Accident Analysis and Prevention 156 (2021) 106126

Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

Railroad accident analysis using extreme gradient boosting

Raj Bridgelall [a,*], Denver D. Tolliver [b]

[a] Department of Transportation, Logistics & Finance, College of Business, North Dakota State University, Fargo, ND, 58108, United States
[b] Upper Great Plains Transportation Institute, North Dakota State University, Fargo, ND, 58108, United States

ARTICLE INFO

Keywords:
Data cleaning
Feature engineering
Financial loss
Machine learning
Principle component analysis
Risk management

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.

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 (ASCE, 2017). 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 (Bridgelall and Tolliver, 2020). Hence, the federal government mandated that railroads deploy a positive train control (PTC) system by 2018 to help prevent accidents caused by human errors (Zhang et al., 2018). 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 (Fig. 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 (Ghofrani et al., 2018).

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 (Murphy, 2012). Therefore, another objective is to compare the classification performance of various types of ML models on the same dataset.

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 (Ilyas and Chu, 2019). A survey of data cleaning for ML found that the failure to discover and repair dirty data can weaken data analysis techniques (Jesmeen et al., 2018). Although a few approaches to data cleaning are common, every dataset poses unique challenges (Bridgelall et al., 2018). Hence, data scientists spend an average of 60 % of their time cleaning and organizing data (Ilyas and Chu, 2019).

* Corresponding author.
E-mail addresses: raj@bridgelall.com (R. Bridgelall), denver.tolliver@ndsu.edu (D.D. Tolliver).

https://doi.org/10.1016/j.aap.2021.106126
Received 23 December 2020; Received in revised form 19 March 2021; Accepted 3 April 2021
Available online 17 April 2021

Although the importance of using clean data is well-known, the research community has paid little attention to the advancement of data cleaning techniques (Rahm and Do, 2000). The most commonly used techniques are those that detect and remove outliers and duplicate records (Ilyas and Chu, 2019). 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).
- Interpreting the importance ranking of the feature relevance in predicting accident type (Sections 3.4 and 4.2).
- 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 (Sections 3.5 and 4.3).

The next section (Section 2) reviews related works and their findings in relation to the contributions of this research. Section 4 mirrors subsections of 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 (Iranitalab and Khattak, 2017). 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 (Ghofrani et al., 2018). This imbalance supports a claim in the introduction 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 (Dabbour et al., 2017). 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 (Liu and Khattak, 2017). 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 % (Keramati et al., 2020). Soleimani et al. (2019) used extreme gradient boosting to identify HRGCs that should be closed to prevent accidents (Soleimani et al., 2019). 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 (Wali et al., 2021).

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 (Liu et al., 2017). 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 (Wang et al., 2020). 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 % (Iranitalab and Khattak, 2020).

The survey of Ghofrani (2018) demonstrated that researchers have also used ML methods to analyze other aspects of railroad operations besides safety (Ghofrani et al., 2018). For example, Li et al. (2014) used ML to learn rules from historical and real-time data to predict railroad maintenance needs (Li et al., 2014). 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 (Lasisi and Attoh-Okine, 2019).

The benchmarking of ML performance is subjective because of its high relevance to the target problem of a particular study in a particular field (Olson et al., 2017). For example, a performance score considered to be “good” in the biotech industry when evaluating vaccine efficacy may be considered “poor” in the automotive industry when evaluating defective unit batches. Subjective performance assessments depend on the level of “acceptable” risk for a given application (Cook, 2007). Therefore, model evaluation often use the fuzzy academic grading system to assess and compare performance levels (Echauz and Vachtsevanos, 1995).

3. Methodology

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

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 processing layer led to the

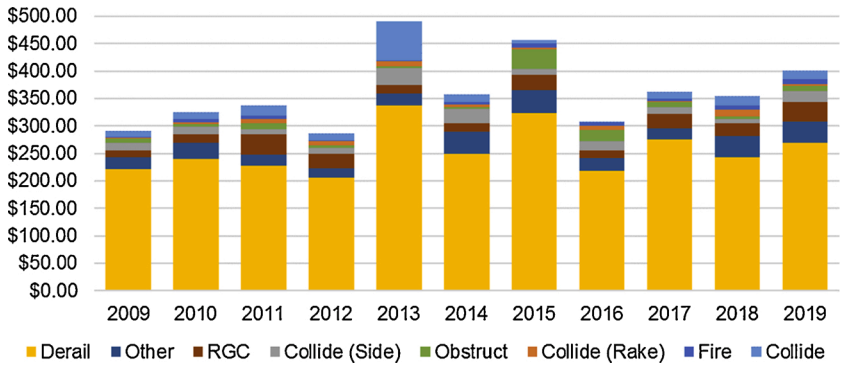

Fig. 1. Annual financial loss reported for different accident types.

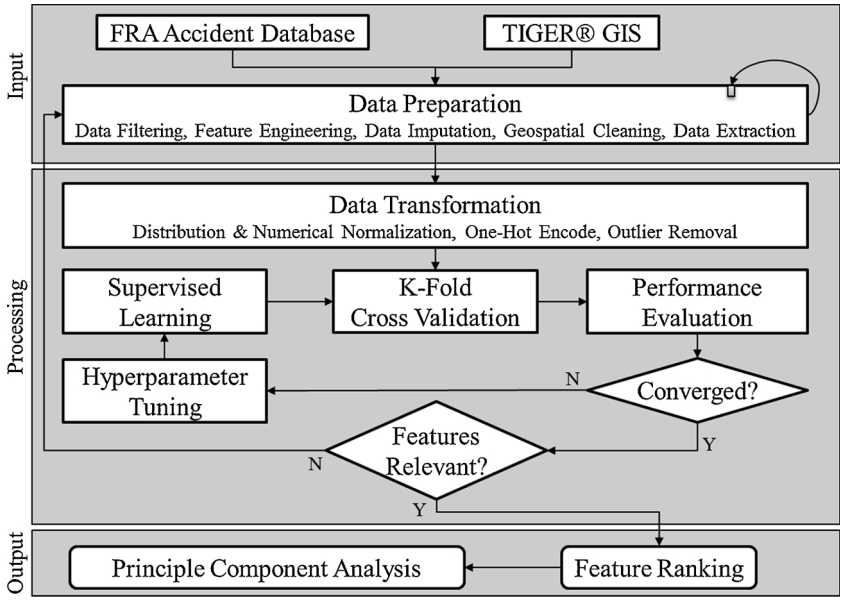

Fig. 2. The methodological framework.

discovery of additional errors that made some features irrelevant. In such cases, the framework logic looped back to the data preparation module to address the issue. The model building and validation processes also contained a loop that 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 source

The comprehensive Federal Railroad Administration (FRA) Rail Equipment Accident database provided the main dataset for the analysis (Bridgelall and Tolliver, 2020). Some of the data schema became inconsistent after the FRA changed reporting requirements for a few of the fields starting June 1, 2011 (FRA, 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.

Merging 8055 records from 2009 to 2011 with 21,242 records from 2012 to June 2020 produced a total of 29,297 records with 145 attributes. 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.

3.2. Data processing

3.2.1. Data cleaning

Table 1 describes criteria used to clean the data by eliminating irrelevant attributes or features. The classification of those criteria is a heuristic synthesis by the authors based on a broad understanding from the literature, which the table cites for further explanation.

Table 2 chronicles each criterion used to reduce the number of fields from 145 to 52.

3.2.2. Data extraction

Passenger trains accounted for a small portion 8.03 % (2354) of accidents. Removing those records enhanced the consistency of the dataset, which is known to improve the ML performance (Murphy,

Table 1
Criteria for Attribute or Feature Elimination.

Criteria	Description	Further Explanation
Sparsity	Attribute is missing more than 85 % of the values or contain zeros.	(Murphy, 2012)
Duplication	Attribute contains the same information as other attributes.	(Ilyas and Chu, 2019)
Correlated	Attribute is more than 90 % correlated with another.	(Géron, 2017)
Redundancy	Attribute contains information that is inherent in other attributes.	(Liu et al., 2012)
Noise	Attribute is not relevant to the target class.	(Rahm and Do, 2000)
Dispersion	Attribute has low variance or carries little or no information.	(Bridgelall et al., 2018)
Combinable	Attribute that can combine with others without losing information.	(Ilyas and Chu, 2019)

Table 2
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

2012). A side benefit was that the eliminated records also removed a few attributes that were associated with passenger trains only. Table 3 chronicles the net reduction of attributes from 52 (Table 2) to 50, and records from 29,297 to 15,087. The statistics shown in the table are the number of records (N), number of attributes or variables (V), and number of metadata fields (M). Adding the target attribute "DERAILED" indicated if the accident was a derailment type or not, and it became the label for supervised ML.

3.2.3. 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 (Manning and Mullahy, 2001). A shifted natural logarithm LN(1 + x) reduced the skew and prevented 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 proportion transformation retained information about the relative relationship among attributes while normalizing the values within the [0,1] range. Table 4 chronicles the transformation of attributes and their effect on reducing the number of attributes from 50 (Table 3) to 40.

3.2.4. Feature selection

Predictive modeling should not contain attributes where values are known only after the outcome. Therefore, the cleaning procedure eliminated *post-event* attributes such as the number of people injured, killed, or evacuated. Table 5 chronicles the feature reduction from 40 (Table 4) to 25.

3.2.5. Feature engineering

The procedures performed the following feature engineering:

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.

Table 6 summarizes the results of the feature engineering.

3.2.6. Data imputation

A few methods such as decision trees and Bayesian classifiers can work with missing data, but most cannot (Abidin et al., 2018). 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 in *feature space*. However, the existing methods did not enhance the contribution of the affected attributes towards predicting the target class. Therefore, this research developed a new method, dubbed *local association pivot* (LAP), to replace missing values based on spatial proximity rather than feature space proximity. The LAP method first creates a pivot table that aggregates non-missing values by a *spatial* 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.

Table 7 summarizes the results of the imputing missing values and the impact of each method. 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

Table 3
Chronicle of Data Reduction after Data Extraction.

Attribute	Statistic	Procedure
ACC_CAT	N: 29,297 V: 52 + 1 = 53 M: 5	Add accident category based on accident cause code: {Track, Equipment, Human, Signal, Miscellaneous}.
PASSTRN	N: 26,943 (92 %) V: 53−4 = 49 M: 5	Dropped accidents involving passenger type trains (PASSTRN = Y). Dropped associated attributes: LOADP1, LOADP2, EMPTYP1, EMPTYP2.
DERAILED	N: 26,943 (92 %) V: 49 + 1 = 50 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.

Table 4
Chronicle of the Transformed and Derived Attributes.

Attribute	Reduction	Procedure
HR24	50–3 + 1 = 48	Combined TIMEHR, TIMEMIN, AMPM to 24 -h 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.

Table 5
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 6
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}. {1} → "Freight", {2, 3, B, C} → "Passenger", {8, D, E} → "Locomotive", {5, 6} → "Cars", {4, 9, A} → "Work", {7} → "Yard"
ACC_TYPE	Renamed TYPE (accident type); repackaged as category labels: {1} → "Derail", {2, 3, 6} → "Collide", {4} → "Collide (Side)", {5} → "Collide (Rake)", {7, 8} → "RGC", {9} → "Obstruct", {10, 11} → "Fire", {12, 13} → "Other"
MOVEx	Renamed MOPERA; repackaged as labels {signal, control, restrict, blocks, not main} {1, D} → "Signal", {2, A, B, C, P} → "Control", {3, L, M, I} → "Restrict", {4, E, F, G, H, J, K} → "Blocks", {5, N, O} → "Not Main"

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.

3.2.7. Geospatial coordinate repair

Aside from missing geospatial coordinates, data entry errors resulted in erroneous or highly skewed geospatial locations. Fig. 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 suggested 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. The result was 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.

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 with the geospatial centroid of the FRA reported county.

Table 8 chronicles the progress of filling missing geospatial coordinates in each step of the procedure. The LAP method used all records prior to data reduction and 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.

Table 9 summarizes the final set of 25 attributes used to build the ML models. *One-hot-encoding* the categorical attributes increased the number of features from 25 to 51. Dispersion represents the relative amount of variability (information) that each attribute contributes to the overall variance in the data. The dispersion measure is the *entropy* and coefficient of variation (CV) for categorical and numerical attributes, respectively.

Table 7
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 % (6473)	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% (2605)	Layer 1: Fill missing CONSIST with: "Freight" if (LOADF1 + EMPTYF1) > 0 otherwise "Passenger" if (LOADP1 + EMPTYP1) > 0 or PASSTRN is "Y"
	8% (2605)	2% (844)	Layer 2: Fill missing CONSIST with: "Freight" if CLASS_RR is "1" (except "Amtrak") otherwise "Passenger" if RAILROAD (reporting railroad) is "Amtrak"
	2% (844)	1% (377)	Layer 3: Fill missing CONSIST with: "Work Train" if TRK_TYP is not "Main"
	1% (377)	0% (0)	Layer 4: Fill missing CONSIST with: "Work Train" if TONS (gross tons, excluding locomotives) is 0 otherwise fill missing CONSIST with "Freight" if TONS > 0
CWR	21 % (6378)	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%, (2049)	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: 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.

3.2.8. 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 (Liu et al., 2012). They could represent noisy data entries or rare events that can bias the training of an ML model, resulting in poor predictive performance. 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) (Rousseeuw and Driessen, 1999)
- Local outlier factor (LOF) (Breunig et al., 2000)
- Isolation forest (IF) (Liu et al., 2012)

Table 10 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.

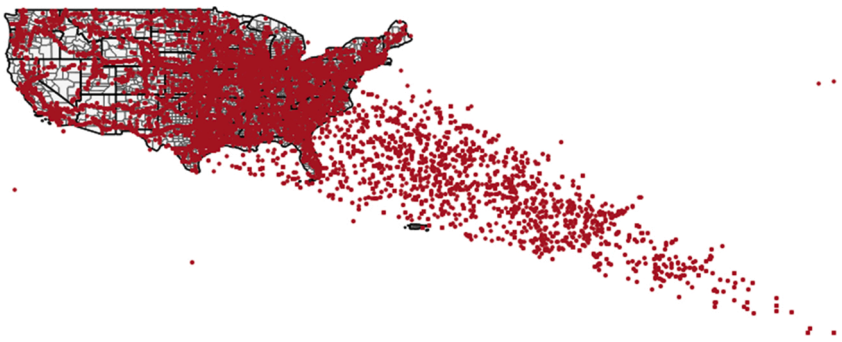

Fig. 3. Positions of the recorded geospatial coordinates in the FRA database.

Table 8
Chronicle of Geospatial Coordinate Cleaning.

Attribute	Missing Before	Missing After	Procedure (N = 29,297 records)
Latitude	21 % (6415)	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 % (6415)	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.
Latitude	2% (817)	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. Add the county name and state name strings to the metadata. Fill missing FRA geospatial coordinates with the county centroid coordinates.
Longitude	2% (820)	0% (6)	
Latitude	0% (6)	0% (0)	Manually fill missing geospatial coordinates for counties in Alaska and Florida.
Longitude	0% (6)	0% (0)	
FIPS5	0% (4)	0% (0)	Fill in missing FIPS5 codes for "Baltimore" and "Skagway" stations.
LAT	0% (0)	0% (0)	Rename Latitude to LAT and Longitude to LON after the geospatial cleaning procedure.
LON	0% (0)	0% (0)	

Table 9
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-h.
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 (0–1)
CARS_HZMT	2.800	Continuous	Proportion of loaded cars carrying hazardous materials (0–1)
HUMANS	0.562	Continuous	Number of humans present on the train.

3.3. Machine learning

Many different types of ML models emerged over the years, and each tend to behave differently on different types of datasets (James et al., 2013). The next subsections describe the different types of models and their hyperparameter tuning to optimize performance on the FRA dataset.

3.3.1. Supervised classification models

Table 11 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 (Géron, 2017). Jame et al. (2013) discusses both the theory and practical implementation of Naïve Bayes (NB), knearest-neighbors (kNN), and tree-based boosting methods (James et al., 2013). 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 (Hastie et al., 2016). Murphy (2012) covers the various methods from a more theoretical and probabilistic perspective (Murphy, 2012).

Table 10
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

3.3.2. 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 James et al. (2013). 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 (Fawcett, 2006). 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 12 describes each metric and summarizes their advantages and disadvantages. All performance metric except the AUC was sensitive to class imbalance in the dataset.

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 (Krawczyk, 2016).

3.4. Feature ranking

Attributes that contain noisy, irrelevant, or redundant information can diminish the performance of ML methods (Yu and Liu, 2003). 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 13 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

Table 11
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.
	Ada Boost (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. 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.

Table 12
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.

Table 13
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)
Chi-Squared	Measures a dependency or association between the feature and the target class by using a chi-square statistic.	(Wang et al., 2010)
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)
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)
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)

some compensating for the weaknesses of the other; therefore, they do not provide identical rankings (Wang et al., 2010). However, a strong correlation among rankings indicates that the top-ranking attributes do contribute most towards ML classification performance.

3.5. 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 (Jolliffe and Cadima, 2016). 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. Therefore, features that are weak components of most PCs tend to be associated with noise in the data. Analyst also use PCA to transform high dimensional data into a lower dimension feature space to enable the visualization of both structure and noise in the dataset (Anowar et al., 2021).

4. Results

The subsections of this section present the results of applying machine learning, attribute ranking, and PCA to the cleaned and transformed dataset presented in the previous section.

4.1. Machine learning

Table 14 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,

Table 14
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	γ: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 (α: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, ε:1), R: E.Net (α:10^{-5}, 0.15), L: IVS (η_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, ε:1.0)
Null	0.500	0.674	0.543	0.455	0.674	No parameters to tune

and optimizer algorithm (OA).

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

As noted in Table 14, 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.

4.2. Feature ranking

Table 15 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 16. The correlation ranges from 84.2 % for the gini and chi-squared methods to 94.5 % for the ANOVA and chi-squared methods.

Fig. 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).

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 (Fig. 5a). The distinction is significant for class 1 tracks because it has the highest frequency of occurrence (Fig. 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 (Fig. 5c). Similarly, the class probability was higher for derailment type accidents in non-signalized territories (Fig. 5d). The probability difference was much lower for the lower ranking attributes, but taken together, they improve the ML classification performance.

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

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.

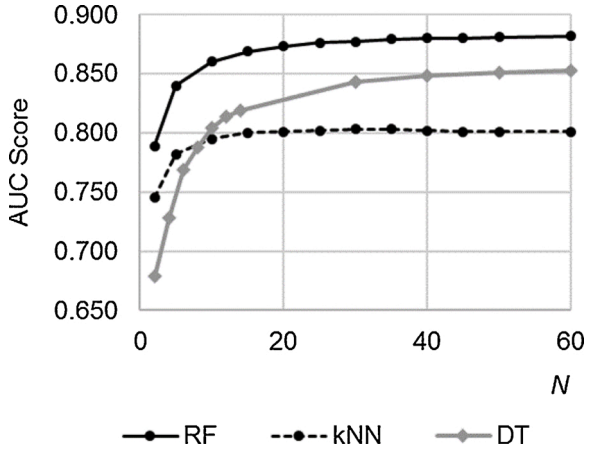

Fig. 4. AUC score as a function of hyperparameter *N*.

Table 15
Feature Importance Ranking.

Feature	ANOVA	χ^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

Table 16
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

4.3. Principle component analysis

Fig. 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.

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

Fig. 8 shows that PC1 and PC4 form elongated elliptical clusters for the top ranking attributes of track class (Fig. 8a), movement authority (Fig. 8b), and track type (Fig. 8c). Fig. 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.

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

5. Discussion

The performance of the top four ML methods reflected the

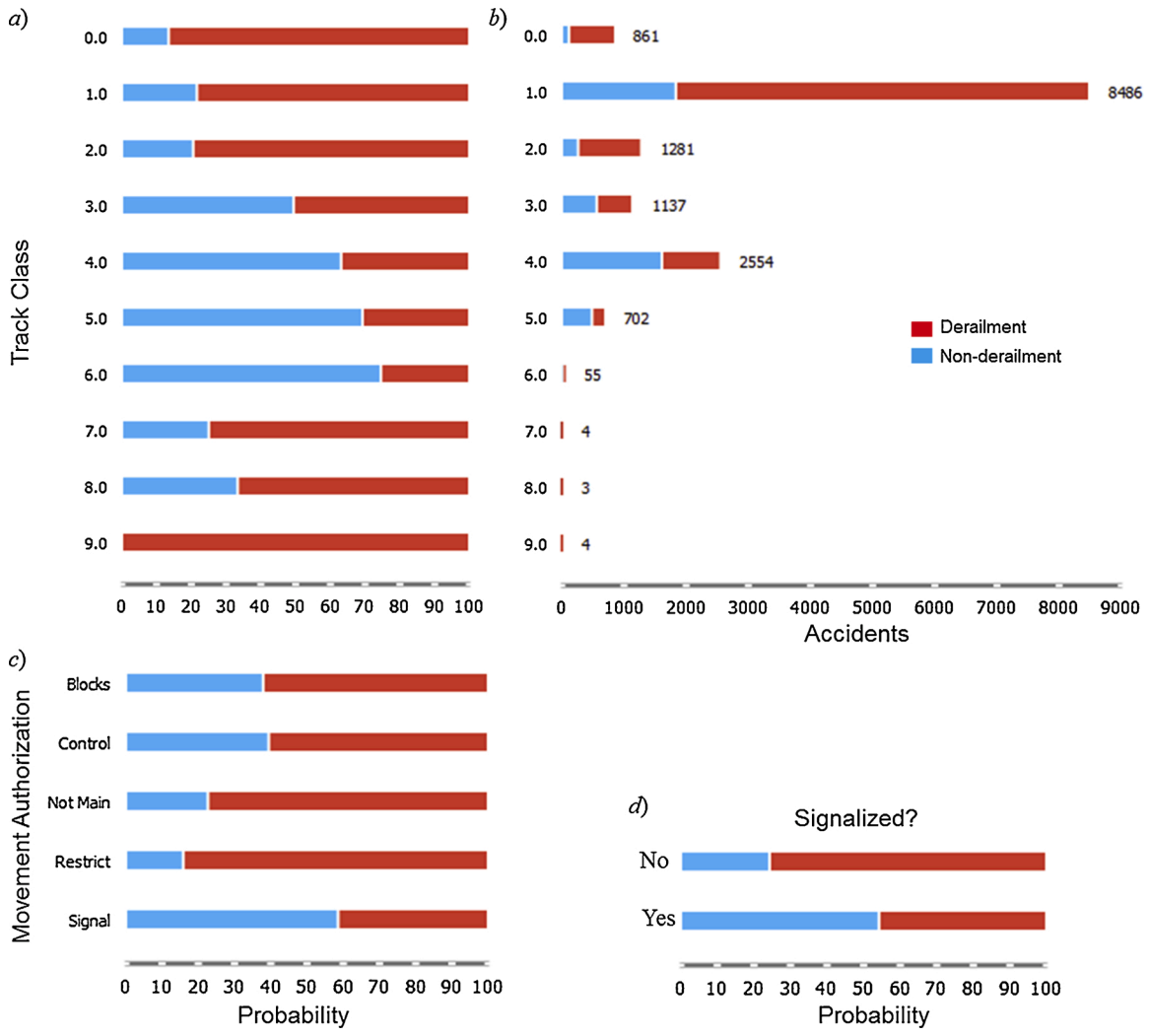

Fig. 5. Class probability for the top two and fourth ranking attributes.

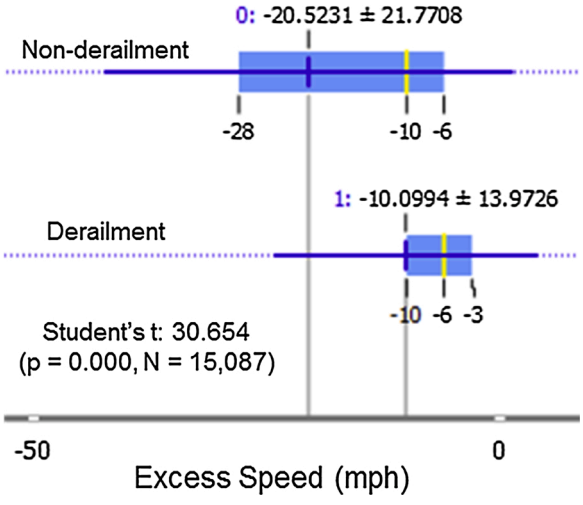

Fig. 6. Distribution and statistics for excess speed.

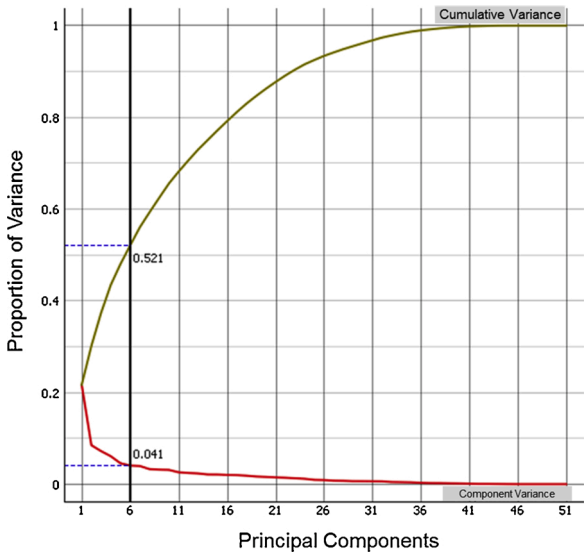

Fig. 7. The proportion of variance in the data that each PC explains.

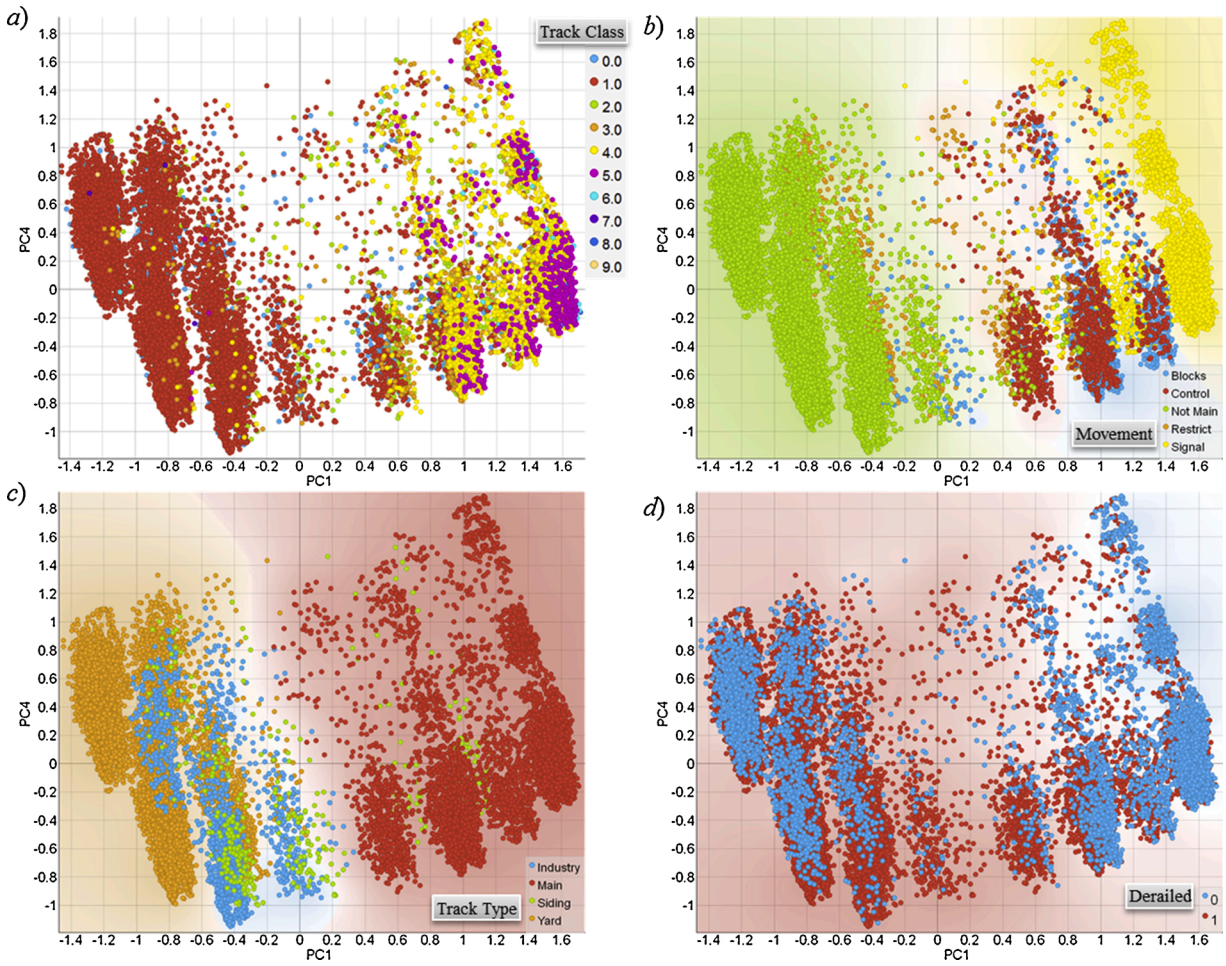

Fig. 8. Data clusters for attributes with high power to distinguish among the target classes.

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, the LAP method provided a more intelligent and effective scheme to impute missing values such as track density based on those of *spatial* neighbors rather than guessing values based on the mean, most frequent, or nearest neighbors in feature space. The geospatial join method provided the next best alternative to repair 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. Using the traditional academic grading system for performance, the top four models provided "good" overall performance based on 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 Fig. 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 (Fig. 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.

Fig. 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 (Fig. 8d) there was an observable bias of derailment type accidents towards clusters on the left. The bias corresponds to clusters of class 1 tracks (Fig. 8a), movement authorities not on the mainline (Fig. 8b), and non-

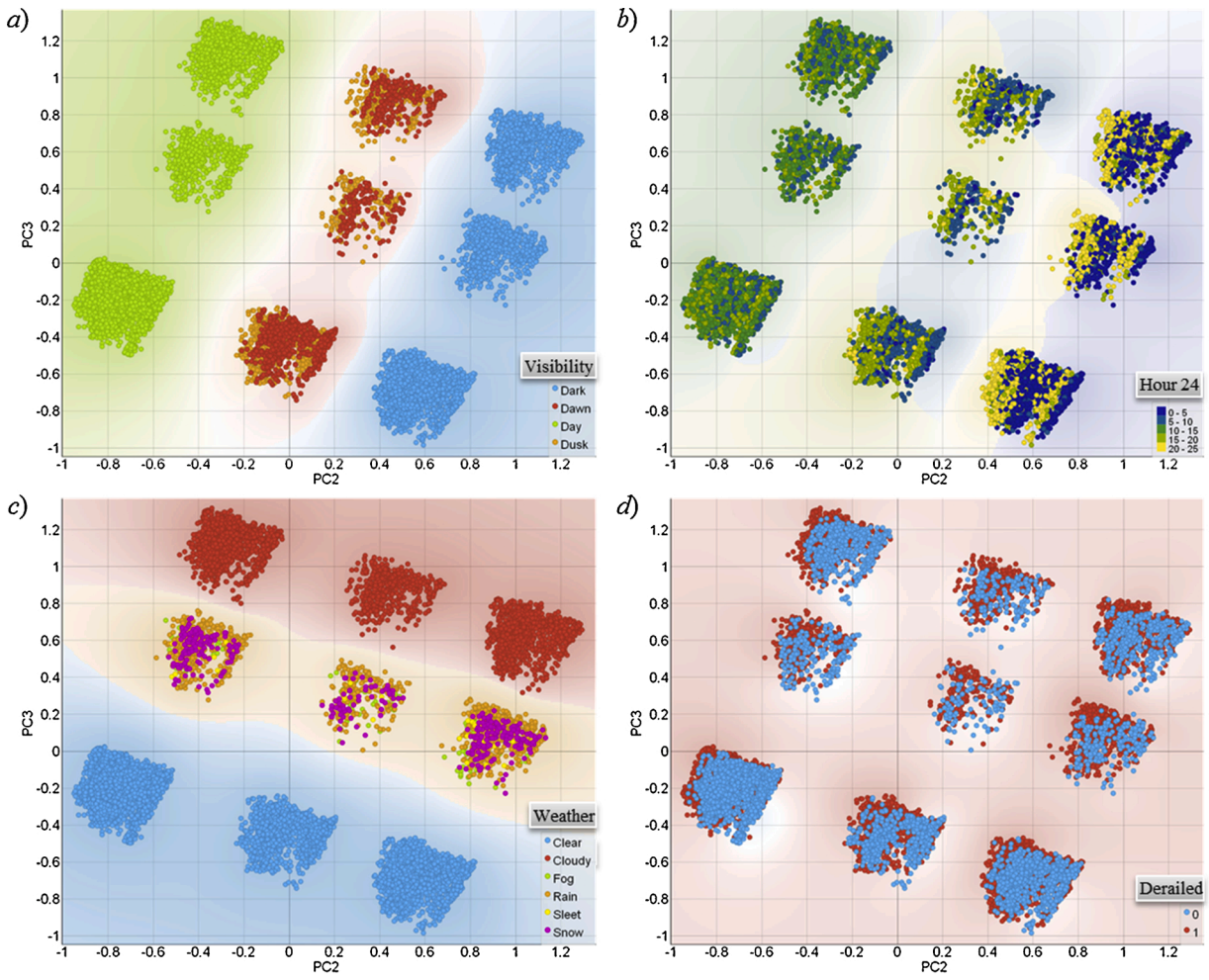

Fig. 9. Data clusters for attributes with low power to distinguish among the target classes.

main track types (Fig. 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.

Fig. 9 shows that PC2 and PC3 form clusters for the attributes of visibility (Fig. 9a), hour (Fig. 9b), and weather (Fig. 9c), which are low-ranking. The even distribution of each target class across each cluster (Fig. 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 (Fig. 9a) suggest similarities in the visibility at dawn and dusk, as expected. Those similarities also corresponded to the separation of "Hour 24" (Fig. 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 (Fig. 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 the dataset. On the other hand, kNN seeks local neighborhoods in feature space 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.

Overall, the analysis suggested that 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. Those findings correspond with the intuition that lower-class tracks, which has lower speed limits, and movements with restricted limits are so designated because those operations are associated with higher safety risks, which the ML confirmed. Similarly, there is less guidance for movements in territories without signalization, so the risk of derailments due to track irregularities or switch problems is higher. However, it may not be wise to go beyond probabilities and statistical associations by assuming general latent reasons for the ML outcome because there are no exclusive distinctions between accident causes for each accident type.

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%–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.

Data availability

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

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.

CRediT authorship contribution statement

Raj Bridgelall: Conceptualization, Methodology, Software, Data curation, Formal analysis, Writing - original draft. **Denver D. Tolliver:** Supervision, Resources, Funding acquisition, Project administration, Validation, Writing - review & editing.

Declaration of Competing Interest

The authors report no declarations of interest.

References

Abidin, N.Z., Ismail, A.R., Emran, N.A., 2018. Performance analysis of machine learning algorithms for missing value imputation. Int. J. Adv. Comput. Sci. Appl. 9 (6).
Agresti, A., 2018. Statistical Methods for the Social Sciences, 5th ed. Pearson, Boston, Massachusetts. p. 608.
Anowar, F., Sadaoui, S., Selim, B., 2021. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). Comput. Sci. Rev. 40, 100378.
ASCE, 2017. Infrastructure Report Card. American Society of Civil Engineers, Reston, VA.
Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J., 2000. LOF: identifying density-based local outliers. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data.
Bridgelall, R., Tolliver, D.D., 2020. Closed form models to assess railroad technology investments. Transp. Plan. Technol. 43 (7), 639–650.
Bridgelall, R., Lu, P., Tolliver, D.D., Xu, T., 2018. Mining connected vehicle data for beneficial patterns in Dubai taxi operations. J. Adv. Transp. 2018, 1–8.
Cook, N.R., 2007. Use and misuse of the receiver operating characteristic curve in risk prediction. Circulation 115 (7), 928–935.
Dabbour, E., Easa, S., Haider, M., 2017. Using fixed-parameter and random-parameter ordered regression models to identify significant factors that affect the severity of drivers' injuries in vehicle-train collisions. Accid. Anal. Prev. 107, 20–30.
Echauz, J., Vachtsevanos, G., 1995. Fuzzy grading system. IEEE Trans. Educ. 38 (2), 158–165.
Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognit. Lett. 27 (8), 861–874.
FRA, 2011. Rail Equipment Accident/Incident Data File Structure and Field Input Specifications. Federal Railroad Administration (FRA), Washington, D.C.
Géron, A., 2017. Hands-On Machine Learning With Scikit-learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed. O'Reilly Media, Sebastopol, California. p. 856.
Ghofrani, F., He, Q., Goverde, R.M., Liu, X., 2018. Recent applications of big data analytics in railway transportation systems: a survey. Transp. Res. Part C-Emerg. Technol. 90, 226–246.
Han, H., Guo, X., Yu, H., 2016. Variable selection using mean decrease accuracy and mean decrease gini based on random Forest. The 7th IEEE International Conference on Software Engineering and Service Science (ICSESS).
Hastie, T., Tibshirani, R., Friedman, J., 2016. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, New York, New York, p. 767.
Ilyas, I.F., Chu, X., 2019. Data Cleaning. Association for Computing Machinery and Morgan & Claypool Publishers, New York, NY, p. 282.
Iranitalab, A., Khattak, A.J., 2017. Comparison of four statistical and machine learning methods for crash severity prediction. Accid. Anal. Prev. 108, 27–36.
Iranitalab, A., Khattak, A.J., 2020. Probabilistic classification of hazardous materials release events in train incidents and cargo tank truck crashes. Reliab. Eng. Syst. Saf. 199, 106914.
James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An Introduction to Statistical Learning With Applications in R, 112. Springer, New York.
Jesmeen, M.Z.H., Hossen, J., Sayeed, S., Ho, C., Tawsif, K., Rahman, A., Arif, E., 2018. A survey on cleaning dirty data using machine learning paradigm for big data analytics. Indones. J. Electr. Eng. Comput. Sci. 10 (3), 1234–1243.
Jolliffe, I.T., Cadima, J., 2016. Principal component analysis: a review and recent developments. Philos. Trans. Math. Phys. Eng. Sci. 374 (2065), 20150202.
Keramati, A., Lu, P., Iranitalab, A., Pan, D., Huang, Y., 2020. A crash severity analysis at highway-rail grade crossings: the random survival forest method. Accid. Anal. Prev. 144, 105683.
Krawczyk, B., 2016. Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell. 5 (4), 221–232.
Lasisi, A., Attoh-Okine, N., 2019. Machine learning ensembles and rail defects prediction: multilayer stacking methodology. ASCE. J. Risk Uncertain. Eng. Syst. A. Civ. Eng. 5 (4), 4019016.
Li, H., Parikh, D., He, Q., Qian, B., Li, Z., Fang, D., Hampapur, A., 2014. Improving rail network velocity: a machine learning approach to predictive maintenance. Transp. Res. Part C-Emerg. Technol. 45, 17–26.
Liu, J., Khattak, A.J., 2017. Gate-violation behavior at highway-rail grade crossings and the consequences: using geo-spatial modeling integrated with path analysis. Accid. Anal. Prev. 109, 99–112.
Liu, F.T., Ting, K.M., Zhou, Z.-H., 2012. Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data 6 (1), 3.
Liu, X., Saat, M.R., Barkan, C.P., 2017. Freight-train derailment rates for railroad safety and risk analysis. Accid. Anal. Prev. 98, 1–9.
Manning, W.G., Mullahy, J., 2001. Estimating log models: to transform or not to transform? J. Health Econ. 20 (4), 461–494.
Murphy, K.P., 2012. Machine Learning : A Probabilistic Perspective. The MIT Press, Cambridge, Massachusetts.

Olson, R.S., Cava, W.G.L., Orzechowski, P., Urbanowicz, R.J., Moore, J.H., 2017. PMLB: a large benchmark suite for machine learning evaluation and comparison. BioData Min. 10 (1), 1–13.
Quinlan, J.R., 1986. Induction of decision trees. Mach. Learn. 1 (1), 81–106.
Rahm, E., Do, H.H., 2000. Data cleaning: problems and current approaches. IEEE Data (base) Eng. Bull. 23, 3–13.
Rousseeuw, P.J., Driessen, K.V., 1999. A fast algorithm for the minimum covariance determinant estimator. Technometrics 41 (3), 212–223.
Soleimani, S., Mousa, S.R., Codjoe, J., Leitner, M., 2019. A comprehensive railroad-highway grade crossing consolidation model: a machine learning approach. Accid. Anal. Prev. 128, 65–77.
Wali, B., Khattak, A.J., Ahmad, N., 2021. Injury severity analysis of pedestrian and bicyclist trespassing crashes at non-crossings: a hybrid predictive text analytics and heterogeneity-based statistical modeling approach. Accid. Anal. Prev. 150, p. 105835.
Wang, H., Khoshgoftaar, T.M., Gao, K., 2010. A comparative study of filter-based feature ranking techniques. In: 2010 IEEE International Conference on Information Reuse & Integration. Las Vegas, Nevada.
Wang, B.Z., Barkan, C.P.L., Saat, M.R., 2020. Quantitative analysis of changes in freight train derailment causes and rates. J. Transp. Eng. Part A Syst. 146 (11), p. 4020127.
Yu, L., Liu, H., 2003. 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..
Zhang, Z., Liu, X., Holt, K., 2018. Positive Train Control (PTC) for railway safety in the United States: policy developments and critical issues. Util. Policy 51 (2018), 33–40.