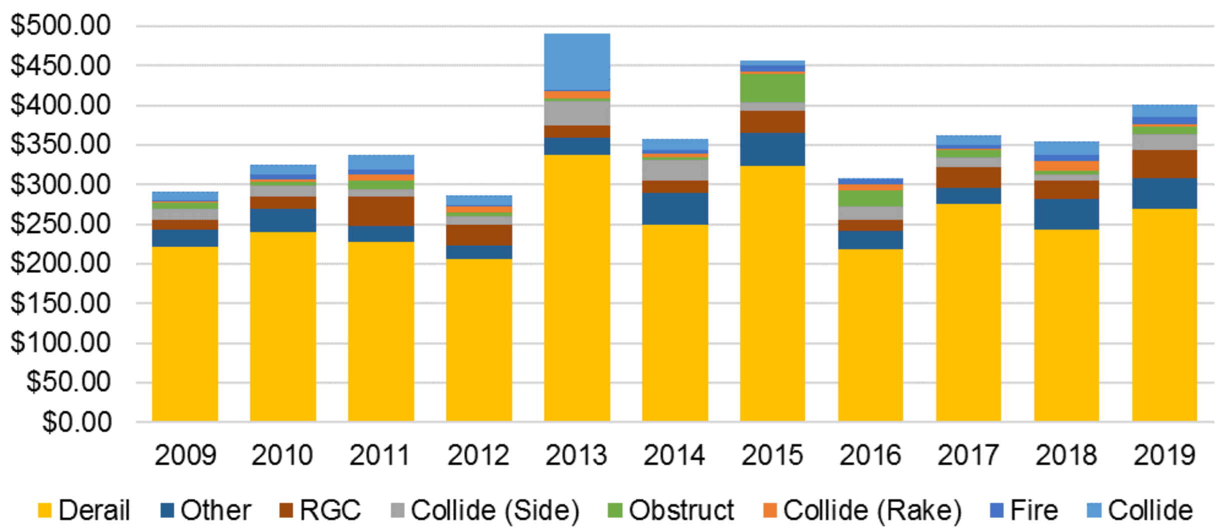

Figure 1: Annual financial loss reported for different accident types.

One of the main challenges in data science is to effectively clean datasets before using them to train ML models. Studies estimate that dirty data costs the U.S. economy trillion of dollars each year [4]. A survey of data cleaning for ML found that the failure to discover and repair dirty data can weaken data analysis techniques [5]. Although a few approaches to data cleaning are common, every dataset poses unique challenges [6]. Hence, data scientists spend an average of 60% of their time cleaning and organizing data [4].

Although the importance of using clean data is well-known, the research community has paid little attention to the advancement of data cleaning techniques [7]. The most commonly used techniques  are those that detect and remove outliers and duplicate records [4]. Even so, those techniques alone cannot effectively clean all types of datasets. Other techniques that can find data entry errors use customized rules to detect violations, for example, house prices exceeding an expected range for a given neighborhood. Custom techniques tend to be heuristic, so they require good familiarity with the data and its meaning. Considering the challenges outlined above, the following are **contributions** of this research:

- A customized framework to clean a relevant subset of the FRA database and to fill 100% of missing values for the important attributes (Section 3.1).
- Brief explanations of how each ML method works to gain understanding about the impact of their hyperparameter tuning (Section 3.2).
- Importance ranking of the feature relevance in predicting accident type (Section 3.3).
- Visualizing and interpreting the classification power of each attribute by principle component analysis (PCA) to gain insights about the performance differences among the ML models evaluated (Section 3.4).

The next section (Section 2) reviews related works and their findings in relation to the contributions of this research. Section 4 mirrors the methods section to present the results. Section 5 discusses the significance and interprets the outcome. Section 6 recaps the findings and concludes with how future research can leverage the methods of this research to further the agenda in accident analysis.

# 2 Related Works

Studies that use ML methods to analyze accidents are more common for roadways than for railroads. For example, Iranitalab and Khattak (2017) compared the performance of Multinomial Logit (MNL), k-Nearest Neighbor (kNN), Support Vector Machines (SVM) and Random Forests (RF) in predicting the crash severity of two-vehicle roadway crashes [8]. They found that kNN and MNL had the best and worst performance, respectively, when applied to crash data from Nebraska, United States. A recent survey of big data analytics applied to railroads found that of 115 journal articles reviewed from 2003 to 2017, only 22% covered railroad safety whereas 49% and 29% covered maintenance and operations, respectively [9]. This imbalance suggests that the research community and the railroad industry can benefit from additional analysis of railroad accident risks.

Several studies used ML techniques to analyze highway-rail grade crossing (HRGC) accidents. Dabbour et al. (2017) applied ordered regression models to HRGC crash data and found that higher train and vehicle speeds were positively correlated with driver injury severity [10]. Liu and Khattak (2017) applied geospatial modeling to HRGC crash data and found that gate violations were more highly associated with two-quadrant than four-quadrant gates [11]. Karamati et al. (2020) applied random survival forest to HRGC crash data and found that adding audible alarm devices to crossings that already have gates and flashing lights can decrease crash likelihood by approximately 50% [12]. Soleimani et al. (2019) used extreme gradient boosting to identify HRGCs that should be closed to prevent accidents [13]. Wali et al. (2021) applied text mining to crash narrative data of railroad trespassing incidents and found that confirmed suicide attempts and the use of headphones or cellphones were more likely to result in fatal injuries [14].

Only a few studies focused on derailment-type accidents. Liu et al. (2017) found that derailment rates on Class 1 railroad mainlines were lower for signalized tracks with higher FRA track class and higher traffic density [15]. Wang et al. (2020) found that most derailment type accidents declined with the greatest reductions in broken rails, irregular track geometry, and wheel-related equipment defects [16]. Iranitalab and Khatta (2020) found that the random forest method of ML outperformed the logistic regression, Naïve Bayes, and support vector machine (SVM) methods in classifying train-level hazmat releases with an AUC score of 87% [17].

The survey of Ghofrani (2018) demonstrated that researchers have also use ML methods to analyze other aspects of railroad operations besides safety [9]. For example, Li et al. (2014) used ML to learn rules from historical and real-time data to predict railroad maintenance needs [18].

Lasisi and Attoh-Okine (2019) proposed a combination of ensemble tree-based ML models to predict rail fatigue defects and achieved an AUC score of 0.783 [19].

# 3 Methodology

Figure 2 shows the methodological framework developed to prepare the data, apply the machine learning methods, rank the features, and to interpret the results.

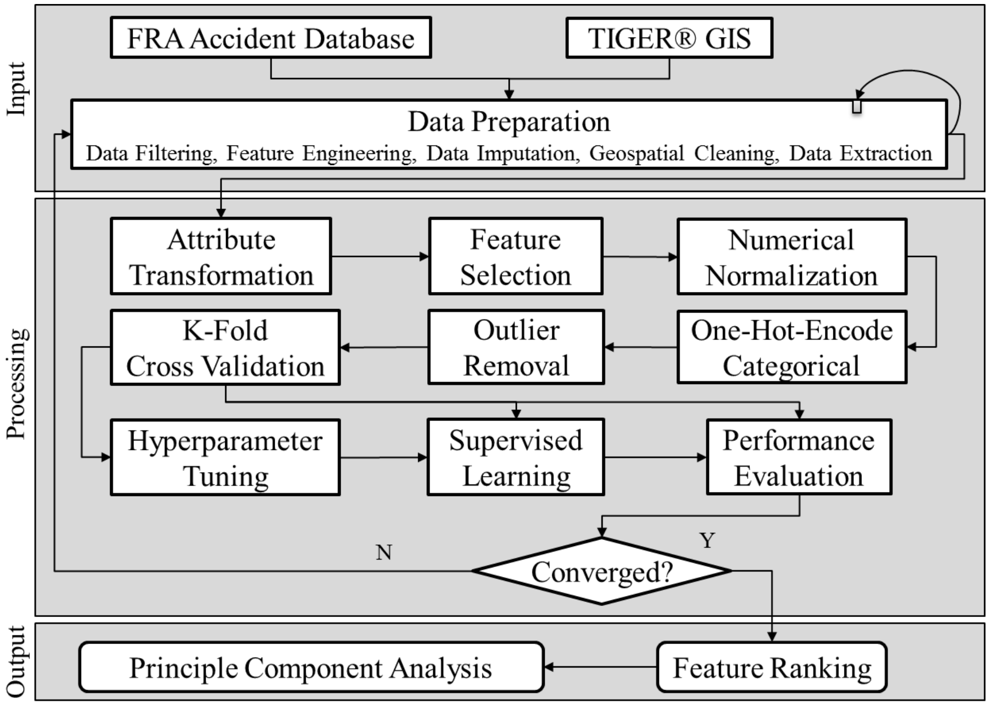

Figure 2: The methodological framework.

The next two subsections cover each procedure shown in the input, processing, and output layers of the framework. The input layer gathers the datasets and prepares the combined data by applying various methods to reduce noise, repair data entry errors, and fill in missing values. The processing layer prepares relevant attributes to train and tune the ML models. The ML process itself resulted in the discovery of additional errors that the data preparation layer subsequently addressed. The looping converged after the ML performance stabilized. The final layer ranked the importance of attributes in classification performance and used PCA to visualize the results for interpretation.

## 3.1 Data Preparation

Erroneous data or attributes that have no influence in deciding the target class (derailment versus non-derailment) become noise and diminish the predictive performance of ML models. Missing or low dispersion data can increase model bias. The next subsections describe some customized data cleaning and imputation methods that this research developed for the FRA dataset.

### *3.1.1 Data Filtering*

"Big data" is often associated with what the literature calls a "curse of dimensionality" where each additional attribute exponentially increases the volume of the feature space to a point where the data becomes too sparse to be statistically significant or to have any structure [20]. Therefore, methods to identify and remove irrelevant attributes or features can increase the cohesiveness and quality of the dataset. Table 1 describes criteria used to eliminate irrelevant attributes or features.

Table 1: Criteria for Attribute or Feature Elimination

| Criteria | Description |
|---|---|
| Sparsity | Attribute is missing more than 85% of the values. |
| Duplication | Attribute contains the same information as other attributes. |
| Sparsity | More than 85% of the attribute contain zeros. |
| Correlated | Attribute is more than 90% correlated with another. |
| Redundancy | Attribute contains information that is inherent in other attributes. |
| Noise | Attribute is not relevant to the target class. |
| Dispersion | Attribute has low variance or carries little or no information. |
| Combinable | Attribute that can combine with others without losing information. |

### *3.1.2 Feature Engineering*

The manipulation of features to improve ML model performance is more art than science because there are no automated or standardized techniques for all types of datasets [20]. The effectiveness of feature engineering requires in-depth knowledge of the dataset, its structure, and the meaning and significance of each attribute. The empirical feature engineering was conducted as follows:

1) Packaged similar features of an attribute to simplify the categories.
2) Converted categorical attributes that have some ranking to ordinal attributes.
3) Binarized categorical attributes that contained only two values by replacing one value with zero and the other with one.
4) Replaced nominal values with a single word label to enhance the ease of interpreting trends with more descriptive legends.

#### *3.1.3 Data Imputation*

A few methods such as decision trees and Bayesian classifiers can work with missing data, but most cannot [21]. Therefore, data scientists developed a few methods to impute or guess missing values. Common approaches are to replace missing values with the mean, median, most frequent, random, or zero value. More intelligent approaches use tree-based ML techniques to fill missing values with those of their nearest neighbors. This research developed a new method, dubbed *local association pivot* (LAP), to replace missing values. The LAP method first creates a pivot table that aggregates non-missing values by a location identifier and by sub-location identifiers if available. The method then merges the pivot table with the dataset by using the main location identifier as the unique merge key. The aggregation method for the pivot depends on the type of missing data. For example, for numerical values such as track density, the method used the maximum of the aggregated value for a location. The method did not use the average value because zero or missing values created an undesirable bias in the aggregation. A fringe benefit of using the LAP method is that it is easy to spot data entry or spelling errors by examining a sorted list of the unique location keys.

### 3.1.4 Geospatial Cleaning

Missing or erroneous geospatial coordinates are impossible to impute or correct if no other spatial information is available in the dataset. The state or county name provides a coarse location identifier that can be helpful for visualizing data on maps. However, a coarse location such as a large state may introduce bias in the ML process. Fortunately, the FRA database contains the station name that is closest to the accident location, so its location can be a surrogate for missing geospatial coordinates.

Aside from missing geospatial coordinates, data entry errors may result in erroneous or highly skewed geospatial locations. Figure 3 shows the positions of the recorded geospatial coordinates relative to a map of the continental United States. There is an observable systematic skew towards the southeast. This skew suggests that there was a lack of resolution for those coordinates because in North America, lower resolution latitude and longitude coordinates would bias towards the south and east, respectively.

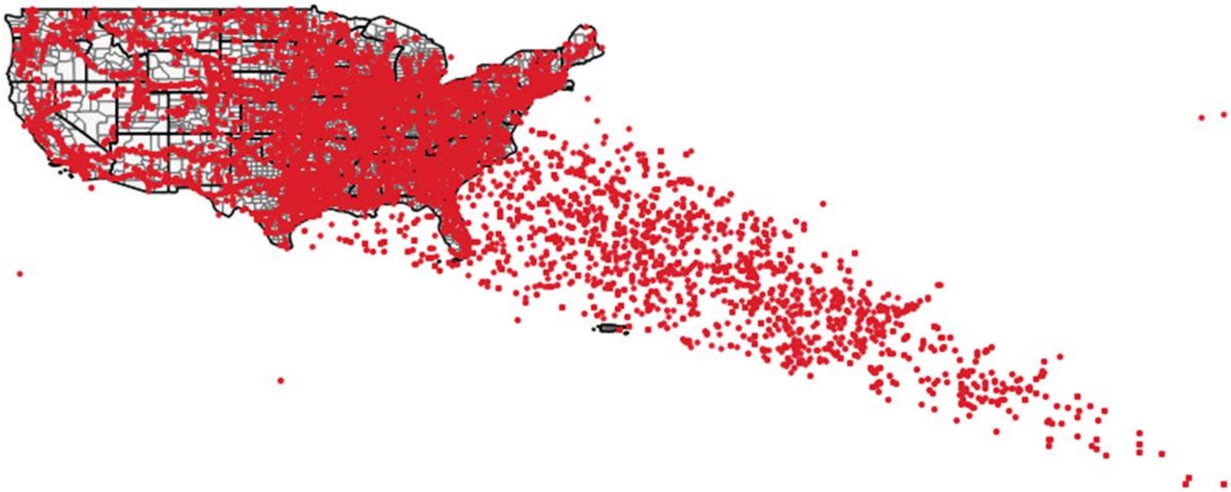

Figure 3: Positions of the recorded geospatial coordinates in the FRA database.

The procedure to clean the geospatial coordinates filled missing values in two stages. First, the LAP method averaged the non-missing geospatial coordinates for accidents that occurred on a given track type near a given station. Second, the procedure merged the records with a map file from the U.S. Census Bureau TIGER® database that contained the geospatial centroid of each county in the United States. A geographic information system (GIS) spatial join method then replaced erroneous geospatial coordinates as follows:

1. Spatially join the TIGER county polygons to the FRA geospatial coordinates using one-to-many mapping. This procedure added the county FIPS code from the TIGER database.
2. Flag any mismatch between the reported FRA county FIPS code and the spatially joined county FIPS code by a Boolean flag MATCH. This flag identified geospatial coordinates that were not located within the FRA county recorded for the accident.
3. Replace the geospatial coordinates of the flagged records with the geospatial centroid of the FRA reported county.

A clear limitation of the LAP method is that it reduces the geospatial error by a small amount. However, the error reduction helps the ML performance without forcing data elimination.

### *3.1.5 Data Extraction*

A consistent dataset improves ML performance [20]. The FRA dataset contained records of both passenger and freight train accidents. Passenger trains operate in environments and circumstances that are often different from those of freight trains. For example, passenger terminals and stations are different from freight and transshipment terminals. Hence, equipment and operations are different for the two types of service. Passenger trains accounted for a small portion 8.03% (2,354) of accidents from 2009 to July 31, 2020. Removing those records not only enhanced the consistency of the dataset, but also removed a few attributes that were associated with passenger trains only.

The FRA database codes the cause of an accident in the "ACCAUSE" attribute and cross-referenced the description in Appendix C of the accident dictionary [22]. The document lists 389 accident-cause codes. The first character of the code indicates the accident cause category as “E” for “Mechanical and Electrical Failures, “T” for “Rack, Roadbed and Structures,” “S” for “Signal and Communication,” “H” for “Train operation – Human Factors,” and “M” for miscellaneous causes that do not fit into any of the other categories. Therefore, the procedure removed records for accidents due to human factors by extracting those where the first character in the cause-code was not “H.”

### *3.1.6 Attribute Transformation*

ML algorithms tend to perform poorly on data with attributes that have a highly skewed distribution because the model could treat data in long tails as outliers or because extreme values provide insufficient examples [23]. This phenomenon is common, for example, in the distribution of annual income (right skew) and the distribution of age at natural demise (left skew). A standard technique is to log transform continuous attributes with highly skewed distributions, including the target attribute if applicable. Using the shifted natural logarithm LN(1 + $x$) prevents an undefined number if attribute value $x$ is zero.

Another transformation that can help to reduce the dimension of a dataset is to replace a set of related attributes with proportions of a base attribute. The advantage of a proportion transformation is that it retains information about the relative relationship among attributes while normalizing the values within the [0, 1] range. The attributes selected for this transformation were the proportion of cars that contained freight (LOADF1), and the proportion of loaded cars that contained hazardous materials (CARS).

It is also possible to increase the information of an attribute by making explicit some knowledge that is within context of another attribute. For example, transforming the absolute train speed with the excess speed, relative to the speed limit for the track class, increases information for that attribute. Finally, attributes that are irrelevant but provide descriptive value can store metadata instead of features to enhance further exploratory data analysis.

### *3.1.7 Feature Selection*

Predictive modeling should not contain attributes where values are known only after the outcome. Therefore, the cleaning procedure must eliminate *post-event* attributes such as the number of people injured, killed, or evacuated. Other post-event attributes include the position of involved equipment, the number of damaged vehicles, and the cause of the accident.

### *3.1.8 Attribute Normalization*

Most ML algorithms work only with numerical data. Categorical attributes contain a finite number of unique labels that have no numeric value, nor do they represent an ordering or ranking. Therefore, the framework applies *one-hot-encoding* to create new binary attributes that represent each category or feature of the attribute where “1” and “0” denotes presence or absence, respectively. Unfortunately, one-hot-encoding grows the dimension of the ML dataset. Therefore, any opportunity to reduce the number of categories in an attribute can alleviate the curse of dimensionality. In this analysis, domain knowledge about the meaning behind the data helped with packaging some of the categories of a few attributes into fewer groups that were also more meaningful.

The performance of many ML algorithms improves when attributes become comparable by normalization, which is to scale them to the same value range. The ML framework uses [0, 1] normalization to make the magnitude of continuous variables comparable with one-hot encoded categorical features. The transformation is

$$\hat{x} = \frac{x - \min x}{\max x - \min x} \tag{1}$$

where $\hat{x}$ is the transformation of attribute vale $x$.

### *3.1.9 Outlier Removal*

Sacrificing a few outlier data points to reduce bias can improve the generalization of a model. Outlier data instances are few and different from the bulk of the dataset [24]. They could represent noisy data entries or rare events that can bias the training of an ML model, resulting in poor predictive performance. Several outlier detection methods are available. The framework used four methods to compare their effect on the model performance:

- One class SVM (OCS) with a radial basis function (RBF) kernel (OCS-RBF)
- Covariance estimator (CE) [25]
- Local outlier factor (LOF) [26]
- Isolation forest (IF) [24]

OCS-RBF applies support vector machine (SVM) classification to assess the similarity of a data instance to the core class. Consequently, OCS-RBF performs well on data that is not Gaussian distributed because it does not assume normally distribute attributes. The CE method fits ellipsis to clusters with central points to identify data instances that are far away, based on the Mahalanobis distance measure. However, CE requires data with a Gaussian distribution. LOF measures the local density of a data instance with respect to the local densities of its k-nearest neighbors. A large deviation of the two local densities indicates that the data instance is isolated. LOF works well with moderately high-dimension datasets because the distance computation scales linearly. The method of IF uses the random forest classification algorithm to detect data instances that are different from most of the data [24]. Outlier removal occurs after one-hot-encoding because some of the algorithms utilize ML methods that work with numerical variables only.

## 3.2 Machine Learning

Many different types of ML models emerged over the years, and each tend to behave differently on different types of datasets [27]. The next subsections describe the different types of models and their hyperparameter tuning to optimize performance on the FRA dataset.

### *3.2.1 Supervised Classification Models*

Table 2 summarizes the 11 different types of ML models used in this analysis. The table provides a brief description of how each algorithm works, their most important hyperparameters (HP), their overall advantages (A) and disadvantages (D). The table groups the models into four broader categories based on their underlying theory of operation: tree-based methods, statistical models, decision boundaries, and learned functions. Numerous excellent books describe the mathematics and theory of operations for each model; they are incorporated here by reference. Géron (2017) discusses both the theory and practical implementation of decision tree (DT), random forest (RF), AdaBoost (AB), logistic regression (LR), support vector machine (SVM), stochastic gradient descent (SGD), and artificial neural network methods [28]. Jame et al. (2013) discusses both the theory and practical implementation of Naïve Bayes (NB), k-nearest-neighbors (kNN), and tree-based boosting methods [27]. Hastie et al. (2016) provides similar coverage for all the models used in this analysis, including some key ML concepts such as bootstrapping, boosting, bagging, and ensemble learning [29]. Murphy (2012) covers the various methods from a more theoretical and probabilistic perspective [20].

Table 2: ML Models Compared

| Type | Model | Algorithm & Hyperparameters | Advantages and Disadvantages |
|---|---|---|---|
| Tree-Based Methods | Decision Tree (DT) | Recursive tree node splitting to maximize the purity of sub-trees. HP: Minimum number of instances in leaves (N), and minimum size of subsets (S). | A: Simple to interpret and to visualize. Works with non-numerical categorical attributes. D: Tends to overfit, resulting in low predictive power on new data. |
| | Random Forest (RF) | Build many full trees for voting. Each tree grows from a bootstrapped dataset and a random subset of attributes. HP: Number of trees (*N*) and minimum size of subsets (S). | A: Offers the simplicity and intuition of decision trees but with less tendency to overfit, therefore, improves generalization on unseen data. D: Incomplete trees diminish insights that full trees might otherwise provide. |
| | AdaBoost (AB) | Sequentially build improved shallow trees for voting. HP: Number of estimators (*N*), learning rate (*R*), boosting algorithm, and regression loss function. | A: Selects only those features that improve predictive power, hence, reducing the computational burden for datasets with very large dimensionality. Less sensitive to overfitting. D: Sensitive to the presence of outliers and data with high incoherence. |
| | Extreme Gradient Boost (XGB) | A highly configurable version of gradient boosting. HP: Number of estimators (N), learning rate (*R*), maximum tree depth (*S*), loss function. | A: Improved performance over gradient boosting and more efficient. D: Sensitive to hyperparameter selection; requires manual intervention to achieve the best configuration for a given dataset. |
| | Gradient Boost (GB) | Sequentially build improved models that fit the errors of previous models. HP: Number of estimators (N), learning rate (*R*), maximum tree depth (*S*), loss function. | A: Efficient and good performance on large datasets; inherently supports missing values. D: Sensitive to hyperparameter selection but has fewer to tune than extreme gradient boosting. |
| Statistical Models | *k*-Nearest Neighbors (*k*-NN) | Determine the class of an instance based on the majority class of its k nearest neighbors. HP: Number of neighbors (*k*), distance method. | A: Method simplicity. D: Sensitive to a skewed class distribution. The computational intensity grows exponentially with the number of instances and attributes. |
| | Naïve Bayes (NB) | Applies Bayes theorem to determine the class probability, given probabilities of the observations. HP: None | A: Fast and simple method. D: Poor performance when attributes are not independent. |
| Decision Boundaries | Logistic Regression (LR) | Establish a decision boundary by using a logistic function to maximally separate classes. HP: Regularization function and strength (C), and probability threshold. | A: Inherits many of the advantages of linear regression; precisions are easy to make. D: Sensitive to noise in the data such as outliers and incorrectly classified instances. Model fitting may fail to converge if there are many highly correlated features. |
| | Support Vector Machine (SVM) | Establish a decision boundary by finding a multidimensional hyperplane to maximally separate classes. HP: Kernel type, cost (C), and regression loss (ε) | A: High accuracy with low computational complexity. D: Sensitive to noisy data and multidimensional planes that lack clear boundaries. |
| Learned Functions | Stochastic Gradient Descent (SGD) | An optimization technique that fits a linear multivariate function to the data. It works best when all features are scaled. HP: Loss function, learning rate method and parameters. | A: An efficient technique on large datasets.<br>D: Sensitive to feature scaling; many hyperparameters; and the true minima may not be achieved because the gradient is only an approximation. |
| | Artificial Neural Network (ANN) | A weighted multilayer linear network that represents a function. HP: Hidden layer neurons (N), solver type, regularization parameter (α), number of iterations (I). | A: Accuracy improves with use and feedback about classification accuracy. D: Requires many training examples to improve classification accuracy. |

### *3.2.2 Hyperparameter Tuning*

Each model requires that the user select values for key parameters (hyperparameters) that affect their performance. Tuning hyperparameters require incremental adjustments while observing a performance metric. The optimization loop uses *k-fold cross validation* to maximize the model *generalization* on the entire dataset while reducing any tendency towards *overfitting* or *underfitting*. Models that have *regularization* parameters provide a means to balance the unavoidable tradeoff between *bias* and *variance,* which improves generalization on unseen data. James et al. (2013) provides an excellent description of the above ML terminologies and concepts, so the book is incorporated here by reference [27].

The performance evaluation metric used was the area under the curve (AUC) of the receiver operating characteristic (ROC). The AUC trends with hyperparameter value adjustments show where each model achieved its best regularized performance. The ROC plots the true positive (TP) rate against the false positive (FP) rate as a function of the class membership probability [30]. Intuitively, AUC measures the power of a model to distinguish among classes in the target attribute. An AUC score of 0.5 indicates that the model has no ability to distinguish among classes of the target whereas a value approaching 1.0 indicates that the model offers a large increase in TP rate for a small price of slightly increasing the FP rate.

The performance evaluation procedure also monitored the classification accuracy (CA), precision (Pc), recall (Rc), and F1 scores. Table 3 describes each metric and summarizes their advantages and disadvantages. All performance metric except the AUC was sensitive to class imbalance in the dataset.

Table 3: Classifier Performance Metric

| Metric | Description | Advantages | Disadvantages |
|---|---|---|---|
| CA | The proportion of predictions that were correct. | Simply calculation. | Sensitive to data imbalance where a no-skill classifier can appear to provide better performance by predicting the dominant class every time. For example, a no-skill classifier will score CA at 90% if the database labels 90% of the accidents as derailments. |
| Pc | The proportion of observations correctly predicted as positives (TP) to the total number of observations predicted as positives (TP + FP). | Measures the probability of mislabeling a negative sample as positive. | A bias towards the majority class can be misleading. |
| Rc | Measures the proportion of positive predictions (TP) to the total number of positive observations (TP + FN) | Measures the probability of correctly labeling all the positive observations. | A bias towards the majority class can be misleading. |
| F1 | The harmonic mean of Pc and Rc, scaled from 0 to 1. | Measures the balance between precision and recall. | Less bias but as a function of Pc and Rc will retain some bias. |
| AUC | Area under the ROC curve that plots TP against FP as a function of class membership probability. | Removes biased scores for imbalanced datasets. | More complex calculation than a simple ratio. Requires the class membership probability for every prediction, which may not be inherently available from a model. |

CA is one of the most often cited performance metric for ML classifiers. However, a high CA score can be misleading if the dataset has high class imbalanced. For example, a no-skill algorithm applied to a dataset with only 5% of the instances from one class and the rest from the other class will appear to have a 95% accuracy if it picks the dominant class for every prediction. Stratified sampling of both the training and testing datasets helps to reduce the imbalance [31].

### 3.3 Feature Ranking

Attributes that contain noisy, irrelevant, or redundant information can diminish the performance of ML methods [32]. Hence, data scientists developed various methods to score features based on the amount of information they contribute towards distinguishing the target classes. This section compares five methods that rank features based on the strength of their association with the classes in the target attribute. Table 11 provides a short description of each method and a reference that provides details about their theory of operations. All methods work best with normalized attributes because their magnitudes become comparable. The diversity of methods result in some compensating for the weaknesses of the other; therefore, they do not provide identical rankings [33]. However, a strong correlation among rankings indicates that the top-ranking attributes do contribute most towards ML classification performance.

Table 11. Feature Ranking by Scoring Methods

| Method | Description | Reference |
|---|---|---|
| ANOVA | Analysis of Variance (ANOVA) measures the difference between average values of a feature in different classes of the target, based on the F distribution. | Agresti (2018) [34] |
| Chi-Squared | Measures a dependency or association between the feature and the target class by using a chi-square statistic. | Wang et al. (2010) [33] |
| Information Gain | The expected amount of entropy reduction. A decrease in entropy (uncertainty) based on the presence of other features will increase information. | Yu and Liu (2003) [32] |
| Gain Ratio | Reduces the bias of Information Gain towards features that have many values by taking the ratio of Information Gain to the intrinsic information (entropy) of the feature. | Quinlan (1986) [35] |
| Gini Decrease | A measure of the inequality among values of a frequency distribution based on their statistical dispersion. A value of zero and one represents perfect equality and inequality, respectively, of the distribution of a feature within each target class. | Han et al. (2016) [36] |

## 3.4 Principle Component Analysis

The method of principle component analysis (PCA) creates a set of new orthogonal basis vectors, each maximally spanning the dimensions of feature space, in the order of the data variance [37]. Each principle component (PC) is a linear combination of all *numerical* features in the dataset. Intuitively, the first two principle components form a plane in feature space that is *closest* to all the data instances, as measured by the Euclidean distance. Data clusters tend to form along the directions of maximum variance. Hence, attributes that most influence the formation of data clusters contribute to inherent structure in the data. The terminology used in the literature is that each PC "explains" some proportion of the total variance (information) in the dataset. Features that are weak components of most PCs tend to be associated with noise in the data. The framework evaluates the trend in proportion of the variance that each PC *explains* to provide insights about the amount and location of noise in the dataset.

# 4 Results

The subsections of this section mirror those of the methodology to organize the presentation of the results from applying the analytical framework described previously. The main procedures are data preparation, machine learning, attribute ranking, and PCA.

## 4.1 Data Preparation

The following subsections describe the data filtering, feature engineering, data imputation, geospatial data cleaning, data extraction, attribute transformation, feature selection, and outlier removal.

### *4.1.1 Data Filtering*

Some of the data schema became inconsistent after the FRA changed reporting requirements for a few of the fields starting June 1, 2011. For example, the report added a field to indicate if the accident occurred in a signalized territory. Hence, there was no entry for the “SIGNAL” field prior to the switchover date. Similarly, a field indicating the method of operation (“MOPERA”) replaced the “METHOD” field that encoded similar information. Consequently, 22% and 79% of the data was missing in the “MOPERA” and “METHOD” fields, respectively. The accident reporting form also added a field “SSB1” to indicate if the track was a continuously welded (CWR) or other. Hence, the “SSB1” field was mostly empty prior to June 1, 2011. Merging 8,055 records from 2009 to 2011 with 21,242 records from 2012 to June 2020 produced a total of 29,297 records with 145 attributes. Table 4 chronicles each criterion used to reduce the number of fields from 145 to 52.

Table 4: Chronicle of Dimension Reduction for 29,297 Records

| Criteria | Attributes Removed | Count |
|---|---|---|
| Sparsity | 19 with > 85% missing data (e.g. DUMMY1-DUMMY7). | 145 - 19 = 126 |
| Duplication | 8 with duplicated information (e.g. IMO, IYR, MONTH, YEAR). | 126 - 8 = 118 |
| Sparsity | 16 with > 90% zero-filled (e.g. CABOOSE1, EVACATE, MIDREM1) | 118 - 16 = 102 |
| Correlated | 12 with > 90% correlation with other attributes (e.g. PASSINJ, PASSKLD) | 102 - 12 = 90 |
| Redundancy | 7 that were redundant with others (e.g. CNTYCD, STATE, COUNTY) | 90 - 7 = 83 |
| Noise | 6 with no relevance to the target (e.g. train number, car number) | 83 - 6 = 77 |
| Noise | 6 with > 20% missing or no relevance to the target (e.g. ADJUNCT1, DIV) | 77 - 6 = 71 |
| Combinable | HUMANS = (engineers + firemen + conductor + brakemen), drop 4. | 71 – 4 + 1 = 68 |
| Correlated | EQATT (equipment attended) correlates with HUMANS, drop 1 | 68 – 1 = 67 |
| Combinable | Combine 15 narrative fields into a single field (NARR), drop original 15. | 67 – 15 + 1 = 53 |
| Combinable | Fill missing MOPERA (method of operation) data with METHOD, drop 1. | 53 – 1 = 52 |

### *4.1.2 Feature Engineering*

Several categorical attributes contained labels that resulted in fewer stratifications when combined. For example, the type of consist (“TYPEQ”) contained 14 different labels to describe sub-categories of the following 6 equipment categories: a “Freight” train, any type of “Passenger” train, any type of “Locomotive,” any set of cars without locomotives (“Cars”), any type of equipment used for maintenance and other non-revenue service work (“Work”), and any type of equipment used to manage yard movements (“Yard”). Similarly, MOPERA (method of operation or movement authorization) contained 21 categories that were simplified into the following 5 broader categories of movement authorization: signaling (“Signal”), direct control (“Control”), restricted limit of movements (“Restrict”), block control for track segments (“Blocks”), and other types of movement authorization (“Not Main”) that were not on the main tracks.

The feature engineering procedure converted track class (“TRKCLAS”) to an ordinal attribute because it encodes speed limits. The FRA track class designation increases the speed limit for both freight and passenger trains in a non-linearly manner from Class 1 (10 mph) through Class 9 (200 mph), which the ordinal encoding from 1 through 9 reflected. The FRA track class designation of “X” for “excepted” has a speed limit of 10 miles-per-hour (mph) but excludes the exclusion of passenger trains. Therefore, an ordinal value of 0 replaced the “X” track class.

The railroad class (“CLASS_RR”) attribute encoded inconsistent values for “Class 1” where some labels were “1” and others were “1L” so the cleaning procedure ensured that attribute values ranged from 1 to 6. The railroad class is a ranking based on their annual operating revenue [1]. Therefore, the procedure re-interpreted the railroad class as an ordinal attribute. Table 5 summarizes the results of the feature engineering procedure.

Table 5: Summary of Feature Engineering

| **Attribute** | **Procedure** |
|---|---|
| CWR | Renamed SSB1 to CWR (continuously welded rail); binarized as “1” = “CWR” and “0” otherwise. |
| LOADED1 | Binarized as “1” = “Y” (first involved car loaded?) and “0” = “N” for non-empty values. |
| WEATHER | Recoded nominal values in WEATHER as labels {Clear, Cloudy, Rain, Fog, Sleet, Snow} |
| TRK_TYP | Renamed TYPTRK (track type) and labeled nominal codes as {Main, Yard, Siding, Industry} |
| VISION | Renamed VISIBLTY and replaced nominal codes as descriptive {Dawn, Day, Dusk, Dark} |
| CLASS_RR | Renamed TYPRR (railroad class) and cleaned to contain only values from 1 to 6. |
| CLASS_TRK | Renamed TRKCLAS (track class) and cleaned to contain ordinal values from 0 to 9 (X → 0) |
| CONSIST | Renamed TYPEQ (consist type); repackaged as {freight, passenger, locomotive, cars, work, yard}.<br>{1} → “Freight”, {2, 3, B, C} → “Passenger”, {8, D, E} → “Locomotive”,<br>{5, 6} → “Cars”, {4, 9, A} → “Work”, {7} → “Yard” |
| ACC_TYPE | Renamed TYPE (accident type); repackaged as category labels:<br>{1} → “Derail”, {2, 3, 6} → “Collide”, {4} → “Collide (Side)”, {5} → “Collide (Rake)”,<br>{7, 8} → “RGC”, {9} → “Obstruct”, {10, 11} → “Fire”, {12, 13} → “Other” |
| MOVEx | Renamed MOPERA; repackaged as labels {signal, control, restrict, blocks, not main}<br>{1, D} → “Signal”, {2, A, B, C, P} → “Control”, {3, L, M, I} → “Restrict”,<br>{4, E, F, G, H, J, K} → “Blocks”, {5, N, O} → “Not Main” |

### *4.1.3 Data Imputation*

Table 6 summarizes the results of the imputing missing values and the impact of each method.

Table 6: Summary of Data Imputation

| Attribute | Missing Before | Missing After | Procedure (N = 29,297, V = 49, M = 3) |
|---|---|---|---|
| TRK_DEN | 51% (15,176) | 0% (0) | Pivot STATION by TRK_TYP, aggregated as maximum TRK_DNSTY (track density). Fill missing data associated with the track type if defined, otherwise use the maximum value. |
| SIG | 22% (6,473) | 0%, 0 | Pivot STATION by TRK_TYP, aggregated as net count SIGNAL (signalized territory). Fill missing data as “1” if net count associated with the track type is greater than 0, otherwise fill with “0” |
| CONSIST | 39% (11,537) | 8% (2,605) | Layer 1: Fill missing CONSIST with:<br>“Freight” if (LOADF1 + EMPTYF1) > 0 otherwise<br>“Passenger” if (LOADP1 + EMPTYP1) > 0 or PASSTRN is “Y” |
|  | 8% (2,605) | 2% (844) | Layer 2: Fill missing CONSIST with:<br>“Freight” if CLASS_RR is “1” (except “Amtrak”) otherwise<br>“Passenger” if RAILROAD (reporting railroad) is “Amtrak” |
|  | 2% (844) | 1% (377) | Layer 3: Fill missing CONSIST with:<br>“Work Train” if TRK_TYP is not “Main” |
|  | 1% (377) | 0% (0) | Layer 4: Fill missing CONSIST with:<br>“Work Train” if TONS (gross tons, excluding locomotives) is 0 otherwise fill missing CONSIST with “Freight” if TONS > 0 |
| CWR | 21% (6,378) | 0% (0) | Fill missing values with “1” if TRK_TYP is “main” and “0” otherwise. |
| MOVEx | 0% (518) | 0% (0) | Fill missing MOVEx based on SIGNAL or TRK_TYP. |
| PASSTRN | 6%, (2,049) | 0% (0) | Fill missing PASSTRN based on CONSIST. Check original flag for consistency with the type CONSIST and the sum of freight and passenger cars (loaded or empty). Flip the flag accordingly. |
| CLASS_RR | 0%, (37) | 0% (0) | Fill missing CLASS_RR (railroad class) by internet search:<br>BLF → 2, {DD, METC} → 3, CN → 1 |
| TRK_TYP | 0%, (15) | 0% (0) | Fill missing TRK_TYP (track type) by inference from the metadata. |
| CLASS_TRK | 0%, (25) | 0% (0) | Fill missing CLASS_TRK (track class) by inference from the metadata. |

For track density (TRK_DEN), the LAP method used the nearest station (STATION) as the location attribute and track type (TRK_TYP) as the sub-location attribute. There were 4,722 unique station names that served as keys for data merging. For the signal (“SIG”) attribute, the LAP method counted the net presence of signalized territories for each track type near the station and assigned “1” if the value was greater than 0 and “0” otherwise. That is, the LAP method voted for the likelihood that the territory near the station used signaling to control movements.

The imputation technique for the type of “CONSIST” attribute used four layers of rule-based inference to fill in missing values. The first layer inferred freight or passenger consist based on the number of freight and passenger cars, respectively, resulting in a reduction of missing values from 39% to only 8%. The next layer inferred freight or passenger consist based on the railroad class, resulting in a further reduction of missing values to only 2%. The next two layers imputed the remaining missing values by inferring the type of consist from the railroad class, the tonnage hauled, and the track type.

Imputing missing values for the type of rail ("CWR") used the probability that "main" track types were continuously welded. A distribution of track type by CWR revealed that "main" track types were more likely to be CWR than other track types. The probabilistic inference method also filled the remaining movement type ("MOVEx") and flag for passenger train ("PASSTRN"). Evaluation of the metadata and an internet search filled the few remaining missing values for track type and track class. Finally, there were no missing values.

#### *4.1.4 Geospatial Cleaning*

Table 7 chronicles the progress of filling missing geospatial coordinates in each step of the procedure. The LAP method filled missing values with the mean value of the non-zero latitude and longitude values for that track type near the station, otherwise the method used the maximum value. Subsequently a GIS spatial join revealed that 21.8% of the records had erroneous geospatial coordinates because their locations on the map did not match the counties reported for the accidents. Hence, the procedure replaced their geospatial coordinates with those of the centroid for the FRA recorded county. There were a few missing county codes that the procedure could not merge, so an internet search filled those missing values based on the station name.

Table 7: Chronicle of Geospatial Coordinate Cleaning

| Attribute | Missing Before | Missing After | Procedure (N = 29,297 records) |
|---|---|---|---|
| Latitude | 21% (6,415) | 2% (817) | Treat zero-filled values as missing. Pivot STATION by TRK_TYP, aggregated as the average geospatial coordinate. Fill missing data with the mean value associated with the track type if available, otherwise fill with the maximum value. |
| Longitude | 21% (6,415) | 2% (820) | Treat zero-filled values as missing. Pivot STATION by TRK_TYP, aggregated as the average geospatial coordinate. Fill missing data with the mean value associated with the track type if available, otherwise fill with the *minimum* value (Longitude is negative in U.S.) |
| REC_ID | 100% | 0% (0) | Add a record identifier as the row index. V: 49+1=50 M: 3. |
| Latitude<br>Longitude | 2% (817)<br>2% (820) | 0% (6)<br>0% (6) | Merge the FRA records with the TIGER® county shapefile by the FIPS5 code. Retain the geospatial centroid coordinates for each county. Add the state name abbreviation and flag (MATCH) to the attributes. V: 50+2=52. Add the county name and state name strings to the metadata. M: 3+2=5. Fill missing FRA geospatial coordinates with the county centroid coordinates. |
| Latitude<br>Longitude | 0% (6)<br>0% (6) | 0% (0)<br>0% (0) | Manually fill missing geospatial coordinates for counties in Alaska and Florida. |
| FIPS5 | 0% (4) | 0% (0) | Fill in missing FIPS5 codes for "Baltimore" and "Skagway" stations. |
| LAT<br>LON | 0% (0)<br>0% (0) | 0% (0)<br>0% (0) | Rename Latitude to LAT and Longitude to LON after the geospatial cleaning procedure. |

### *4.1.5 Data Extraction*

Table 8 chronicles the reduction of data and attributes after the data extraction process.

Table 8: Chronicle of Data Reduction after Data Extraction

| Attribute | Statistic | Procedure |
|---|---|---|
| ACC_CAT | N: 29,297<br>V: 52+1=53<br>M: 5 | Add accident category:<br>{Track, Equipment, Human, Signal, Miscellaneous}. |
| PASSTRN | N: 26,943 (92%)<br>V: 53-4 = 49<br>M: 5 | Dropped accidents involving passenger type trains.<br>Dropped associated attributes:<br>LOADP1, LOADP2, EMPTYP1, EMPTYP2. |
| DERAILED | N: 26,943 (92%)<br>V: 49+1 = 50<br>M: 5 | Added "Derailed" as the target attribute. |
|  | N: 25,035 | Dropped records where the accident cause was missing, 7% (1908) |
|  | N: 15,088 | Dropped records where human factors were a cause, 39.7% (9947) |
|  | N: 15,087 | Dropped 1 record with a missing value for WEATHER. |

The statistics shown in the table are the number of records (N), number of attributes or variables (V), and number of metadata fields (M). The algorithm used the "PASSTRN" as a flag to drop records of accidents involving only passenger trains. Adding an accident category (“ACC_CAT”) flag helped the data extraction code to drop records of accidents caused by human factors. Adding the target attribute “DERAILED” indicated if the accident was a derailment type or not, and it became the label for supervised ML.

### *4.1.6 Attribute Transformation*

Table 9 chronicles the transformation of attributes and their effect on feature reduction.

Table 9: Chronicle of the Transformed and Derived Attributes

| Attribute | Reduction | Procedure |
|---|---|---|
| HR24 | 50–3+1 = 48 | Combined TIMEHR, TIMEMIN, AMPM to 24-hour continuous, then drop old. |
| TRK_DEN_LG | 48-1+1 = 48 | Log Transform: TRK_DEN, then drop old. |
| TRNSPD_LG | 48-1+1 = 48 | Log Transform: TRNSPD, then drop old. |
| TONS_LG | 48-1+1 = 48 | Log Transform: TONS, then drop old. |
| POS_CAR | 48+1-1 = 48 | Rename and recode POSITON1 (position of first involved car) as the fractional position relative to the number of cars. 0 is front, 1 is back. |
| N_CARS | 48+1 = 49 | Add N_CARS as the sum of loaded and empty cars. |
| CARS_LD | 49+1-4 = 46 | Add CARS_LD as proportion of N_CARS loaded. Drop: LOADF1, EMPTYF1, POSITON1, PASSTRN |
| CARS_HZMT | 46+1-1 = 46 | Add CARS_HZMT as proportion of CARS_LD that carry Hazmat. Drop CARS (number of cars carrying hazmat) |
| SPD_OVR | 46+1–1 = 46 | Add to capture difference in train speed and speed limit for CLASS_TRK. Dropped field HIGHSPD. |
| Metadata | 46-6= 40 | Converted 6 attributes (REC_ID, SC, STATION, RAILROAD, RR3, IYR) to metadata: 5+6=11. |

The procedure combined the three attributes related to time into a single attribute (HR24) that represented the hour as a continuous value within the range [0, 24). The combined attributes were hour (“TIMEHR”), minute (“TIMEMIN”), and AM flag (“AMPM”). The shifted log transformations reduced the skew of the track density (“TRK_DEN”), train speed (“TRNSPD”), and tonnage hauled (“TONS”) attributes. The three proportional transformations were relative to the number of cars (N_CARS), derived from the sum of loaded and empty cars. The “SPD_OVR” attribute was the excess train speed relative to the speed limit for the track class operated on. Hence, the value was negative for trains that were operating below the speed limit. Finally, the transformation procedure identified six attributes as irrelevant to the target and converted them to metadata. Examples were the state code (“SC”), station name (“STATION”), railroad name (“RAILROAD”), track maintenance organization (“RR3”), and the incident year (“IYR”).

### 4.1.7 Feature Selection

Table 10 chronicles the feature reduction after eliminating post-event attributes. Table 11 summarizes the final set of 25 attributes used to build the ML models. The ML did not use the 11 metadata fields, but they supported further descriptive analysis. One-hot-encoding the categorical attributes increased the number of features from 25 to 51. The dispersion indicates the amount of variability in the distribution of each attribute. The dispersion measure is the *entropy* and coefficient of variation (CV) for categorical and numerical attributes, respectively. The entropy of an attribute is

$$H(X) = -\sum_{i=1}^{N} P(x_i) \log P(x_i) \tag{2}$$

where $x_i$ is the $i$th category value and $P(x_i)$ is a probability estimate based on their frequency of occurrence in the dataset. For numerical attributes, the CV was the ratio of the standard deviation to the mean value.

Table 10: Chronicle of the Eliminated Attributes

| Attribute | Reduction | Process |
|---|---|---|
| POSCAR | 40-1 = 39 | Relative position of the first involved car in the train. |
| LOADED_1 | 39-1 = 38 | Boolean: Is first involved car loaded? Missing (22%, 6568) |
| ACCDMG | 38-1 = 37 | Total reported damage in U.S. dollars. |
| CASKLD | 37-1 = 36 | Total killed for all involved railroads. |
| CASINJ | 36-1 = 35 | Total injured for all involved railroads. |
| CARSHZD | 35-1 = 34 | Number of cars that released hazardous materials. |
| CARSDMG | 34-1 = 33 | Number of cars damaged or derailed. |
| POSITON2 | 33-1 = 32 | Position of car on the train that caused the accident. |
| EMPTYF2 | 32-1 = 31 | Number of empty freight cars that derailed. |
| LOADF2 | 31-1 = 30 | Number of loaded freight cars that derailed. |
| HEADEND2 | 30-1 = 29 | Number of headend locomotives that derailed. |
| ACC_TYPE | 29-1 = 28 | Type of accident. Missing (0%, 83). |
| ACC_CAT | 28-1 = 27 | Accident cause category. |
| CAUSE | 27-1 = 26 | Accident cause code. |
| MATCH | 26-1 = 25 | Temporary geospatial filter flag for county mismatch. |

Table 11: Summary of the ML Attributes, their Dispersion, and Type

| Attribute | Dispersion | Type | Description (N=15,087, V=25, T=1) |
|---|---:|---|---|
| DERAILED | 0.631 | Categorical | Target attribute: 1 if the accident type was derailment. |
| REGION | 0.400 | Categorical | Cleaned FRA region code for accident location. |
| LAT | 0.133 | Continuous | Cleaned latitude coordinate |
| LON | -0.126 | Continuous | Cleaned longitude coordinate |
| CLASS_RR | 0.796 | Ordinal | Cleaned railroad class. |
| MONTH | 0.549 | Ordinal | Incident month. |
| DAY | 0.561 | Ordinal | Incident day. |
| HR24 | 0.541 | Continuous | Transformed time to fractional 24-hour. |
| TEMP | 0.391 | Continuous | Temperature (degrees Fahrenheit) |
| VISION | 1.110 | Categorical | Visibility: {Dawn, Day, Dusk, Dark} |
| WEATHER | 0.977 | Categorical | Weather: {Clear, Cloudy, Rain, Fog, Sleet, Snow} |
| TRK_TYP | 1.050 | Categorical | Track Type: {Main, Yard, Siding, Industry} |
| TRK_CL | 0.753 | Ordinal | Track Class: {X as 0, 1 through 9} |
| CWR | 0.685 | Binary | 1 if the rail type was continuously welded, 0 otherwise. |
| MOVEx | 1.250 | Categorical | Movement: {Blocks, Control, Signal, Not Main, Restrict} |
| TRK_DEN_LG | 0.972 | Continuous | log(1+x) of annual track density in millions of gross tons. |
| SIG | 0.590 | Binary | 1 if used signals to control train movements, 0 otherwise. |
| TRNSPD_LG | 0.589 | Continuous | log(1+x) of train speed in miles per hour (mph). |
| SPD_OVR | -1.304 | Continuous | Difference between train speed and limit for track class. |
| CONSIST | 0.950 | Categorical | Consist: {Freight, Locomotive, Cars, Work, Yard} |
| TONS_LG | 0.757 | Continuous | log(1+x) of gross tonnage, excluding power units. |
| LOCOS | 0.704 | Ordinal | Number of headend locomotives. |
| N_CARS | 0.915 | Ordinal | Total number of cars. |
| CARS_LD | 0.704 | Continuous | Proportion of the number of cars that were loaded (0 to 1) |
| CARS_HZMT | 2.800 | Continuous | Proportion of loaded cars carrying hazardous materials (0 to 1) |
| HUMANS | 0.562 | Continuous | Number of humans present on the train. |

### *4.1.8 Outlier Removal*

Table 12 summarizes the AUC performance metric for a random forest classifier after removing outliers using each of the four methods, with the various hyperparameter selections shown. All algorithm and parameter selection produced similar performance. The framework used the LOF algorithm with 20 nearest neighbors and 1% outliers because of its slight AUC performance edge. The method removed 126 outliers to result in 15,087 – 126 = 14,961 records used to train and evaluate the ML models.

Table 12: Outlier Algorithm Performance Evaluation

| **Algorithm** | **Hyperparameters** | **AUC** |
|---|---|---|
| One class SVM | Nu: 1%, Kernel Coefficient: 0.01 | 0.881 |
| One class SVM | Nu: 1%, Kernel Coefficient: 0.1 | 0.878 |
| One class SVM | Nu: 10%, Kernel Coefficient: 0.01 | 0.879 |
| Local Outlier Factor | C: 1%, Neighbors: 10, Euclidean | 0.879 |
| Local Outlier Factor | C: 1%, Neighbors: 20, Euclidean | 0.882 |
| Local Outlier Factor | C: 1%, Neighbors: 50, Euclidean | 0.880 |
| Isolation Forest | C: 0% | 0.881 |
| Isolation Forest | C: 1% | 0.880 |
| Isolation Forest | C: 5% | 0.880 |
| Covariance Estimator | C: 1% | 0.817 |

## 4.2 Machine Learning

Table 13 summarizes the stabilized performance of each ML algorithm, sorted by the AUC metric. The null model is a no-skill model that predicts the dominant class each time. It provided a baseline to compare the performance score of skilled classifiers. As expected, the CA score for the no-skill classifier reflected the class imbalance of 67.42% for derailment type accidents versus non-derailment type accidents. However, the AUC performance of the null classifier was lowest as expected.

Tracking the AUC trend with 10-fold cross validation and stratified sampling produced the optimum hyperparameter values shown in the table. Hyperparameters with common names across some models were the learning rate (L), loss function (LF), regularization (R) parameters, and optimizer algorithm (OA).

Table 13: Model Performance and Optimum Hyperparameter Settings

| Model | AUC | CA | F1 | PR | RC | Optimum Hyperparameters |
|---|---|---|---|---|---|---|
| XGB | 0.888 | 0.828 | 0.875 | 0.859 | 0.892 | $\gamma$:0, Max Depth: 6, Min Child Weight: 1, R:1, w:1, L:0.2, |
| GB | 0.884 | 0.824 | 0.872 | 0.854 | 0.891 | LF: LR, Trees (*N*): 100, L: 0.2, Min Samples Leaf: 1 |
| RF | 0.882 | 0.821 | 0.817 | 0.817 | 0.821 | Trees (*N*): 60, Attributes/Split: 5, Min Subset: 5 |
| DT | 0.854 | 0.803 | 0.801 | 0.800 | 0.803 | Max Depth: 10, Min Samples Leaf (*N*): 90, Min Subset: 5 |
| ANN | 0.838 | 0.786 | 0.785 | 0.784 | 0.786 | Hidden Nodes: 100, Activation: ReLu, OA: Adam ($\alpha$:$10^{-4}$) |
| LR | 0.832 | 0.783 | 0.777 | 0.777 | 0.783 | R (L2, C:5) |
| SGD | 0.828 | 0.783 | 0.776 | 0.776 | 0.783 | LF: (LR, $\varepsilon$:1), R: E.Net ($\alpha$:$10^{-5}$, 0.15), L: IVS ($\eta_0$:$10^{-2}$, *t*:0.25) |
| kNN | 0.803 | 0.765 | 0.759 | 0.758 | 0.764 | *N*: 30, Distance (Euclidean, Weights: Uniform) |
| NB | 0.794 | 0.725 | 0.730 | 0.740 | 0.725 | No parameters to tune |
| ADB | 0.713 | 0.746 | 0.746 | 0.747 | 0.746 | Trees (*N*): 50, LF: Linear, OA: SAMME.R, LR: 1.0 |
| SVM | 0.626 | 0.654 | 0.639 | 0.633 | 0.654 | Kernel: Sigmoid, R (C:0.2, $\varepsilon$:1.0) |
| Null | 0.500 | 0.674 | 0.543 | 0.455 | 0.674 | No parameters to tune |

To demonstrate the effect of hyperparameter tuning, Figure 4 plots the AUC score for a range of hyperparameter *N* associated with RF, kNN, and DT.

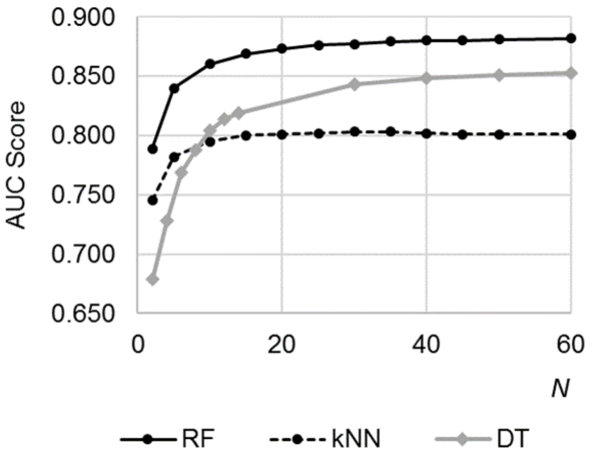

Figure 4: AUC score as a function of hyperparameter *N*.

As noted in Table 13, the hyperparameter *N* represents the number of trees of a RF, the minimum number of samples to retain in the leaves of a DT, and the number of nearest neighbors for the kNN algorithm. The asymptotic trend was similar for all hyperparameters tuned.

### 4.3 Feature Ranking

Table 14 shows the importance ranking of the first 30 features in their strength of association with the target class. The rank by each of the five scoring methods are correlated as indicated by their pairwise correlation coefficients listed in Table 15. The correlation ranges from 84.2% for the gini and chi-squared methods to 94.5% for the ANOVA and chi-squared methods.

Table 14: Feature Importance Ranking

| Feature | ANOVA | $\chi^2$ | Info. Gain | Gain Ratio | Gini |
|---|---|---|---|---|---|
| TRK_CL | 1 | 2 | 4 | 3 | 2 |
| MOVEx=Signal | 2 | 3 | 3 | 1 | 4 |
| SPD_OVR | 3 | 1 | 7 | 11 | 3 |
| SIG | 4 | 4 | 5 | 2 | 5 |
| HUMANS | 5 | 7 | 6 | 10 | 6 |
| TRK_TYP=Main | 6 | 5 | 9 | 6 | 7 |
| CWR | 7 | 6 | 1 | 8 | 8 |
| MOVEx=Not Main | 8 | 11 | 11 | 9 | 11 |
| LOCOS | 9 | 9 | 10 | 12 | 9 |
| CONSIST=Cars | 10 | 8 | 14 | 4 | 12 |
| TRK_TYP=Industry | 11 | 10 | 12 | 7 | 14 |
| TRK_TYP=Yard | 12 | 16 | 2 | 18 | 16 |
| TONS_LG | 13 | 14 | 15 | 20 | 17 |
| CARS_LD | 14 | 18 | 13 | 19 | 13 |
| CONSIST=Yard | 15 | 15 | 18 | 17 | 19 |
| N_CARS | 16 | 12 | 28 | 16 | 10 |
| MOVEx=Restrict | 17 | 17 | 26 | 15 | 20 |
| LAT | 18 | 20 | 22 | 32 | 22 |
| TEMP | 19 | 22 | 24 | 30 | 21 |
| TRK_TYP=Siding | 20 | 21 | 25 | 13 | 24 |
| VISION=Dark | 21 | 24 | 21 | 24 | 25 |
| CLASS_RR | 22 | 13 | 30 | 14 | 15 |
| TRK_DEN_LG | 23 | 19 | 20 | 22 | 18 |
| REGION=7.0 | 24 | 23 | 19 | 21 | 26 |
| VISION=Day | 25 | 31 | 29 | 35 | 27 |
| REGION=8.0 | 26 | 26 | 27 | 23 | 28 |
| REGION=6.0 | 27 | 27 | 23 | 28 | 29 |
| REGION=2.0 | 28 | 28 | 33 | 25 | 30 |
| TRNSPD_LG | 29 | 25 | 37 | 5 | 1 |
| REGION=3.0 | 30 | 29 | 31 | 31 | 31 |

Figure 5 shows the probability distribution of derailment and non-derailment type accidents for the top two attributes (track class, movement authorization) and the fourth ranking attribute (signalized territory).

Table 15: Correlation of Ranking Methods

| Method A | Method B | Correlation |
|---|---|---|
| ANOVA | Chi-Squared | 0.945 |
| ANOVA | Info. Gain | 0.897 |
| Gain Ratio | Gini | 0.843 |
| Gini | Chi-Squared | 0.842 |

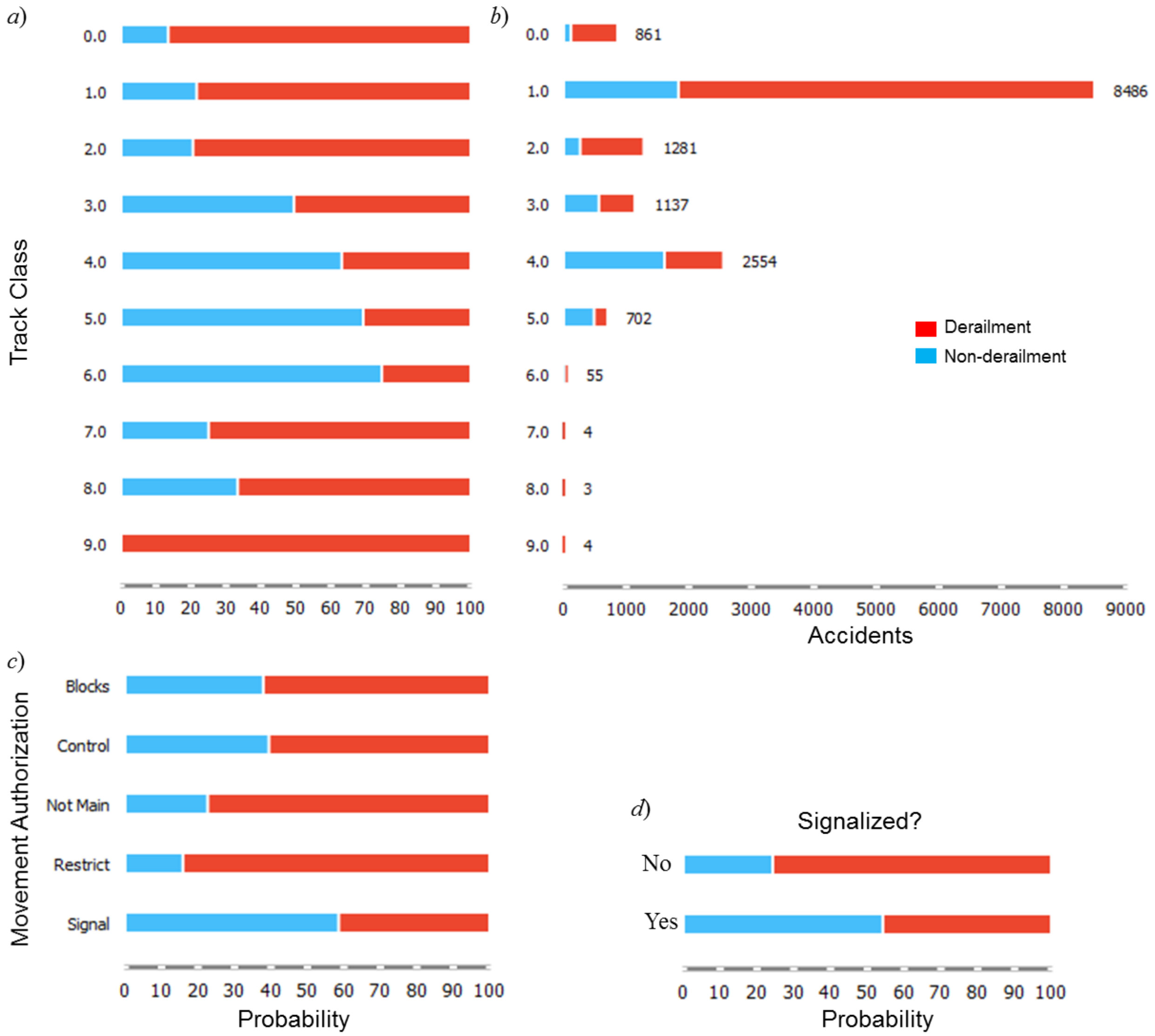

Figure 5: Class probability for the top two and fourth ranking attributes.

The distributions show that these attributes have some power to separate derailment from non-derailment type accidents, but with uncertainty based on the amount of overlap in their class distributions. For example, the class probability was higher for derailment type accidents on class 0, 1, 2, 7, 8, and 9 tracks (Figure 5a). The distinction is significant for class 1 tracks because it has the highest frequency of occurrence (Figure 5b). Similarly, the class probability was higher for derailment type accidents where movement authority was within restricted limits (restricted) or where movement was not on main tracks (Figure 5c). Similarly, the class probability was higher for derailment type accidents in non-signalized territories (Figure 5d). The probability difference was much lower for the lower ranking attributes, but taken together, they improve the ML classification performance.

Figure 6 is a box plot that shows the distribution and statistics of excess speed for derailment and non-derailment type accidents.

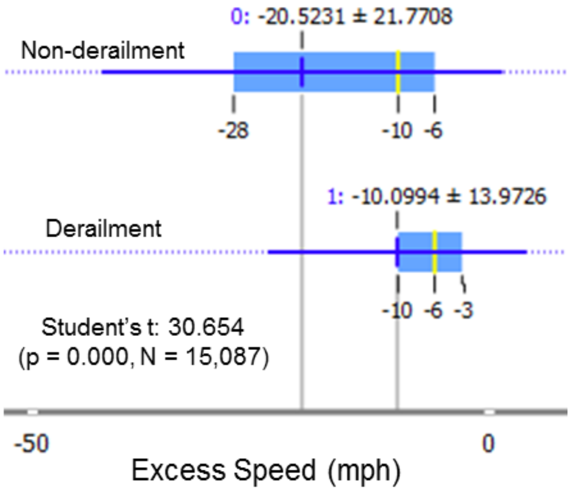

Figure 6: Distribution and statistics for excess speed.

All accidents tended to occur below the speed limit for the track class on which they operated. However, derailment type accidents tended to occur closer to the speed limit than non-derailment type accidents. A student's t-test shows that the p-value was near zero, which indicated that the mean difference of 10 mph (16 kph) was statistically significant. The highlighted boxes in the figure indicates the values of the first quartile (25%) through the third quartile (75%) of the dataset. The solid vertical and horizontal lines indicate the mean and standard deviation, respectively. The lighter solid vertical lines indicate the median values.

## 4.4 Principle Component Analysis

Figure 7 plots the proportion of variance in the data that each PC explained. The top and bottom curves show the cumulative variance and component variance explained, respectively, as a function of each addition PC in their ranked order. This analysis indicated that the first six PCs explained just over half of the variance in the dataset. Each of the remaining 45 of 51 total PCs incrementally explain less than 4% of the variance each, but together account for the remaining half of the variance explained.

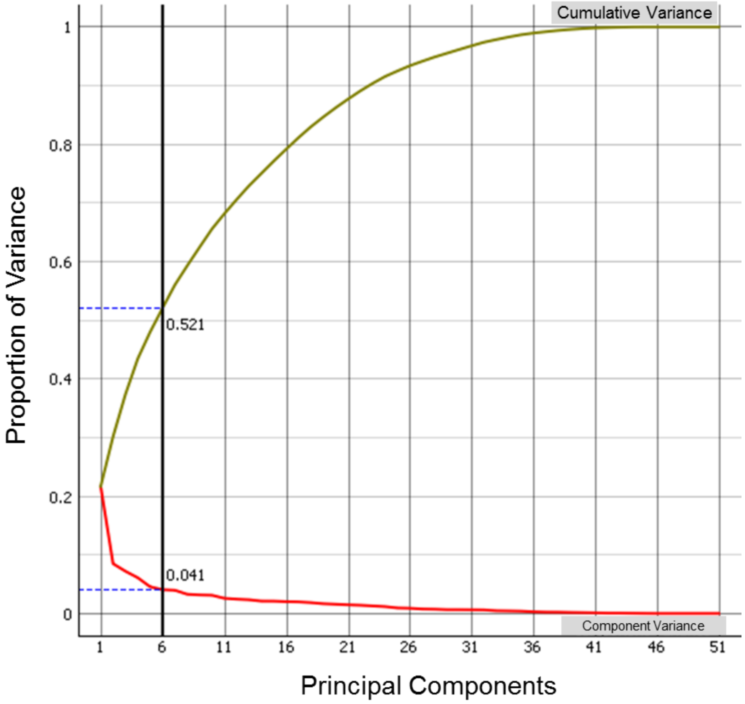

Figure 7: The proportion of variance in the data that each PC explains.

Figure 8 and Figure 9 are visualizations of the PC clusters that suggest structure and noise in the dataset.

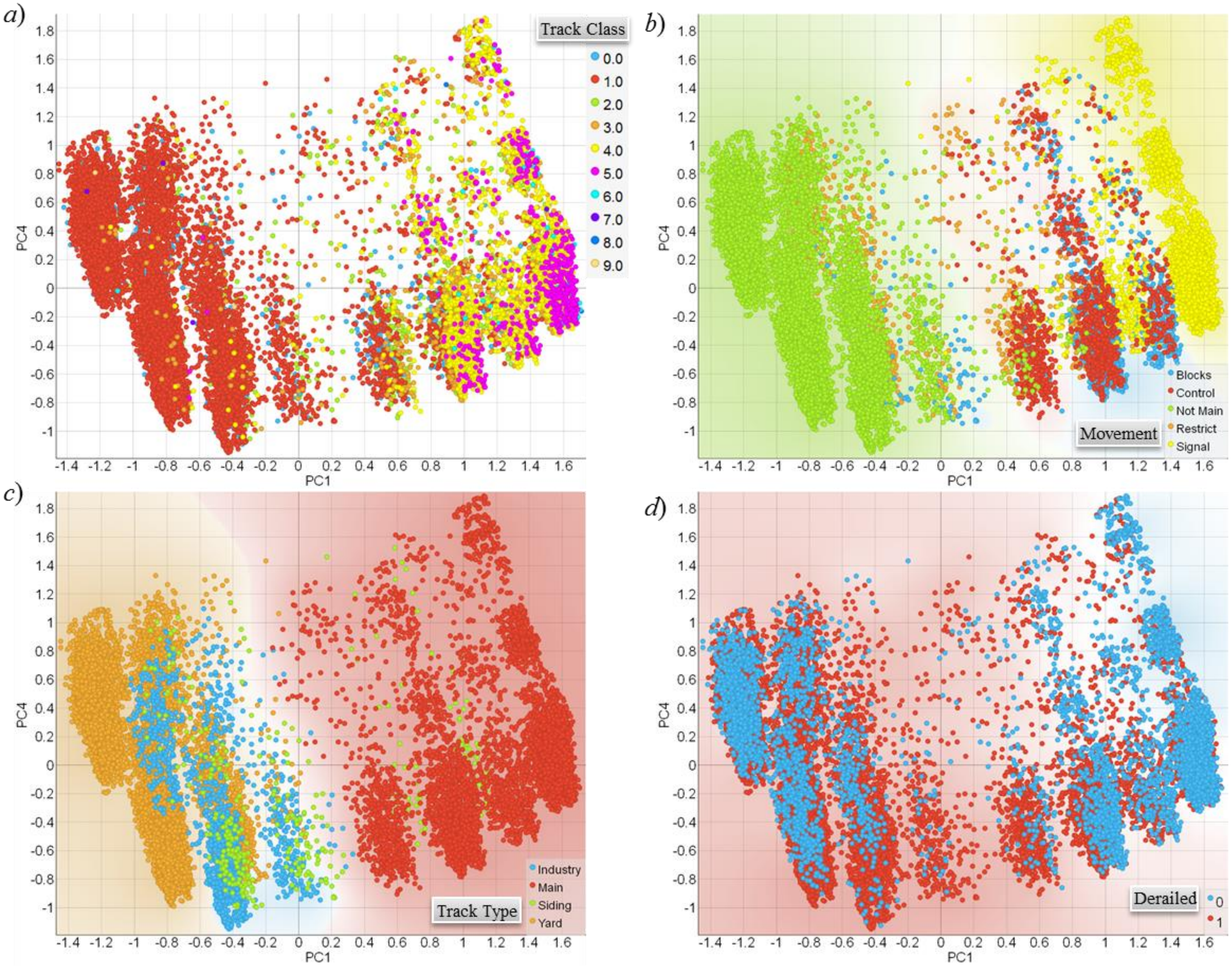

Figure 8: Data clusters for attributes with high power to distinguish among the target classes.

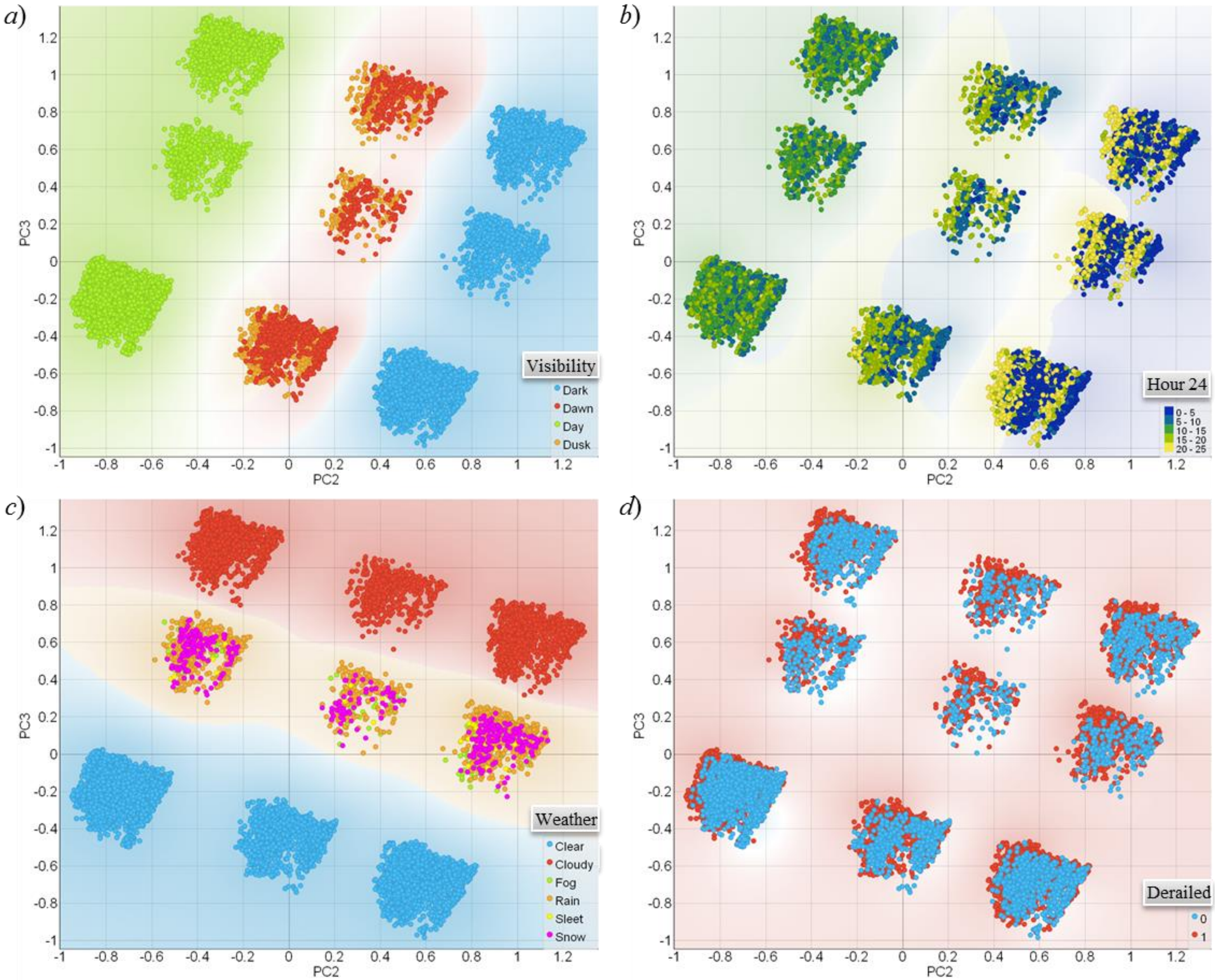

Figure 9: Data clusters for attributes with low power to distinguish among the target classes.

Figure 8 shows that PC1 and PC4 form elongated elliptical clusters for the top ranking attributes of track class (Figure 8a), movement authority (Figure 8b), and track type (Figure 8c). Figure 8d shows the distribution of the target class in the same PC feature space, where the color shading indicates a bias towards the left clusters with negative PC1 values.

Figure 9 shows that PC2 and PC3 form nine distinct clusters for visibility (Figure 9a), hour (Figure 9b), and weather (Figure 9c). Figure 9d shows the distribution of the target class across each cluster. The clusters of the higher-ranking attributes (Figure 8) are less distinct than those of the lower ranking attributes (Figure 9), which is further discussed for interpretation in the next section.

# 5 Discussion

The overall good performance of the top four ML methods supported the effectiveness of the custom data cleaning procedures, including the LAP technique introduced for imputing missing values. The LAP method was most effective in filling missing values for track density, but that attribute ranked low in importance for classification. Although effective, one limitation of the LAP technique is that it provided a course imputation of the geospatial coordinates, based on an aggregation of entries from other records where a value was present for the track type near that station. However, in lieu of an alternative, the LAP imputed values to enable the operation of all ML methods. The geospatial join method provided the next best alternative to replace erroneous or low-resolution geospatial data. The distinctive southeast skew pattern revealed those records with low-resolution data entry.

The top four algorithms of XGB, GB, RF, and DT were all based on the theories of decision trees. They all achieved an AUC score greater than 85%. The highest AUC score of nearly 89% for XGB was associated with a classification accuracy and balanced precision-recall scores (F1) of nearly 83% and 88%, respectively. All methods were sensitive to hyperparameter tuning as demonstrated in the performance improvement trends of Figure 4. The hyperparameter tuning sensitivity cautions against using the default values suggested for each method.

All feature ranking methods and PCA pointed to track class (TRK_CL), signalized movement authority (MOVEx = Signal), speed excess, and signalized territory (SIG) as the most important features in ML classifier performance. The interpretation of an attribute rank is its relative power to separate the distributions of the categories in the target class. That is, an exceptionally high overlap of the two class distributions ranked the attribute exceptionally low in importance towards classifier performance. It is rare that any one attribute can completely distinguish among class members with 100% accuracy, otherwise there would be no need to use additional attributes as explanatory factors for classification. Rather, a combination of attributes contributes their ability to help determine the probability of class membership. Poor classification results with all types of classification models may indicate that all attributes have a high degree of overlap in their class probability distributions.

The PCA result (Figure 7) shows that the first 6 PCs explain more than half the variance in the dataset but that it takes the remaining PCs, which accounted for 88% of the PCs, to explain the remaining half of the variance in the data set. This outcome indicates that the first six PCs represented the bulk of the information in the dataset. By extension, the remaining PCs likely account for noise in the dataset based on the slow accumulation of the variance they explained. This result suggests that just under half of the variance in the dataset lack structure and, therefore, constitutes the noise in the dataset.

Figure 8 further illustrates structure in the dataset by clusters formed from PC1 and PC4 for the top-ranking features of track class, movement authority, and track type. One can visualize the amount of noise by the amount of attribute contamination of clusters and isolation from clusters. Even though the target class was spread across all clusters (Figure 8d) there was an observable bias of derailment type accidents towards clusters on the left. The bias corresponds to clusters of class 1 tracks (Figure 8a), movement authorities not on the mainline (Figure 8b), and non-main track types (Figure 8c). This result suggests that features that align with the cluster where the derailment class is biased associates more with derailment than non-derailment type accidents.

Figure 9 shows that PC2 and PC3 form clusters for the attributes of visibility (Figure 9a), hour (Figure 9b), and weather (Figure 9c), which are low-ranking. The even distribution of each target class across each cluster (Figure 9d) agreed with their low importance ranking. Interestingly, the level of isolation noise was much lower for those lower-ranking attributes. The contamination noise in the center column of the cluster grid (Figure 9a) suggest similarities in the visibility at dawn and dusk, as expected. Those similarities also corresponded to the separation of "Hour 24" (Figure 9b) where day, night, and visibility transition times corresponded to the expected hour ranges. The contamination in the center row of clusters of the cluster grid (Figure 9c) suggest similarities in weather conditions like snow, sleet, rain, and fog. Hence, the clustering results were as expected. The low level of isolation noise observed for the clusters of the low-ranking features would have helped the ML performance more had the situation occurred for the clusters of the highest-ranking features.

The above insights about the location of structure and noise in the dataset provided clues to understand the reason for the performance differences of each ML method. Randomized tree-based methods tend to train on various cross-sections of a dataset and use voting to determine the class likelihood. In contrast, the other methods tend to leverage structure in the dataset. Hence, the randomized tree-based methods such as XGB, GB, and RF performed better by discovering patterns across noisy neighborhoods in dataset. On the other hand, kNN seeks local neighborhoods to predict class membership based on attribute similarity. Consequently, noisy neighborhoods can hamper classification performance as evidenced by the low performance rank of kNN. Methods such as SVM and LR seek clear decision boundaries in multidimensional feature space. Hence, the lack of clear hyperplanes between the target classes hampered their performance. In fact, SVM achieved the lowest performance.

One limitation of the railroad accident database is that it does not necessarily list accidents where the financial loss was below $10,500 because the FRA does not require railroads to report those. A second limitation is that the financial loss includes only the costs of repairing equipment, signal systems, and infrastructure structures. Losses do not include costs associated with cleanup, lost freight, societal damages, fatalities, injuries, and line closures. Nevertheless, financial loss was not a pre-incident explanatory variable, but any future analysis that uses it should be aware of this limitation in the dataset.

# 6 Conclusions

Railroads have been one of the most important modes of transport for more than a century. Unfortunately, accidents continue to plague their operating safety and efficiency. Derailments have consistently dominated other accident types and resulted in the greatest financial loss. Therefore, gaining insights into factors that are more strongly associated with derailments than other accident types can inform more cost-effective and impactful risk management strategies.

Recent advancements in computing capacity and their cost reduction has enabled machine learning (ML) methods to uncover patterns in large multidimensional datasets that are difficult to analyze with common rule-based and statistical methods. However, there are many types of ML techniques, and no single method works best for all types of datasets. Therefore, this work applied 11 different types of ML models to a large multidimensional dataset of railroad accidents to compare their performance in predicting derailments from other accident types. The extreme gradient boosting (XGB) classifier provided the best predictive performance with an AUC score of 89%. The model could distinguish accident type with an accuracy of 83%. Principle component analysis (PCA) revealed that high feature contamination noise and isolation noise would prevent significant further gains in classification accuracy by any algorithm.

The good ML performance affirmed the relevance and sufficiency of the attributes in their contribution towards distinguishing derailments from other accident types. Hence, knowing the relative importance of those attributes towards classification accuracy can lead to insights for decision-making in railroad risk management. The importance ranking used five different methods that agreed on the ranking with correlations ranging from 84.2% to 94.5%. The ANOVA and chi-squared methods agreed with the highest correlation that the top four attributes were track class, the type of movement authority, the excess speed, and the presence of signalization in the territory. The feature distribution for each target class and the PCA agreed that relative to non-derailment type accidents, derailments were more strongly associated with lower track classes, non-signalized territories, and movement authorizations with restricted limits. Derailments also tended to occur at 10 mph (16 kph) below the speed limit of the track class whereas non-derailment type accidents tended to occur at 20 mph (32 kph) below the limit.

The good ML performance also suggests that the custom data imputation techniques presented were effective in filling missing values. The data-cleaning framework also demonstrated a spatial join technique that addressed 21.8% of the geospatial data entry errors. The detailed chronicle of the cleaning procedures will help other researchers save a substantial amount of time in data preparation when using the same dataset. Future work will leverage the framework to examine trends in accidents caused by human error to determine the effectiveness of PTC deployments relative to historic accident rates.

## 7 Data Availability

This paper cited the sources of all the data used, which are currently publicly available.

## 8 Conflicts of Interest

The authors declare that there is no conflict of interest about the publication of this article.

## 9 Credit

Raj Bridgelall: conceptualization, methodology, software, data curation, formal analysis, writing—original draft preparation. Denver Tolliver: supervision, resources, funding acquisition, project administration, validation, writing—reviewing and editing.

## 10 Funding Statement

The authors conducted this work with support from North Dakota State University and the Mountain-Plains Consortium, a University Transportation Center funded by the U.S. Department of Transportation.

References

[1] ASCE, "Infrastructure Report Card," American Society of Civil Engineers, Reston, VA, 2017.

[2] R. Bridgelall and D. D. Tolliver, "Closed form models to assess railroad technology investments," *Transportation Planning and Technology,* vol. 43, no. 7, pp. 639-650, 2020.

[3] Z. Zhang, X. Liu and K. Holt, "Positive Train Control (PTC) for railway safety in the United States: Policy developments and critical issues," *Utilities Policy,* vol. 51, no. 2018, pp. 33-40, 2018.

[4] I. F. Ilyas and X. Chu, Data Cleaning, New York, NY: Association for Computing Machinery and Morgan & Claypool Publishers, 2019, p. 282.

[5] M. Z. H. Jesmeen, J. Hossen, S. Sayeed, C. Ho, K. Tawsif, A. Rahman and E. Arif, "A Survey on Cleaning Dirty Data Using Machine Learning Paradigm for Big Data Analytics," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 10, no. 3, pp. 1234-1243, 2018.

[6] R. Bridgelall, P. Lu, D. D. Tolliver and T. Xu, "Mining Connected Vehicle Data for Beneficial Patterns in Dubai Taxi Operations," *Journal of Advanced Transportation,* vol. 2018, pp. 1-8, 2018.

[7] E. Rahm and H. H. Do, "Data Cleaning: Problems and Current Approaches," *IEEE Data(base) Engineering Bulletin,* vol. 23, pp. 3-13, 2000.

[8] A. Iranitalab and A. J. Khattak, "Comparison of four statistical and machine learning methods for crash severity prediction.," *Accident Analysis & Prevention,* vol. 108, pp. 27-36, 2017.

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

[10] E. Dabbour, S. Easa and M. Haider, "Using fixed-parameter and random-parameter ordered regression models to identify significant factors that affect the severity of drivers' injuries in vehicle-train collisions.," *Accident Analysis & Prevention,* vol. 107, pp. 20-30, 2017.

[11] J. Liu and A. J. Khattak, "Gate-violation behavior at highway-rail grade crossings and the consequences: Using geo-Spatial modeling integrated with path analysis.," *Accident Analysis & Prevention,* vol. 109, pp. 99-112, 2017.

[12] A. Keramati, P. Lu, A. Iranitalab, D. Pan and Y. Huang, "A crash severity analysis at highway-rail grade crossings: The random survival forest method.," *Accident Analysis & Prevention,* vol. 144, p. 105683, 2020.

[13] S. Soleimani, S. R. Mousa, J. Codjoe and M. Leitner, "A Comprehensive Railroad-Highway Grade Crossing Consolidation Model: A Machine Learning Approach.," *Accident Analysis & Prevention,* vol. 128, pp. 65-77, 2019.

[14] B. Wali, A. J. Khattak and N. Ahmad, "Injury severity analysis of pedestrian and bicyclist trespassing crashes at non-crossings: A hybrid predictive text analytics and heterogeneity-based statistical modeling approach," *Accident Analysis & Prevention,* vol. 150, p. 105835, 2021.

[15] X. Liu, M. R. Saat and C. P. Barkan, "Freight-train derailment rates for railroad safety and risk analysis.," *Accident Analysis & Prevention,* vol. 98, pp. 1-9, 2017.

[16] B. Z. Wang, C. P. L. Barkan and M. R. Saat, "Quantitative Analysis of Changes in Freight Train Derailment Causes and Rates," *Journal of Transportation Engineering, Part A: Systems,* vol. 146, no. 11, p. 4020127, 2020.

[17] A. Iranitalab and A. J. Khattak, "Probabilistic classification of hazardous materials release events in train incidents and cargo tank truck crashes," *Reliability Engineering & System Safety,* vol. 199, p. 106914, 2020.

[18] H. Li, D. Parikh, Q. He, B. Qian, Z. Li, D. Fang and A. Hampapur, "Improving rail network velocity: A machine learning approach to predictive maintenance," *Transportation Research Part C-emerging Technologies,* vol. 45, pp. 17-26, 2014.

[19] A. Lasisi and N. Attoh-Okine, "Machine Learning Ensembles and Rail Defects Prediction: Multilayer Stacking Methodology," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering,* vol. 5, no. 4, p. 4019016, 2019.

[20] K. P. Murphy, Machine Learning : A Probabilistic Perspective, Cambridge, Massachusetts: The MIT Press, 2012.

[21] N. Z. Abidin, A. R. Ismail and N. A. Emran, "Performance Analysis Of Machine Learning Algorithms For Missing Value Imputation," *International Journal of Advanced Computer Science and Applications,* vol. 9, no. 6, 2018.

[22] FRA, "Rail Equipment Accident/Incident Data File Structure and Field Input Specifications," Federal Railroad Administration (FRA), Washington, D.C., 2011.

[23] W. G. Manning and J. Mullahy, "Estimating log models: to transform or not to transform?," *Journal of Health Economics,* vol. 20, no. 4, pp. 461-494, 2001.

[24] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation-Based Anomaly Detection," *ACM Transactions on Knowledge Discovery From Data,* vol. 6, no. 1, p. 3, 2012.

[25] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics,* vol. 41, no. 3, pp. 212-223, 1999.

[26] M. M. Breunig, H.-P. Kriegel, R. T. Ng and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000.

[27] G. James, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning with Applications in R, vol. 112, New York: Springer, 2013.

[28] A. Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed., Sebastopol, California: O'Reilly Media, 2017, p. 856.

[29] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and

Prediction, 2nd ed., New York, New York: Springer, 2016, p. 767.

[30] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861-874, 2006.

[31] B. Krawczyk, "Learning from Imbalanced Data: Open Challenges and Future Directions," *Progress in Artificial Intelligence,* vol. 5, no. 4, pp. 221-232, 2016.

[32] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," in *The Twentieth International Conference on Machine Learning (ICML-2003),* Washington, D.C., 2003.

[33] H. Wang, T. M. Khoshgoftaar and K. Gao, "A Comparative Study of Filter-Based Feature Ranking Techniques," in *2010 IEEE International Conference on Information Reuse & Integration*, Las Vegas, Nevada, 2010.

[34] A. Agresti, Statistical Methods for the Social Sciences, 5th ed., Boston, Massachusetts: Pearson, 2018, p. 608.

[35] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning,* vol. 1, no. 1, pp. 81-106, 1986.

[36] H. Han, X. Guo and H. Yu, "Variable Selection Using Mean Decrease Accuracy and Mean Decrease Gini Based on Random Forest," in *The 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2016.

[37] I. T. Jolliffe and J. Cadima, "Principal Component Analysis: A Review and Recent Developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 374, no. 2065, p. 20150202, 2016.