Preprint Manuscript: Bridgelall, R., and Tolliver, D. (2024). Railroad Accident Analysis by Machine Learning and Natural Language Processing. *Journal of Rail Transport Planning & Management*, 29(March), 100429. DOI: 10.1016/j.jrtpm.2023.100429.

Railroad Accident Analysis by Machine Learning and Natural Language Processing

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

Funding: 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.

Data Availability: The data used in this study is publicly available from the United States Federal Railroad Administration as cited within the manuscript where described.

Railroad Accident Analysis by Machine Learning and Natural Language Processing

Abstract

The evolving complexities of railroad systems also increase their vulnerability to failure from human error. This study compared the outcomes of two workflows that incorporated 11 different machine learning techniques to identify characteristics of railroad operations that are generally associated with human-caused accidents. The first workflow engineered features from the fixed attribute fields of a large railroad accident database and the second applied natural language processing to extract features from the unstructured accident narratives. Both workflows applied a Shapely game-theoretic model to rank the importance of features based on their marginal contribution towards predicting accident cause. Among several interesting findings, some of the most unexpected were that human-caused accidents are generally not associated with high train speeds nor derailment type accidents, and that shoving cars is riskier than pulling cars. Those, and other findings, from this study can inform management decisions, planning, and policies to minimize the risk of human-caused accidents.

Keywords: data cleaning; feature engineering; game theory; machine learning; risk management; text mining

1 Introduction

Human factors have consistently been the dominant cause of railroad accidents. Analysis of the U.S. Federal Railroad Administration (FRA) equipment accident database revealed that humans caused more than 35% of all the reported accidents from 2009 to 2019. The next dominant factor was track- and roadbed-related problems, which caused 23% of the accidents during the same period. A few high-profile accidents that resulted in disastrous derailments and fatalities were due to excessive train speeds or operators leaving track switches in the wrong position. Those accidents dominated media attention and inspired a federal rule mandating railroads to deploy a positive train control (PTC) system to help prevent these types of accidents (Zhang, Liu, & Holt, 2018). Hence, it is tempting to expect that PTC will eliminate all human-caused accidents. However, exploratory data analysis of the FRA accident database revealed that there are many human-caused accidents that result from factors that PTC does not address (Figure 1). Therefore, the **research question** is: What aspects of railroad operations are generally associated with human-caused accidents? An **objective** of this research is to determine how machine learning (ML) and natural language processing (NLP) techniques compare in answering the research question.

The **contributions** of this research are:

- Two data mining workflows to compare the performance of ML and NLP techniques in predicting human-caused accidents (Section 3).
- A ranking of factors associated with human-caused accidents by applying a game theoretic model (Section 4).

Other than Section 3 and Section 4 mentioned above, the next section (Section 2) reviews related works in railroad accident analysis that applied NLP or ML techniques. Section 5

discusses the analysis outcome and provides an interpretation of the findings. Section 6 recaps the methodological approach and findings to conclude the work and point to future research.

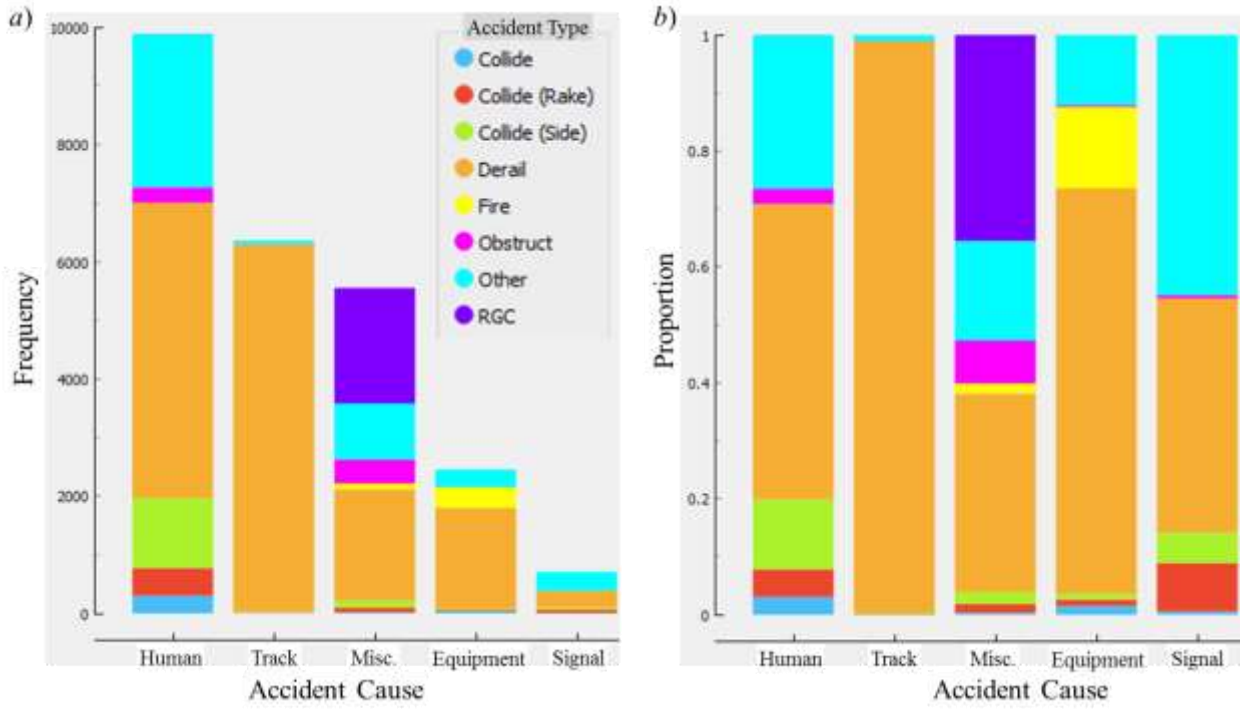

Figure 1: Causes and types of railroad accidents by a) frequency and b) proportion.

2 Literature Review

Analysts found that degraded human performance remains a major contributing factor in railway incidents (Kyriakidis, Simanjuntak, Singh, & Majumdar, 2019). There were a handful of studies to characterize factors in human error that caused railroad accidents. Catelani et al. (2021) incorporated the concept of Eustress in simulations to estimate the impact of stress on railroad operator performances based on the Yerkes-Dodson empirical relationship between pressure and performance (Catelani, Ciani, Guidi, & Patrizi, 2021). In a more general study, Shin et al. (2021) assessed the relative importance of components in human–system interactions that can lead to control system hazards (Shin, Lee, Shin, Jang, & Park, 2021).

Some accident records contain both structured and unstructured data. The former are fixed field attributes that record information such as dates, types of equipment involved, weather conditions, infrastructure characteristics, and location features. Unstructured data is mainly in the form of short narratives that attempt to describe the accident (Suh, 2021). A 2018 review paper found that only one study on railroad accident analysis applied text mining to the unstructured data (Bala & Bhasin, 2018). Brown (2016) conducted that study by applying text mining to narratives in the FRA accident database to predict the cost of extreme railroad accidents (Brown, 2016). The main finding was that keywords in the accident narratives provided insights that the structured attributes could not. Later, Williams and Betak (2019) compared the performance of latent semantic analysis (LSA) and latent dirichlet allocation (LDA) in railroad accident topic classification (Williams & Betak, 2019). They found that the two topic modeling techniques were complementary and that both identified tractor-trailer trucks as a problem at rail grade crossings (RGCs). More recently, Soleimani et al. (2021) added text mining and spatial analysis to identify RGCs that should be closed to prevent accidents (Soleimani, Leitner, & Codjoe, 2021). In a related study, Wali et al. (2021) applied text mining to narratives of railroad trespassing accidents and confirmed that fatal injuries tend to occur after suicide attempts and during the use of headphones or cellphones (Wali, Khattak, & Ahmad, 2021). Most recently, Song et al. (2022) estimated freight train derailment severity with both structured and unstructured data by applying LDA to extract critical topics from the accident narratives (Song, Zhang, Qin, Liu, & Hu, 2022).

Studies that applied ML to the structured attributes of the FRA database were more common than those that applied NLP to the unstructured attributes. Liu and Khattak (2017) used geospatial modeling to determine that RGC gate violations were more highly associated with

two-quadrant than four-quadrant gates (Liu & Khattak, 2017). Zhou et al. (2020) found that random forest was better than decision trees at predicting accidents at highway rail grade crossings because it could better accommodate unbalanced data (Zhou, Lu, Zheng, Tolliver, & Keramati, 2020). Gao et al. (2021) found that combining convolution neural network (CNN) and resampling to reduce data imbalance provided the best accident prediction accuracy among deep learning, random forest, and decision tree methods (Gao, Lu, & Ren, 2021). Panda et al. (2022) found that the gradient boosting ML method provided the best accuracy in predicting railroad accident severity (Panda, Mishra, Dash, & Nawab, 2022).

Haleem and Gan (2015) applied a mixed logit model to RGC crash data and found that the likelihood of injury increased with higher train speeds and older drivers (Haleem & Gan, 2015). Saunders et al. (2019) used associative data mining to discover that the addition of passive signage yielded only a slight improvement in compliance with rules designed to vacate the dynamic envelope zone when stopping at a RGC (Saunders, Mousa, & Codjoe, 2019).

Iranitalab and Khattak (2020) found that the random forest model outperformed logistic regression, naïve Bayes, and support vector machine ML methods in predicting hazardous material releases (Iranitalab & Khattak, 2020). Noguchi et al. (2020) applied network theory to the fusion of transport accident network and transport environmental factors to illustrate the complex process of hazardous material releases from railroad accidents (Noguchi, Hienuki, & Fuse, 2020). Liu et al. (2017) found that signalized territories and tracks rated at higher classes were associated with fewer train derailments (Liu, Saat, & Barkan, 2017). Data mining by Wang et al. (2020) confirmed that reductions in broken rails, irregular track geometry, and wheel-related equipment defects can reduce derailment-type accidents (Wang, Barkan, & Saat, 2020).

Overall, there is a gap in the literature about studies applying the same ML techniques to both structured and unstructured data to gain insights from railroad accidents. Our work compared the efficacy of two distinct workflows employing the same 11 machine learning techniques to identify critical attributes linked to human-caused accidents in railroad operations. We could not find any other work that applied a Shapley value-based game-theoretic model to rank the significance of features according to their marginal contribution in predicting railroad accident causality, using features extracted from both structured and unstructured data.

3 Methodology

Figure 2 and Figure 3 show the four-layer workflow developed to process the structured and unstructured data, respectively. The first two layers of each workflow are unique, whereas the last two are similar. An important characteristic of these workflows is the logic test that enables looping back to previous layers. In so doing, the processes converge to maximize the predictive performance of each model type.

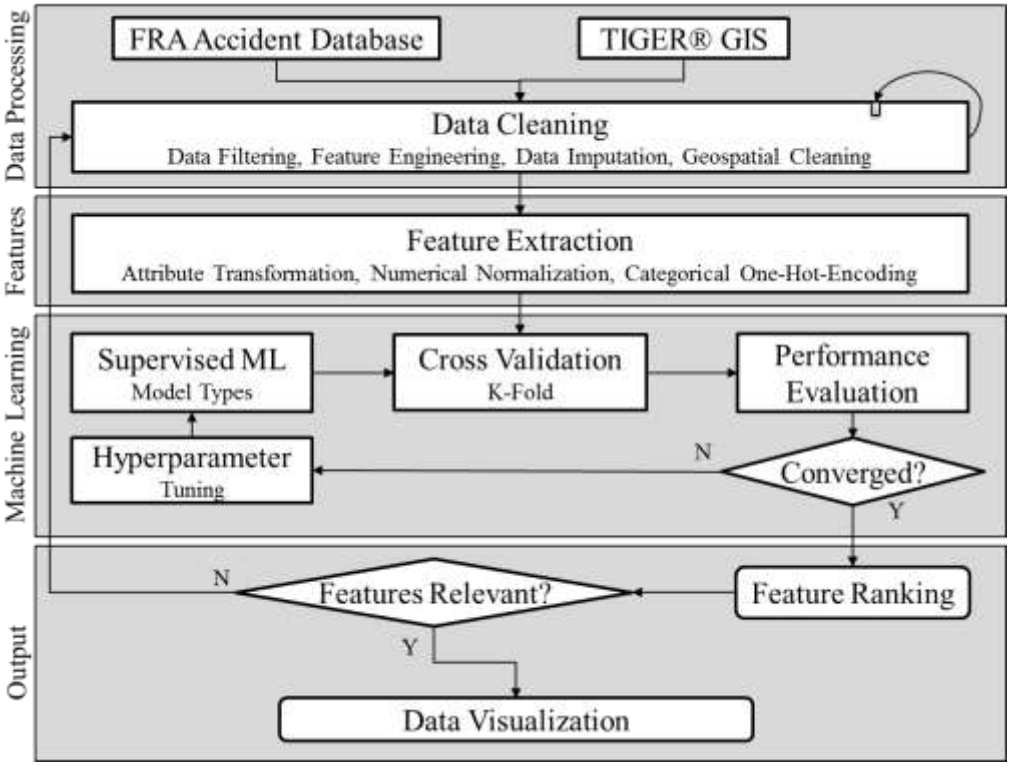

Figure 2: The workflow for fixed field machine learning.

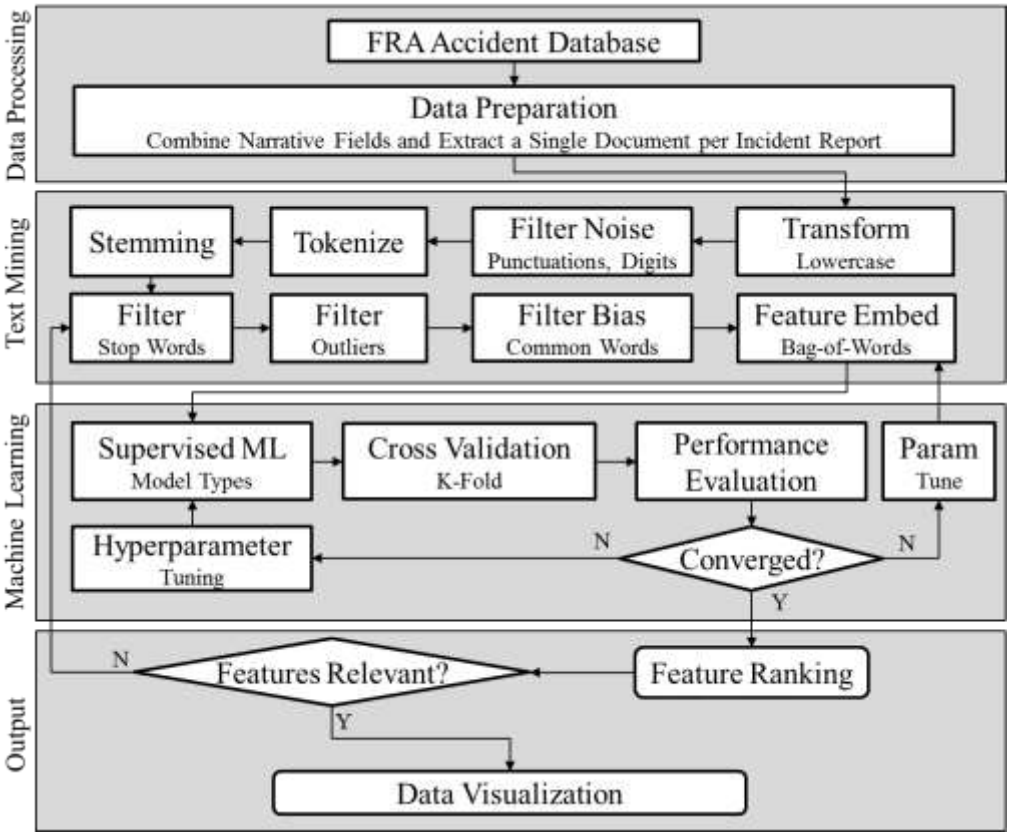

Figure 3: The workflow for text mining and machine learning.

3.1 Data

The FRA dataset contained more than 26,000 accident records from January 1, 2009, to June 30, 2020, each containing 145 fields (FRA, 2011). The Topologically Integrated Geographic Encoding/Line (TIGER/Line™) shapefiles from the U.S. Census Bureau encoded polygon boundary representations of all U.S. counties (USCB, 2019). The associated data tables of the TIGER database contained 11 fields that listed information about each of the 3,108 counties on the continental United States, such as their names, state, geospatial coordinates of their centroid, land area, and water area. The procedures of Figure 2 extracted structured features after combining fixed fields from both datasets. The procedures of Figure 3 aggregated accident narratives from the FRA database and applied natural language processing (NLP) to extract features. The extracted features from both workflows fed the ML procedures.

3.2 Models from Structured Data

The following subsections describe the data processing and feature extraction procedures applied to the structured portion of the data.

3.2.1 Data Processing

The data processing layer of the workflow consisted of a set of procedures to clean the structured (fixed attribute) data and extract relevant features. The *data filtering* procedure removed attributes that were too sparse, redundant, or irrelevant. Sparse attributes had a considerable proportion of missing or zero values, which means that they would contribute little or no information toward predicting the target class. Redundant attributes are highly correlated with other attributes, so they also provide no new information to improve predictive performance. Irrelevant attributes such as railcar, track, and county identifiers add complexity or noise to the ML models.

The *feature engineering* procedure reduced features and categories by fusion and transformation to increase information and reduce noise. An example of feature fusion was the combining of fields containing the hour, minute, and AM/PM flag into a 24-hour continuous variable. An example of category fusion was the reduction of 14 types of equipment stored in the "CONSIST" field to just 6 categories, based on their functional similarities. The resulting categories were "freight," "passenger," "work train," "yard equipment," "cars" and "locomotives."

The *data imputation* procedure filled missing values of a feature to enable the operation of ML models that are intolerant of missing data. For example, ML models such as decision trees and naïve Bayes can work with attributes that have missing values, but others, such as neural networks and logistic regression, cannot. The most popular and effective imputation methods fill missing values with the mean, most frequent value, or the value of their nearest neighbor in feature space (Abidin, Ismail, & Emran, 2018). The rationale for using such methods is that common values will prevent the affected data instances from biasing the predictions toward any one class without affecting the ability of non-missing values of its other features to contribute meaningfully towards predictions.

The *geospatial data cleaning* procedure used a geographic information system (GIS) to compare the county associated with the reported geospatial coordinates (latitude and longitude) to the county reported for the accident. The procedure then replaced the geospatial coordinates of any mismatched instances with the centroid coordinates of the reported county from the TIGER shapefile. The spatial join procedure of the GIS assured consistency between the reported county name, the reported FIPS code, and those of the joined geospatial location.

3.2.2 Feature Extraction

The *attribute transformation* procedure converted the attribute values of highly skewed distributions to improve their symmetry. The shifted natural logarithm, LN(1 + x), transform and the squared, x^2, transform reduced the skew of right and left skewed distributions, respectively. The rationale for skew reduction is that some ML models assume normally distributed features or operate better with less skewed distributions (Manning & Mullahy, 2001). Another transformation was to replace absolute values with proportional values to improve comparability among ML models. For example, the procedure converted the position of cars on trains to a relative position with respect to the train length. That is, the feature encodes whether the involved cars were toward the front, center, or rear of a train of any length.

The *numerical normalization* procedure converted all values to the [0, 1] range. The rationale for normalization was that ML models work with comparable values when applying weights to features. Also, some algorithms that compute a gradient in multiple dimensions work best when all attributes have the same range (Géron, 2017).

The *one-hot-encoding* procedure created a new binary feature for each value of a categorical attribute. This transformation assures the operation of ML algorithms that use numerical data only (Géron, 2017). Treating binary values as numerical values also makes them comparable to the normalized numeric features. Table 1 summarizes the data processing procedures and their actions.

Table 1: Data Processing Procedures

Procedure	Actions
Filter Missing	Remove attributes with a large proportion of missing values or constants.
Filter Irrelevant	Remove irrelevant or highly correlated attributes.
Feature Engineering	Combine attributes and categories to increase information or reduce noise.
Data Imputation	Fill missing values using the mean, most frequent, or the mean value of a nearby location.
Geospatial Cleaning	Repair low-resolution or erroneous geospatial coordinate entries.
Transformation	Reduce the skew of attribute distributions.
Normalization	Convert the values of all attributes to the [0, 1] range.
One-Hot-Encoding	Convert categorical variables to numerical features.

3.3 Models from Unstructured Data

The following subsections describe the data processing and feature extraction for the unstructured portion of the data.

3.3.1 Data Processing

The first workflow that operated on the structured portion of the data reduced the dataset by eliminating some features and instances. The second workflow then operated on the remaining data by extracting the accident narratives. The resulting text documents, without the fixed field attributes, became the corpus on which to apply the text mining procedures.

3.3.2 Text Mining

NLP is a branch of machine learning that focuses on the communication interface between computers and the natural human language forms that include text, speech, vocalizations, signing, images, and body language (Aggarwal, 2015). Text mining is a subset of NLP that seeks to extract meaning from text documents only. The workflow of Figure 3 used a collection of procedures to extract features from the corpus of accident narratives. The first step was to reduce word redundancy by *transforming* all characters to lowercase. The next step was to *filter* out punctuations and digits that conveyed no linguistic meaning. The third step was to *tokenize* by extracting words into an array.

Stemming is a normalization procedure that reduces feature redundancy by transforming all inflected forms of a word to the lexical stem of its root word. For example, a stemming algorithm would map the words “damaged”, “damaging”, “damages”, and “damage” to the word stem “damag”. The stemming procedure of this workflow used the Porter stemmer for the English language, based on its popularity in NLP and search engine design (Jones & Willett, 1997).

Stop words facilitate correct usage of grammar but they do not convey information that distinguishes one document from another. Examples of stop words include “the,” “on,”, “at,” “which,” “and,” “but,” and any other word that a data scientist wishes to define as such to improve the performance objective. The feedback loop from the ML layer provides a means to modify the stop words list while evaluating the predictive performance of the ML models. The feedback loop also provided a means for the analyst to validate the meaning of a stop word based on the context in the text narrative.

Outlier words can create noise in the data because of their sparsity across documents. Similarly, words that appear in most of the documents, for example more than 90%, can create a bias on the corpus that adds no information to individual instances. Hence, the outlier filter removed seldom used and frequently used words with threshold adjustments accommodated through the feedback loop to improve predictive performance.

Text embedding is a procedure that converts word arrays to numbers that become features of the ML models. Among text embedding methods, the bag-of-words (BOW) model is one of the most effective and easiest to implement. The BOW model is a simplified representation of a narrative as an array of unordered words from the union of documents analyzed (Aggarwal, 2015). The BOW hyperparameters allow for various numeric representations of a document such as the word count or a binary flag indicating the word presence. Another hyperparameter is

feature normalization by a weight equal to the inverse document frequency (IDF). The IDF of a term is the logarithm of the inverse proportion of documents containing the term. Hence, the product of the term frequency (TF) and the IDF (TF-IDF) reduces the importance of common words in the corpus to minimize bias. A regularization hyperparameter allows for normalizing the length of each vector either to the sum of the elements (L1-norm) or the sum-of-squares of the elements (L2-norm).

3.4 Machine Learning

Both workflows compared the predictive performance of 11 different ML model types because no single model works best on all types of datasets. Table 2 summarizes the models selected and provides a brief description of their fundamental theory of operation. The references provided expand on the details of their mathematical formulation and their practical implementation in a programming language such as Python.

Hyperparameter tuning is the process of iteratively adjusting various model parameters to maximize the predictive performance. Models that have regularization parameters allow for adjustments to generalize on new data by preventing overfitting to the training data. Examples of hyperparameters for tree-based models include the number of attributes to select for partial tree growing and the minimum number of data instances in a leaf node when growth stops. The ML and output layers of the two workflows are identical, except for the hyperparameter tuning of the BOW model in the feedback loop of the second workflow.

Table 2: Machine Learning Implemented in the Workflows

Model	Description	Reference
Logistic Regression (LR)	Fits the data to a logistic function of the linear combination of attributes to estimate the probability of a binary class.	Géron (2017) (Géron, 2017)
Support Vector Machine (SVM)	Finds a hyperplane in multidimensional feature space that maximally separates the classes.	Géron (2017) (Géron, 2017)
Stochastic Gradient Descent (SGD)	Fits a linear multivariate function to the data by randomly selecting data instances to calculate parameter updates that minimize a selected loss function.	Géron (2017) (Géron, 2017)
Decision Tree (DT)	Grows a logic tree by recursively splitting nodes to maximize the purity of child or leaf nodes.	Géron (2017) (Géron, 2017)
Random Forest (RF)	Grows many shallow and partial decision trees by randomly selecting a subset of attributes and data subset to split nodes, and then uses majority vote to predict the class.	Hastie et al. (2016) (Hastie, Tibshirani, & Friedman, 2016)
AdaBoost (ADB)	Sequentially build shallow decision trees (stumps) that improve on the prediction errors of previous trees, and then uses majority vote to predict the class.	Hastie et al. (2016) (Hastie, Tibshirani, & Friedman, 2016)
Multi-layer Perceptron (MLP)	A feed-forward and fully connected artificial neural network that learns a function with one or more inner layers of neurons.	Géron (2017) (Géron, 2017)
Naïve Bayes (NB)	Uses Bayes probability theory to predict a class given the observed set of features, and assuming that they are independent.	Jame et al. (2013) (James, Witten, Hastie, & Tibshirani, 2013)
k-Nearest Neighbors (kNN)	Predicts a class based on the majority vote of its k-nearest neighbors in feature space.	Jame et al. (2013) (James, Witten, Hastie, & Tibshirani, 2013)
Gradient Boosting (GB)	Sequentially build improved models to predict the errors or residuals of previous models.	Natekin & Knoll (2013) (Natekin & Knoll, 2013)
Extreme Gradient Boosting (XGB)	A highly configurable version of gradient boosting that incorporates regularization.	Chen & Guestrin (2016) (Chen & Guestrin, 2016)

The *cross-validation* (K-fold) procedure cyclically partitions the data into *k* subsets so that the model trains on the union of *k*-1 subsets and tests on the remaining subset until all subsets participate in the testing exactly once. The reported performance is the average value of the test metric across all folds.

The *performance evaluation* procedure used five test metrics derived from the true positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) rates of the classification models. The five metrics used were the classification accuracy (CA), precision (PR), recall (RC), F1-score, and AUC score. The CA is the proportion of correct predictions. The PR is the proportion of correct positive predictions, where $PR = TP/(TP + FP)$. The *RC* is the proportion of positive predictions recalled from the true positive instances, where $RC = TP/(TP + FN)$. *F1* is the harmonic mean of the PR and RC scores, where $F1 = TP/(TP + \alpha)$ and $\alpha = (FN + FP)/2$. The advantage of these performance metrics is their simplicity while providing different insights about how noise affects the *sensitivity* and *specificity* of a model. However, a heavy class imbalance can lead to misleading results. For example, an unskilled classifier that always predicts the majority class can appear to provide good performance. The AUC is another metric that is insensitive to data imbalance by integrating TP as a function FP across a range of sensitivity thresholds (Fawcett, 2006). The AUC stands for "area under the curve" of the receiver operating characteristic (ROC), which evolved from information theory in radio communications. The "curve" is a two-dimensional plot of the TP rate against the FP rate, both as a function of the class membership probability threshold. An unskilled classifier typically produces a value close to 0.5, which establishes a baseline performance to evaluate skilled classifiers.

3.5 Feature Ranking

The Shapley additive explanations used in game theory can explain the contribution of players towards the outcome of a stochastics process with certain rules (Štrumbelj & Kononenko, 2014). The output of the model is a SHAP value that represents the normalized contribution of a feature. The workflows of Figure 2 and Figure 3 adopted the SHAP value to quantify the contribution of

a feature towards predicting the target class. The SHAP value of a feature is the average of its marginal contribution across all permutations of the collective contribution from the subset of other features. Given that *S* is a subset of *N* features of the predictive model and that *v*(*S*) is the value of their collective contribution towards a prediction, the marginal contribution of feature $\{i\}$ is the contribution difference $\mathrm{v}(S \cup \{\mathrm{i}\}) - \mathrm{v}(S)$ after including $\{i\}$ as a feature. The SHAP value ϕ of feature $\{i\}$ is the weighted sum of its marginal contribution over all possible combinations of selecting *S* features from the set of *N* features. That is,

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)] \tag{1}$$

where the notation |*S*| represents the number of non-zero entities and "!" is the factorial operator. The above expression represents the *global* SHAP value for a feature, thus ranking its importance as a contributor towards predicting the target class.

The *local* SHAP value for feature $\{i\}$ when adding instance *j* is

$$\phi_i^{(j)} = h_i(x_j) - E[h_i(x)] \tag{2}$$

where $h_i(x)$ is the *marginal* prediction model for feature $\{i\}$ and *E* is the expected value operator. That is, the local SHAP value is the difference between how much the candidate feature of an instant contributes towards predicting the target class and the expected marginal value of that feature.

4 Results

The subsections of this section mirror those of the methodology section to discuss the results of applying the two workflows to the data. The last two subsections discuss the results of the predictive classification and the feature ranking methods.

4.1 Models from Structured Data

The following two subsections describe the results of the data processing and feature extraction to build models from the structured attributes.

4.1.1 Data Processing

The *data filtering* procedure removed 35 attributes that had more than 85% of their values missing or filled with zeros, and 8 attributes that contained duplicate information. The procedure also removed 12 attributes that were irrelevant to the classification target, based on evaluating the SHAP values via the feedback loop of the workflow. Examples of irrelevant attributes included train numbers, car numbers, and other numerical identifiers. The data filtering procedure also removed 19 attributes that were highly correlated or redundant with others.

The *feature engineering* procedure combined four fields that specified the number of engineers, firemen, conductor, and brakemen on the train into a single field (HUMANS) to quantify the number of human operators (excluding passengers) present. The method also simplified the categories of consist type (CONSIST) and operating method (MOVEx) by reducing them from 14 to 6, and from 21 to 5, respectively.

The *data imputation* procedure did not merely fill missing values using the mean, most frequent, or nearest neighbor in feature space. Rather, the procedure applied more intelligence by filling missing values for track density (TRK_DEN) and the signalized territory flag (SIG) by using the mean and most frequent non-missing values, respectively, for the track class near the closest location, which was encoded in the field STATION. The procedure also filled missing values for CONSIST by inference from the values of other attributes such as the number of loaded or empty freight or passenger cars, the type of railroad (freight or passenger), or the method of operation, for example, yard movements, maintenance, or mainline operation. The

geospatial data cleaning method previously described filled missing or erroneous geospatial coordinates, which accounted for more than 19% of those attributes.

In summary, the data processing procedures described above reduced the original 145 attributes of the 26,943 instances in the dataset to 42 relevant attributes. The number of data instances dropped to 25,035 after removing those with missing target labels.

4.1.2 Feature Extraction

The *attribute transformation* procedure used the shifted log function to transform the track density, train weight, and train speed. The procedure normalized the number of loaded cars (CARS_LD) and cars carrying hazardous materials (CARS_HZMT) to the number of cars on the train (N_CARS). In summary, the attribute transformation procedure further reduced the number of attributes from 42 to 38.

Table 3 summarizes the final set of attributes selected for ML after applying the first workflow. The *numerical normalization* procedure converted all continuous and ordinal feature types to the [0, 1] range. The *one-hot-encoding* procedure converted all categorical attributes to a binary feature array that indicated the presence or absence of a category for each data instance. The categorical variable encoding procedure increased the number of features from 38 to 74 features.

The dispersion column of Table 3 indicates the amount of spread in the distribution of each attribute.

Table 3: Summary of the ML Attributes, their Dispersion, and Type

Attribute	Dispersion	Type	Description
HC	0.672	Binary	Target attribute: 1 if the accident type was human caused
REGION	1.980	Categorical	FRA region code for accident location
LAT	0.133	Continuous	Cleaned latitude coordinate
LON	-0.129	Continuous	Cleaned longitude coordinate
CLASS_RR	0.818	Ordinal	Cleaned railroad class
MONTH	0.541	Ordinal	Incident month
DAY	0.557	Ordinal	Incident day
HR24	0.562	Continuous	Transformed time to fractional 24-hour
TEMP	0.382	Continuous	Temperature (degrees Fahrenheit)
VISION	1.130	Categorical	Visibility: {Dawn, Day, Dusk, Dark}
WEATHER	0.952	Categorical	Weather: {Clear, Cloudy, Rain, Fog, Sleet, Snow}
TRK_TYP	1.010	Categorical	Track Type: {Main, Yard, Siding, Industry}
TRK_CL	0.755	Ordinal	Track Class: {X as 0, 1 through 9}
CWR	1.280	Binary	1 if the rail type was continuously welded, 0 otherwise
SIG	1.855	Binary	1 if used signals to control train movements, 0 otherwise
MOVEx	1.120	Categorical	Movement: {Blocks, Control, Signal, Not Main, Restrict}
TRK_DEN_LG	1.027	Continuous	log(1+x) of annual track density in millions of gross tons
TONS_LG	0.846	Continuous	log(1+x) of gross tonnage, excluding power units
TRNSPD_LG	0.606	Continuous	log(1+x) of train speed in miles per hour (mph)
SPD_OVR	-1.335	Continuous	Difference between train speed and limit for track class
CONSIST	1.080	Categorical	Consist: {Freight, Passenger, Locomotive, Cars, Work, Yard}
HUMANS	0.579	Continuous	Number of humans present on the train
HEADEND1	0.757	Ordinal	Number of headend locomotives
N_CARS	0.998	Ordinal	Total number of cars (sum of loaded + empty cars)
CARS_LD	0.766	Continuous	Proportion of the number of cars that were loaded (0 to 1)
CARS_HZMT	2.772	Continuous	Proportion of loaded cars carrying hazardous materials (0 to 1)
CARS	3.336	Ordinal	Number of cars carrying hazardous materials
CARSHZD	21.950	Continuous	Number of cars that released hazardous materials
ACC_TYPE	1.290	Categorical	The type of accident {derail, collide, obstruct, etc.}
CARSDMG	4.975	Continuous	Number of hazmat cars damaged or derailed
POSITON2	4.863	Continuous	Position of car on the train that caused the accident
EMPTYF2	2.926	Continuous	Number of empty freight cars that derailed
LOADF2	2.253	Continuous	Number of loaded freight cars that derailed
HEADEND2	3.340	Continuous	Number of headend locomotives that derailed
POS_CAR	0.923	Continuous	Relative position of the first involved car in the train
LOADED_1	0.929	Binary	Is first involved car loaded?
ACCDMG	3.552	Continuous	Total reported damage in U.S. dollars
CASKLD	9.574	Continuous	Total killed for all involved railroads
CASINJ	20.339	Continuous	Total injured for all involved railroads

The measure of dispersion for the continuous, ordinal, and binary attributes was the *coefficient of variation*, which is the ratio of the standard deviation to the mean of the distribution for that attribute. The measure of dispersion for the categorical attributes was the *entropy*, which is the sum of products of the probability of occurrence for each category of the attribute. The amount

of dispersion is proportional to the amount of information that an attribute contributes, so it was the first criteria for initial inclusion for performance evaluation in the logic loop of the workflow.

4.2 Models from Unstructured Data

The following two subsections describe the data processing and text mining results from the second workflow after applying it to the unstructured features.

4.2.1 Data Processing

The FRA database contained an accident narrative in 15 fixed-length fields. The procedure combined the text from the 15 narrative fields into a single document associated with each record. The procedure then stripped the documents into a corpus and passed it to the text mining layer.

4.2.2 Text Mining

The first procedure of the text mining layer applied a lowercase transformation to the corpus so that the downstream procedures recognized any case combination of the same word. Subsequently, a noise filter removed all punctuation and digits before feeding the result into a tokenizer that produced a list of single words for each document. The Porter stemming algorithm then reduced all inflected forms of each word to their lexical root so that the next procedure could effectively remove all *stop* words and their variants. The subsequent outlier and common word filters removed word stems that appeared in fewer than 10% and more than 90% of the documents, respectively. In summary, the collection of text mining procedures reduced the number of unique words and symbols from 53,173 to only 32 relevant word stems for embedding using the BOW model. Figure 4a and Figure 4b shows a word cloud before and after the text mining, respectively. The font size of each word or word stem in the cloud is proportional to their frequency of occurrence in the corpus. It is evident that *stop* words such as “was,” “to,”

"the," "and," and "of" dominated the unprocessed corpus and that the subsequent text mining procedures effectively distilled the most relevant words.

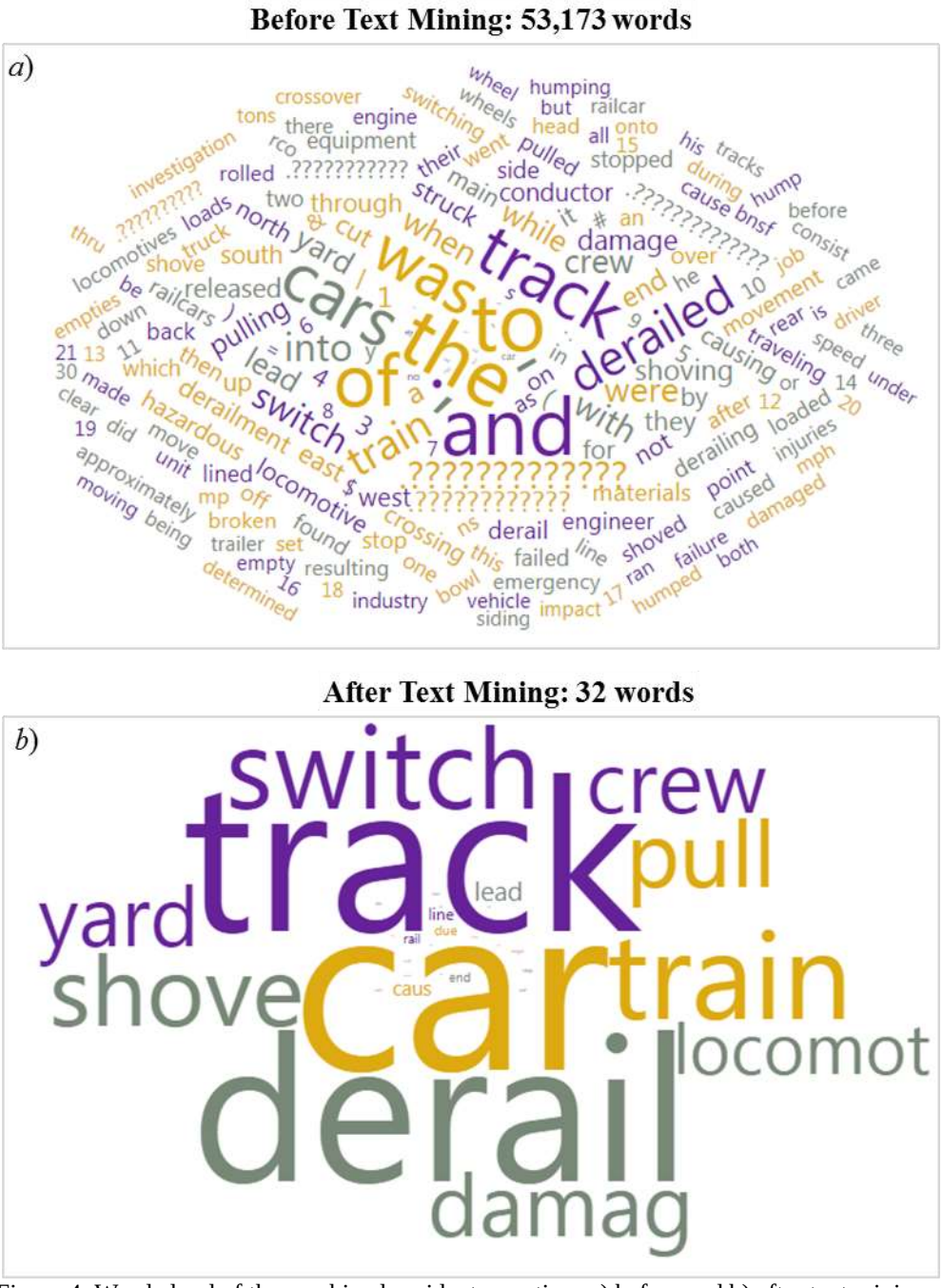

Figure 4: Word cloud of the combined accident narratives a) before and b) after text mining.

4.3 Machine Learning

Table 4 summarizes the performance of each ML algorithm after hyperparameter tuning and learning from the structured features. The performance is in descending order of the AUC score.

Table 4: Model Performance and Optimum Hyperparameter Settings for Fixed Attribute Learning

Model	AUC	CA	F1	PR	RC	Optimum Hyperparameters
RF	0.872	0.783	0.709	0.758	0.666	Trees (N): 60, Attributes/Split: 5, Min Subset: 5
XGB	0.870	0.783	0.719	0.741	0.698	γ:0, Max Depth: 6, Min Child Weight: 1, R:1, w:1, L:0.2
GB	0.864	0.775	0.707	0.734	0.681	LF: LR, Trees (N): 100, L: 0.2, Min Samples Leaf: 1
MLP	0.820	0.735	0.734	0.734	0.735	Hidden Nodes: 100, Activation: ReLu, OA: Adam (α:10-4)
DT	0.816	0.733	0.731	0.730	0.733	Max Depth: 10, Min Samples Leaf (N): 90, Min Subset: 5
LR	0.814	0.730	0.725	0.726	0.730	R (L2, C:5)
SGD	0.811	0.729	0.724	0.725	0.729	LF: (LR, ε:1), R: E.Net (α:10-5, 0.15), L: IVS (η0:10-2, t:0.25)
kNN	0.781	0.711	0.703	0.707	0.711	N: 30, Distance (Euclidean, Weights: Uniform)
NB	0.744	0.664	0.667	0.693	0.664	No parameters to tune
ADB	0.701	0.714	0.714	0.713	0.714	Trees (N): 50, LF: Linear, OA: SAMME.R, LR: 1.0
SVM	0.666	0.591	0.588	0.655	0.591	Kernel: Sigmoid, R (C:0.2, ε:1.0)
Null	0.500	0.603	0.453	0.363	0.603	No parameters to tune

The optimum hyperparameter settings shown include values for the learning rate (L), loss function (LF), regularization (R), and optimizer algorithm (OA) where applicable.

Table 5 summarizes the performance for the ML models built from the text-mined features. The performance is in the order of the AUC score.

Table 5: Model Performance and Optimum Hyperparameter Settings for Text Attribute Learning

Model	AUC	CA	F1	PR	RC	Optimum Hyperparameters
RF	0.889	0.822	0.821	0.821	0.822	Trees (*N*): 60, Attributes/Split: 5, Min Subset: 5
MLP	0.841	0.775	0.773	0.773	0.775	Hidden Nodes: 100, Activation: ReLu, OA: Adam (α:10^{-4})
ADB	0.837	0.788	0.787	0.787	0.788	Trees (*N*): 50, LF: Linear, OA: SAMME.R, LR: 1.0
kNN	0.814	0.744	0.732	0.746	0.744	*N*: 30, Distance (Euclidean, Weights: Uniform)
DT	0.794	0.736	0.730	0.732	0.736	Max Depth: 10, Min Samples Leaf (*N*): 90, Min Subset: 5
LR	0.792	0.731	0.722	0.729	0.731	R (L2, C:5)
SGD	0.786	0.728	0.724	0.724	0.728	LF: (LR, ε:1), R: E.Net (α:10^{-5}, 0.15), L: IVS (η_0:10^{-2}, *t*:0.25)
NB	0.781	0.721	0.718	0.717	0.721	No parameters to tune
No-skill	0.500	0.603	0.453	0.363	0.603	No parameters to tune
SVM	0.444	0.609	0.474	0.654	0.609	Kernel: Sigmoid, R (C:0.2, ε:1.0)

The list excludes the boosting models of XGB and GB because their performance was similar the other boosting model (ADB). The RF model was the top performer in both cases of learning from the structured and unstructured features. The *null* model was an unskilled classifier that predicted the dominant class each time and served as a baseline for comparison with the skilled classifiers. The AUC performance of the null classifier was lowest as expected. As previously described, the CA score for the null model reflected the class imbalance of 60.3% for human-caused accidents.

4.4 Feature Ranking

The feedback logic of the workflow flagged irrelevant or noisy features, based on their low SHAP values, for elimination in previous layers. Figure 5 and Figure 6 are "bee-plots" to visualize the feature's impact on the predictive performance of the RF model. The diagrams list the top 18 features vertically in the order of their *global* impact. Each point in the dot cloud represents the *local* impact of that feature for a single training instance. The horizontal position of a point represents the SHAP value corresponding to the feature value for that instance. Like a binned histogram, the height of the dot cloud is proportional to the number of instances with feature values that correspond to the SHAP value on the horizontal axis. The color of each point represents the normalized value of that feature across all values in the dataset.

For the structured features, the top contributors to predictive performance were 1) the category of derailment-type accidents, 2) the position of the first involved car, and 3) the number of loaded cars that derailed. Binary features such as one-hot-encoded categorical features had either a high value of one (red color) when the attribute was present or a low value of zero (blue color) when the attribute was absent.

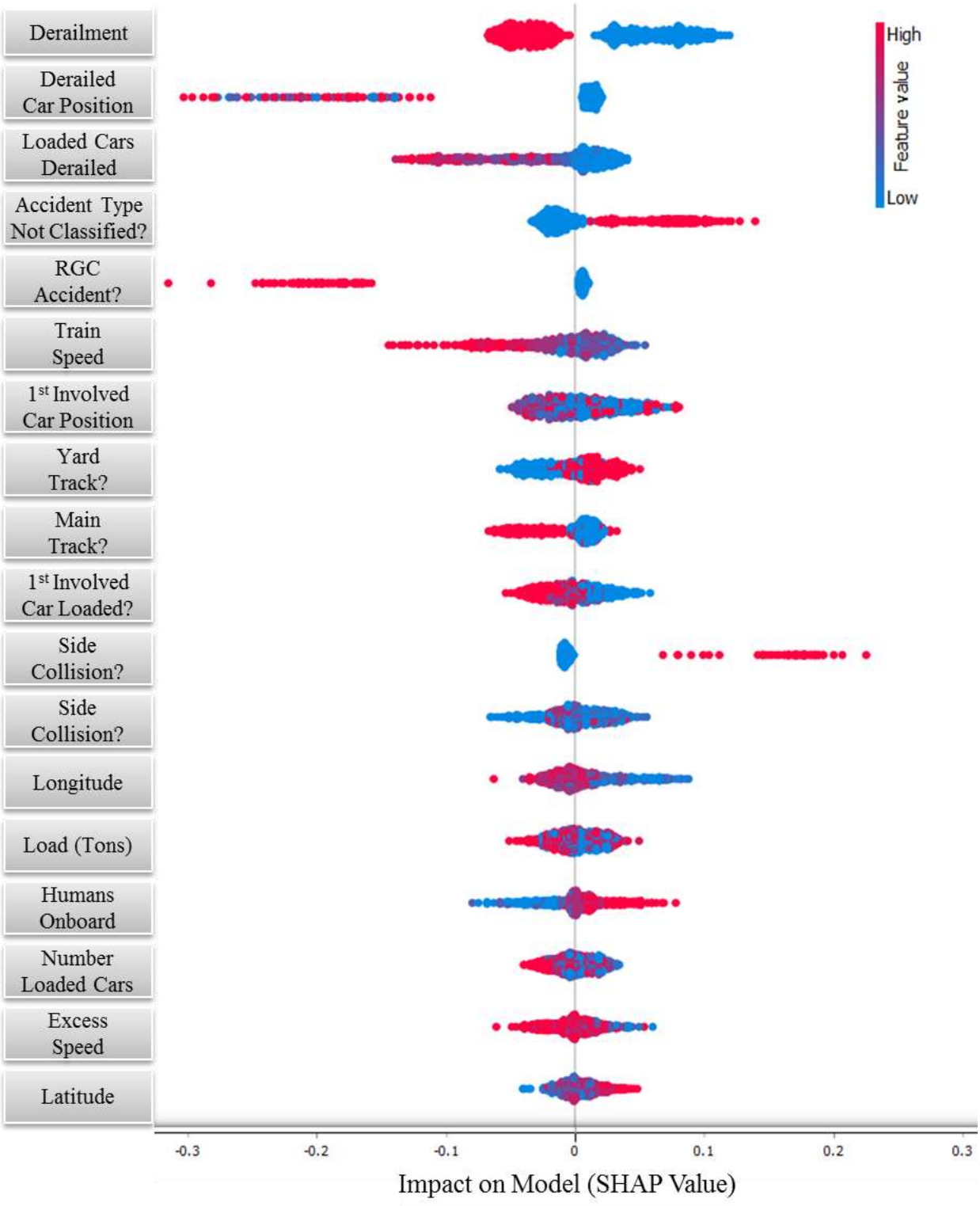

Figure 5: Feature ranking on model impact for fixed attribute learning.

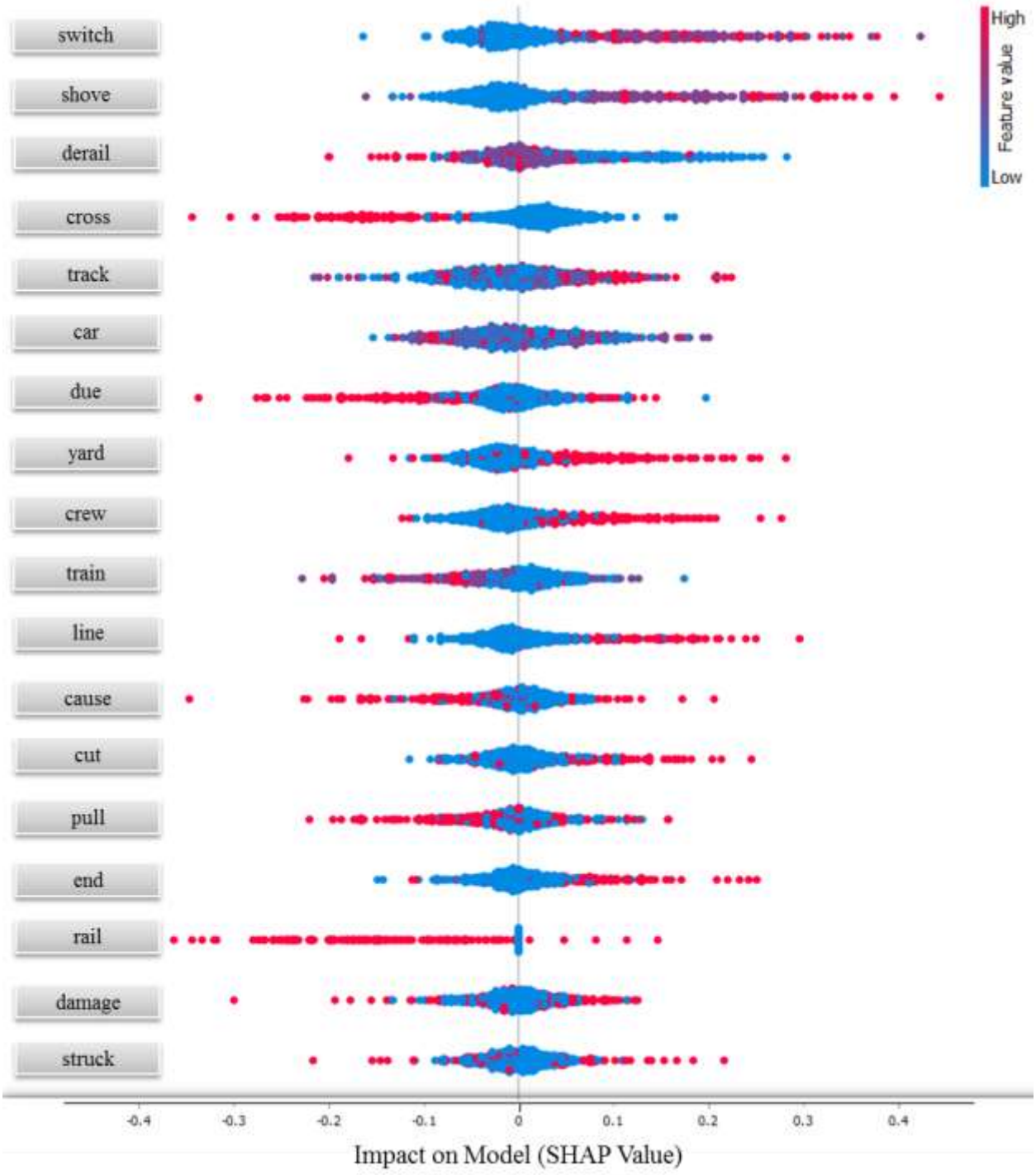

Figure 6: Feature ranking on model impact for text attribute learning.

For example, low values for "derailment" meant that the accident type category was *not* derailment, and that the influence was toward predicting the target class of human-caused accidents.

For the unstructured features, the top three contributors to predictive performance were the words “switch,” “shove,” and “derail.” For instance, the presence of words “switch” and “shove” in an accident narrative contributed more to the prediction of human-caused accidents than when those words were absent.

5 Discussion

The random forest ML model provided the best predictive performance for both workflows. The AUC score for models using the structured and unstructured features were 0.872 and 0.889, respectively. Interestingly, the CA of 82.2% was higher for the ML model using text-mined features. Analyst may interpret the AUC score as a confidence level in the ranking of features that contribute towards predicting the target class of human-caused accidents. The results from the best predictive model built from the structured features suggest that human-caused railroad accidents are associated with the following:

1) Non-derailment and non-RGC type accidents
2) Cars near the front of the train
3) Unloaded cars
4) Low train speed or speeds below the speed limit for the track class operated on
5) Yard or non-main tracks
6) Side collisions
7) Presence of humans onboard.

Interestingly, the three top features (derailment, derailed car position, loaded cars derailed) collectively suggest that human factors are *not* generally associated with derailment type accidents. The fourth ranking feature (accident type not classified) suggests that human-caused accidents tended to be associated with cases where the accident type was not specified. The fifth

ranking feature suggests human-caused accidents are generally not associated with RGC type accidents. Counter to prior beliefs, the sixth ranking feature suggests human-caused accidents are generally not associated with high train speeds. In sharp contrast to derailments, side collisions are more strongly associated with human-caused accidents.

The results from the best model built with text-mined features suggest that human-caused accidents are generally associated with narratives containing the words “switch,” “shove,” “yard,” “line,” “crew,” “cut,” and “struck,” but are not associated with narratives containing the words “derail” or “cross.” The interpretations are that human-caused accidents are generally associated with:

1) activities such as shoving, switching, or cutting cars from trains, more than with pulling (“pull”) cars,
2) non-derailment type accidents,
3) non-crossing type accidents,
4) collisions (“struck”) more than other accident types,
5) yard and line tracks more than with main tracks, and
6) the presence of a crew.

The following findings are interesting:

1) Shoving cars was more associated with human-caused accidents than pulling cars, an insight that only the NLP learning models provided.
2) Agreement between the two ML workflows that human-caused accidents are generally not associated with derailments or crossings.
3) Agreement between the two ML workflows that human-caused accidents are generally associated with yard and non-main tracks.

4) Agreement between the two ML workflows that human-caused accidents are generally associated with the presence of a crew, which is intuitively sound.

The last three findings demonstrated that the information contained within the structured and unstructured data was consistent.

The worst-performing model for both types of predictive classification was SVM. This result suggests that there were no clear hyperplanes in the dataset that separated human-caused accidents from other accident causes. Some evidence of this lack of separation between the classes is observable by the clustering of instances or a lack of gap near the zero SHAP value for most features.

One limitation of this ML approach is the difficulty of automating the processes to continuously improve the predictions with new data. Another limitation is the need to involve a data scientist in feature engineering and to interpret the results, both of which requires more art than science. Some of the findings of this analysis were intuitive, but others were not. Hence, it becomes challenging to formulate precise risk management strategies based on the outcome of predictive analysis, which itself is probabilistic in nature. Nevertheless, some of the distinct findings, such as that shoving cars is riskier than pulling cars when a crew is involved, and that human factors are more pronounced with yard accidents and not with derailments, can inform policies to minimize the risk of such future accidents.

Railroads can use the above findings to guide management decisions, strategic planning, and policy development aimed at mitigating the risks associated with human-caused railroad accidents in the following ways:

1. Enhanced Training: develop comprehensive training programs that focus on activities with higher risk such as shoving, switching, or cutting cars from trains.

2. Standard Operating Procedures: implement clear operating procedures and safety protocols for high-risk activities such as those found in this study.
3. Improved Supervision and Monitoring: invest in technologies and systems to help improve the efficiency and effectiveness of supervising and monitoring crew performance, especially during high-risk activities. Such investments could include real-time monitoring systems, solutions to leverage an installed positive train control system to reduce dependence on human decision-making, periodic safety audits, and regular site inspections.
4. Infrastructure and Track Layout Optimization: invest in improving yard and non-main tracks that are associated with high-risk operations by improving track layouts, upgrading yard facilities, and implementing automatic switches and warning systems.
5. Focused Accident Prevention Policies: develop targeted policies that address the specific types of accidents found to be more strongly associated with human factors. Such policies may include additional safety measures, enhanced protocols for train operators to follow during high-risk situations and encouraging open communications about safety concerns.
6. Crew Resource Management: adopt crew resource management (CRM) techniques to foster a culture of safety and teamwork by improving communication, coordination, and decision-making among crew members. Additional considerations for CRM could include fatigue management and fatigue awareness training.
7. Periodic Evaluations and Revisions: establish regular evaluations and revisions of safety policies and procedures to update them based on reports of best practices, new regulations, emerging trends, and insights gained from studies like this.

6 Conclusions

The emphasis on high-profile accidents due to speeding and switches placed in the wrong position diverted attention from other human errors that have caused many accidents. This study developed two different workflows to compare how they rank features associated with human-caused accidents. One workflow used features extracted from the structured data and the other used text-mined features from the unstructured data. Among 11 different ML models evaluated, the random forest (RF) technique provided the best predictive performance in both workflows. However, the ML workflow that incorporated NLP provided a slight performance edge as well as additional insights that the structured attribute workflow did not.

One surprising result of the analysis was that human-caused accidents were generally *not* associated with high train speeds or derailment-type accidents. Another interesting finding was that shoving cars is riskier than pulling cars, especially when a crew is involved with such activities. These and other findings detailed in the discussion section above can inform management decisions and policies to minimize the risk of such accident types. The discussion above provided several examples of how railroads can use the insights gained from this study to improve railroad safety and risk management.

The workflows presented are easily generalizable to analyze other types of classification problems. The NLP workflow may require even fewer modifications when dealing with English narratives. The Shapely feature ranking technique and the associated data visualization instrument are also reusable without much modification. Future work will leverage the workflow to examine trends in accidents caused by human error to determine the effectiveness of PTC deployments relative to expectations. Future work will also examine influences in human-caused accidents for different track types.

7 References

Abidin, N., Ismail, A., & Emran, N. (2018). Performance Analysis Of Machine Learning Algorithms For Missing Value Imputation. *International Journal of Advanced Computer Science and Applications, 9*(6). doi:10.14569/IJACSA.2018.090660

Aggarwal, C. C. (2015). *Data Mining.* New York, New York, United States of America: Springer International Publishing.

Bala, M., & Bhasin, A. (2018). A Review on Analysis of Railway Traffic Accident with Data Mining Techniques. *International Journal of Computer Sciences and Engineering, 6*(6), 1251-1256. doi:10.26438/IJCSE/V6I6.12511256

Brown, D. (2016). Text Mining the Contributors to Rail Accidents. *IEEE Transactions on Intelligent Transportation Systems, 17*(2), 346-355. doi:10.1109/TITS.2015.2472580

Catelani, M., Ciani, L., Guidi, G., & Patrizi, G. (2021). An enhanced SHERPA (E-SHERPA) method for human reliability analysis in railway engineering. *Reliability Engineering and System Safety, 215*. doi:10.1016/j.ress.2021.107866

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 785-794). doi:10.1145/2939672.2939785

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861-874. Retrieved from http://people.inf.elte.hu/kiss/11dwhdm/roc.pdf

FRA. (2011). *Rail Equipment Accident/Incident Data File Structure and Field Input Specifications.* Washington, D.C.: Federal Railroad Administration (FRA). Retrieved November 25, 2020, from https://safetydata.fra.dot.gov/OfficeofSafety/publicsite/DownloadFStructure.aspx

Gao, L., Lu, P., & Ren, Y. (2021). A deep learning approach for imbalanced crash data in predicting highway-rail grade crossings accidents. *Reliability Engineering and System Safety, 216*. doi:10.1016/j.ress.2021.108019

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

Haleem, K., & Gan, A. (2015). Contributing factors of crash injury severity at public highway-railroad grade crossings in the U.S. *Journal of Safety Research, 53*, 23-29. doi:10.1016/J.JSR.2015.03.005

Hastie, T., Tibshirani, R., & Friedman, J. (2016). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). New York, New York: Springer.

Iranitalab, A., & Khattak, A. (2020). Probabilistic classification of hazardous materials release events in train incidents and cargo tank truck crashes. *Reliability Engineering & System Safety, 199*, 106914. doi:10.1016/J.RESS.2020.106914

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R* (Vol. 112). New York: Springer. doi:10.1007/978-1-4614-7138-7

Jones, K. S., & Willett, P. (Eds.). (1997). *Readings in information retrieval.* Burlington, Massachusetts, United States of America: Morgan Kaufmann.

Kyriakidis, M., Simanjuntak, S., Singh, S., & Majumdar, A. (2019). The indirect costs assessment of railway incidents and their relationship to human error-the case of signals passed at danger. *Journal of Rail Transport Planning & Management*, 34-45. doi:10.1016/j.jrtpm.2019.01.001

Liu, J., & Khattak, A. (2017). Gate-violation behavior at highway-rail grade crossings and the consequences: Using geo-Spatial modeling integrated with path analysis. *Accident Analysis & Prevention, 109*, 99-112. doi:10.1016/J.AAP.2017.10.010

Liu, X., Saat, M., & Barkan, C. (2017). Freight-train derailment rates for railroad safety and risk analysis. *Accident Analysis & Prevention, 98*, 1-9. doi:10.1016/j.aap.2016.09.012

Manning, W., & Mullahy, J. (2001). Estimating log models: to transform or not to transform? *Journal of Health Economics, 20*(4), 461-494. doi:10.1016/S0167-6296(01)00086-8

Natekin, A., & Knoll, A. (2013). Gradient Boosting Machines, a Tutorial. *Frontiers in Neurorobotics, 7*(21). doi:10.3389/fnbot.2013.00021

Noguchi, H., Hienuki, S., & Fuse, M. (2020). Network theory-based accident scenario analysis for hazardous material transport: A case study of liquefied petroleum gas transport in japan. *Reliability Engineering and System Safety, 203*. doi:10.1016/j.ress.2020.107107

Panda, C., Mishra, A. K., Dash, A. K., & Nawab, H. (2022). Predicting and explaining severity of road accident using artificial intelligence techniques, SHAP and feature analysis. *International Journal of Crashworthiness, 23*(2), 186-201. doi:10.1080/13588265.2022.2074643

Saunders, W., Mousa, S., & Codjoe, J. (2019). Market basket analysis of safety at active highway-railroad grade crossings. *Journal of Safety Research, 71*, 125-137. doi:10.1016/J.JSR.2019.09.002

Shin, S. U., Lee, S. H., Shin, S. K., Jang, I., & Park, J. (2021). STPA-Based Hazard and Importance Analysis on NPP Safety I&C Systems Focusing on Human–System Interactions. *Reliability Engineering and System Safety, 213*. doi:10.1016/j.ress.2021.107698

Soleimani, S., Leitner, M., & Codjoe, J. (2021). Applying machine learning, text mining, and spatial analysis techniques to develop a highway-railroad grade crossing consolidation model. *Accident Analysis & Prevention, 152*, 105985-105985. doi:10.1016/J.AAP.2021.105985

Song, B., Zhang, Z., Qin, Y., Liu, X., & Hu, H. (2022). Quantitative analysis of freight train derailment severity with structured and unstructured data. *Reliability Engineering & System Safety, 224*, 108563. doi:10.1016/j.ress.2022.108563

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems, 41*(3), 647-665. doi:10.1007/S10115-013-0679-X

Suh, Y. (2021). Sectoral patterns of accident process for occupational safety using narrative texts of OSHA database. *Safety Science, 142*, 105363. doi:10.1016/j.ssci.2021.105363

USCB. (2019). *TIGER/Line Shapefiles Technical Documentation.* Washington, D.C.: United States Census Bureau (USCB). Retrieved from https://www2.census.gov/geo/tiger/TIGER2019/COUNTY/

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. *Accident Analysis & Prevention, 150,* 105835. doi:10.1016/j.aap.2020.105835

Wang, B., Barkan, C., & Saat, M. (2020). Quantitative Analysis of Changes in Freight Train Derailment Causes and Rates. *Journal of Transportation Engineering, Part A: Systems, 146*(11), 4020127. doi:10.1061/JTEPBS.0000453

Williams, T., & Betak, J. (2019). A Comparison of LSA and LDA for the Analysis of Railroad Accident Text. *Journal of Ubiquitous Systems and Pervasive Networks, 11*(1), 11-15. doi:10.5383/JUSPN.11.01.002

Zhang, Z., Liu, X., & Holt, K. (2018). Positive Train Control (PTC) for railway safety in the United States: Policy developments and critical issues. *Utilities Policy, 51*(2018), 33-40.

Zhou, X., Lu, P., Zheng, Z., Tolliver, D., & Keramati, A. (2020). Accident Prediction Accuracy Assessment for Highway-Rail Grade Crossings Using Random Forest Algorithm Compared with Decision Tree. *Reliability Engineering and System Safety, 200*. doi:10.1016/j.ress.2020.106931