Neural Computing and Applications (2022) 34:14253–14274
https://doi.org/10.1007/s00521-021-06780-3

S.I. : EMERGING TRENDS IN AI & ML

Classification and pattern extraction of incidents: a deep learning-based approach

Sobhan Sarkar[1] · Sammangi Vinay[2] · Chawki Djeddi[3,4] · J. Maiti[5,6]

Received: 27 March 2021 / Accepted: 22 November 2021 / Published online: 6 January 2022

Abstract

Classifying or predicting occupational incidents using both structured and unstructured (text) data are an unexplored area of research. Unstructured texts, i.e., incident narratives are often unutilized or underutilized. Besides the explicit information, there exist a large amount of hidden information present in a dataset, which cannot be explored by the traditional machine learning (ML) algorithms. There is a scarcity of studies that reveal the use of deep neural networks (DNNs) in the domain of incident prediction, and its parameter optimization for achieving better prediction power. To address these issues, initially, key terms are extracted from the unstructured texts using LDA-based topic modeling. Then, these key terms are added with the predictor categories to form the feature vector, which is further processed for noise reduction and fed to the adaptive moment estimation (ADAM)-based DNN (i.e., ADNN) for classification, as ADAM is superior to GD, SGD, and RMSProp. To evaluate the effectiveness of our proposed method, a comparative study has been conducted using some state-of-the-arts on five benchmark datasets. Moreover, a case study of an integrated steel plant in India has been demonstrated for the validation of the proposed model. Experimental results reveal that ADNN produces superior performance than others in terms of accuracy. Therefore, the present study offers a robust methodological guide that enables us to handle the issues of unstructured data and hidden information for developing a predictive model.

Keywords Incident prediction · Topic modeling · Deep neural network · Optimization

✉ Sobhan Sarkar
sobhan.sarkar@iimranchi.ac.in

Sammangi Vinay
vsammangi3@gatech.edu

Chawki Djeddi
c.djeddi@univ-tebessa.dz

J. Maiti
jmaiti@iem.iitkgp.ac.in

[1] Indian Institute of Management Ranchi, Ranchi, Jharkhand 834 008, India

[2] Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332, USA

[3] Department of Mathematics and Computer Science, Larbi Tebessi University, Tebessa, Algeria

[4] LITIS Lab, Rouen University, Rouen, France

[5] Department of Industrial and Systems Engineering, IIT Kharagpur, Kharagpur 721302, India

[6] Centre of Excellence on Safety Engineering and Analytics, IIT Kharagpur, Kharagpur 721302, India

1 Introduction

As per the report given by the International Labour Organization (ILO), a total of 2.3 million people died globally in a year because of occupational incidents and diseases including 0.36 million cases of fatalities [1]. Nearly 4% of the total gross domestic product is drained off because of the occupational incidents [2]. In Europe, it is reported by the European Statistical Office (EUROSTAT) that about 3.2% of workers face an incident at work in the European Union [3]. Behind each of the incidents, there is a chain of multiple factors interacting with each other in a specific pattern. If the pattern is identified, the incident outcomes can be predicted. Once the outcomes are predicted, the occurrence of incidents can be minimized. A predictive model, in such circumstances, is playing a key role by identifying the inherent patterns and subsequently predicting the outcomes. Therefore, the use of the predictive model is utmost important in incident analysis and prevention.

In practice, once an incident is taken place, safety professionals narrate the incident event in their own language and log them into the electronic database. Therefore, the exactness of the incident mostly depends on the experience and quality of writing of the personnel who logs information into the system. It is often found that the incident narratives, which are in the form of unstructured texts, remain underutilized or sometimes unutilized since the proper utilization of these unstructured text data for information retrieval demands an extensive human effort. Reviewing the incident narratives during the investigation is extremely time-consuming. In addition, narratives are sometimes written in such a way that useful information can hardly be effectively retrieved, and thus analyzed. In fact, analysis of this kind of data is so difficult that the inherent information in incident texts are mostly ignored, which may result in a biased decision-making. On top of that, a huge amount of unstructured texts along with categorical, numerical or other forms of data collected at industry level put the decision-making in a challenging situation, particularly in terms of prediction. In order to resolve the issue, a number of practitioners and academic researchers have put a lot of efforts by employing machine learning (ML) algorithms for prediction of incidents. In ML, there are predominantly two kinds of approaches used for the purpose of classification: (i) non-tree-based approaches, such as support vector machine (SVM) [4], k-nearest neighbor (k-NN) [5], artificial neural network (ANN) [6, 7], Bayesian Network (BN) [8], and (ii) tree-based approaches, for example, decision tree (DT).

To exemplify, Sorock et al. [9] used 3,686 insurance claims for the analysis of crash incidents. In their experiment, keywords from the accident narratives were used in order to identify the types of pre-crash vehicle activities and types of crash incidents. Lehto & Sorock [10] used Bayesian model to perform similar kind of work by identifying the pre-crash activities and crash types from incident narratives. The experiment showed that the model could learn from a computer search for 63 key terms pertaining ro incident categories. Wellman et al. [11] used fuzzy Bayesian model to classify injury narratives into 13 external causes of injury and poisoning categories. In a similar vein, Noorinaeini & Lehto [12] also used two singular value decomposition (SVD)-Bayesian models and one SVD-regression model to classify injury narratives into external causes of injury and poisoning categories. Their experiments explored that all the three models were capable of learning from human knowledge for classification. In 2007, a notable study by Pons-Porrata et al. [13] showed the development of a topic discovery system based on a new incremental hierarchical clustering algorithm and Testor Theory to extract and classify the implicit knowledge in news streams. Experimental results showed its usefulness and effectiveness in not only topic detection, but also in classification and summarization tool. Brooks [14] used SAS Text Miner to mine free texts of workers' compensation claims and classify into two categories. Their experimental results suggested that text mining can be used as a stand-alone tool for free-text analysis. Fan & Li [15] used text mining to retrieve historical cases from a case library. They showed that natural language-based case document retrieval is superior to the case-based reasoning and more practical for implementation in construction sites. Abdat et al. [16] used Bayesian Network-based model to extract recurrent occupational accident movement with movement disturbances (OAMD) scenarios from narratives. Using this approach, a total of eight scenarios were extracted to describe 143 OAMDs in the construction and metallurgy sectors. In 2014, Sanchez-Pi et al. [17] used ontology-based automatic text classification from unstructured texts in an oil and gas industry. Their proposed approach included text analysis, recognition, and classification of failed occupational health control. Later, in 2016, they extended their ontological concept and made it more domain-dependent [18] for oil and gas industry. Goh & Ubeynarayana [19] used text mining classification techniques to classify a total of 1000 publicly available construction accident narratives into two categories, accident and near miss cases. They employed six machine learning algorithms, namely k-nearest neighbor (KNN), support vector machine (SVM), linear regression (LR), decision tree (DT), random forest (RF), and Naive Bayes (NB). Experimental results showed that SVM is the best algorithm in classification of 251 cases. In addition, it was found that the unigram tokenization with linear SVM performs the best. Zhang et al. [20] used Deep Belief Network (DBN) and Long Short-Term Memory (LSTM) methods on three million accident-related tweets to classify traffic accidents. From the experiments, it is explored that DBN outperforms SVM and supervised Latent Dirichlet allocation (sLDA). Song & Suh [21] used patent analysis using latent Dirichlet allocation (LDA), for extraction of the latent topics and main keywords contained in documents, and network analysis for monitoring change patterns and relations to identify the trends in technology development that prevent the risks of various industrial systems. Apart from these, LDA has been used in different areas, including for pattern extraction from OSHA databases [22], construction reports [23], and investigation reports generated from manufacturing plants [24, 25]. All the reports are prepared in natural languages. Therefore, natural language processing (NLP) is an essential task for accident analysis for the extraction of useful information hidden in texts. Brown [26] used NLP for identification of the contributors to rail accidents from accident narratives and implemented random forest (RF) to check the predictive power of the contributors toward the accident occurrences. In addition, Nenonen [27], in his study, also mentioned that useful information may also be obtained if injury narratives or incident reports are analyzed properly. Moreover, there are a few more interesting applications using DT found in the refinery industry [28], the petrochemical industry [29], railway [30, 31], road [32], and the aviation industry [33]. In summary, it is often seen that the unstructured texts are very important source of information; however, they remain often under-utilized or sometimes unutilized.

These algorithms have been used effectively in different application domains, including shipbuilding [6], mining [34], construction [34], and service [35]. Of them, ANN is found to be very effective due to the inherent features, for example, the learning ability from data, parallel operation, distributed memory, fault tolerance, etc. It is used in a wide spectrum of application domains. For examples, ANN was successfully used with the backward algorithm (BA-ANN) in the prediction of an outburst of coal and gas by He et al. [36]. It was also used to develop an advanced detection system for the prediction of the rating of worker's health in a hot as well as humid conditions in construction sites [37]. The approaches like ANN and SVM attain widespread popularity since they hold a strong theoretical underpinning enabling us in dealing with the complexity of the problem, learning from the historical information, and more importantly, exploring adaptability with nonparametric theory. However, the interpretation of SVM and ANN model is rather difficult.

Although the aforementioned ML algorithms including ANN [25], DT [38], SVM [39] have been used in frequent in accident analysis; however, these algorithms are only capable enough to effectively utilize the existing attributes of the dataset for prediction of accident occurrences. Nevertheless, there exist a number of hidden attributes or factors within the dataset in a different form of data, which can be hardly retrieved and used by the conventional ML approaches [40]. For instance, hidden attributes can be characterized as ones that require linear or nonlinear transformation. The deep learning method, in this case, can be a better choice for the researchers to obtain information from the data, including the attributes present in either explicit or hidden form. The approach utilizes a number of hidden layers to extract the hidden factors underlying within data to better predict the incident outcomes [40]. Deep neural network (DNN), in this case, works extremely well (in particular NLP-based analysis) [41, 42] as it offers advantages of efficient generation of new features from raw data and accurate classification of feature vectors [43].

DNN, proposed in the early 1980s [44] and revamped in 2002 [45], faced training difficulties initially in deep architectures. Later, it was used in a broad spectrum of application areas, including fraud detection [40], dynamic planning of public bicycle-sharing system [46], time series prediction [47], Spark-based computation [48, 49], pattern recognition [50], speech recognition [51], classification [52, 53], image processing [54, 55], and video processing [41]. The main characteristic of this approach is that it can show better classification performance in the case of analysis of complex and a large amount of data [56–58]. More importantly, due to its basic structure, it can perform superior classification tasks. Typically, it consists of a predefined number of layers of cascaded auto-encoder (AE) and a softmax classifier [59], which basically helps DNN to produce joint advantages of efficient attribute generations and accurate classification. These characteristics of DNN help to obtain more advantages than other conventional classification algorithms available in the literature.

There are a large number of optimization-based approaches available in the array of literature. Most of them are not found useful in training a DNN structure due to its aforementioned problems. However, a number of optimization techniques have been proposed recently to deal with the complexity inherent to machine learning approaches, including training of a DNN structure. Of them, the optimization algorithms, for example, gradient descent (GD) [60, 61], stochastic gradient descent (SGD) [62], and conjugate gradient [63] are found to be very useful. In GD approach, the algorithm can easily be used for linear systems. However, it is not usually recommended in the case of high dimensional search space of the optimization task, where a number of local minima exist. Hinton and Salakhutdinov [60] suggested that the GD approach can be useful for training a DNN architecture in such a case where the optimization parameters like weights are initialized with the values close to an optimum solution. In fact, the condition is very difficult to be fulfilled and consequently, this algorithm gets trapped into local minima. Moreover, the speed of convergence of this algorithm is found to be very slow while dealing with a large dataset. In case of high dimensional optimization problems, stochastic GD algorithm is frequently used due to its faster rate of convergence, and easy implementation. Another optimization algorithm, namely root mean square propagation (RMSProp), in this case, produces better performance since it uses the average of the second moments of the gradients (the uncentered variance) [64]. Moreover, a comprehensive review by Ruder suggested that adaptive moment estimation (ADAM) comparatively superior to RMSProp due to its better bias-correction procedure [65].

Based on the review, some issues are identified in occupational incident prediction and analysis domain, which are summarized below.

(i) Hidden information of unstructured text (i.e., brief description) is not used for the extraction of the incident pattern.
(ii) In the domain of incident prediction and analysis, the use of unstructured data (i.e., text) along with categorical data is very little.
(iii) Very little use of the deep neural network (DNN) for the incident prediction using both structured and unstructured data.
(iv) Parameter optimization using adaptive moment estimation (ADAM) is untouched in DNN.

Therefore, to address these issues, our present study endeavors to contribute as follows:

(i) Unstructured text (e.g., incident narratives) along with structured data (i.e., categorical predictor attributes) has been used together to extract the feature vector from each sample corresponding to each occupational incident. All feature vectors constitute together to make the feature map.
(ii) *K*-modes algorithm, missing value handling, and class imbalance handling are adopted to obtain the noise-free feature map.
(iii) ADAM-based DNN is developed for the prediction of the occupational incident.

In this paper, we propose adaptive moment estimation (ADAM)-based DNN (i.e., ADNN) classifier for the prediction of incident outcomes using incident data collected from a steel manufacturing plant. The motivation behind the use of this optimization algorithm on DNN is based on its few advantages. For example, it demands less memory can be used easily and efficiently. In our study, it has been demonstrated that the proposed approach is better than other optimized DNN classifiers, namely RMSProp and SGD-based DNN. In addition, the classification performance of the proposed approach has been tested on five different benchmark datasets from the literature. Finally, a comparative study has been made between the performance of ADNN classifier and that of the other state-of-the-art classifiers, namely SVM, k-NN, and RF to explore the efficiency of our proposed strategy in incident prediction.

The remainder of the paper is structured in the following way: In Sect. 2, the proposed methodology has been discussed in brief. In Sect. 3, a case study is provided with data collection, data description, and data preprocessing steps. Results and discussions of the analyses are presented with the statistical tests in Sect. 4, and finally, in Sect. 5, the conclusion is drawn with the scopes of future studies.

2 Methodology

The proposed methodology comprises four phases which is displayed in Fig. 1.

In Phase-I, common data pre-processing tasks, namely topic modeling, missing data handling, class imbalance handling, outlier handling, and data transformation are performed. In the next phase, i.e., Phase-II, three optimizers, namely SGD, RMSProp, and ADAM are used to tune the parameters of DNN algorithm. In Phase-III, the best classifier is selected through comparative study and in the final phase, i.e., Phase-IV, prediction of incident outcomes using the best classifier is performed. Initially, our models have been over-fitted on the test data. The training accuracies have been found much higher than testing accuracies. To overcome this shortcoming, hyper-parameter tuning of the models has been done using 10-fold cross validation on training dataset following some earlier studies [66–68]. Testing dataset has been used only for the model evaluation. In this cross validation process, first, the dataset has been shuffled randomly. Then, the dataset has been split into 10 groups. One of the 10 groups has been taken as test dataset and rest of them have been taken as training dataset. Then, the model or classifier has been fitted with the training set and evaluated it on the test set based on the performance score. This procedure repeats for 10 times on different test sets and the model performance has been obtained by averaging out the 10 different performance scores. Based on this average performance score, the final model is evaluated. Then, the algorithm with the highest accuracy is considered as the best one. The methods of data pre-processing (i.e., text data handling using topic modeling, missing data handling, class imbalance handling, outlier handling, data transformation from categorical to continuous form), classification (i.e., DNN), and optimization (i.e., SGD, RMSProp, and ADAM) used in this study are briefly discussed in the following section. Some important notations used in this study are mentioned in "Appendix A".

2.1 Data pre-processing

In this section, the five data pre-processing tasks: (i) text data handling, (ii) missing data handling, (iii) class imbalance handling, (iv) outlier handling, and (v) data transformation are discussed below.

2.1.1 Text data handling

The latent Dirichlet allocation (LDA)-based topic modeling is discussed as a data pre-processing tool, which has been used on unstructured text data. To determine the

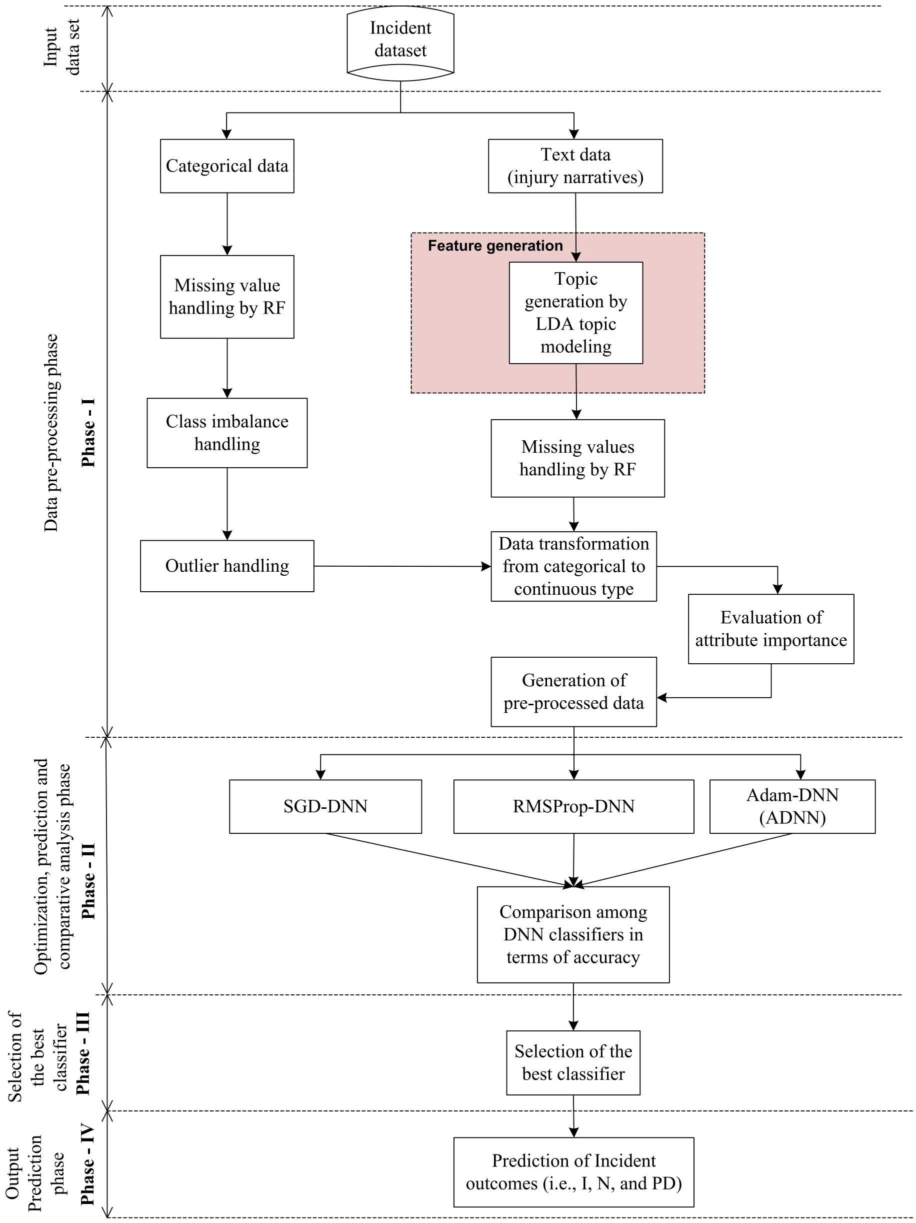

Input data set
Incident dataset
Data pre-processing phase
Phase - I
Categorical data
Text data (injury narratives)
Feature generation
Missing value handling by RF
Topic generation by LDA topic modeling
Class imbalance handling
Missing values handling by RF
Outlier handling
Data transformation from categorical to continuous type
Evaluation of attribute importance
Generation of pre-processed data
Optimization, prediction and comparative analysis phase
Phase - II
SGD-DNN
RMSProp-DNN
Adam-DNN (ADNN)
Comparison among DNN classifiers in terms of accuracy
Selection of the best classifier
Phase - III
Selection of the best classifier
Output Prediction phase
Phase - IV
Prediction of Incident outcomes (i.e., I, N, and PD)

Fig. 1 Proposed research methodological flowchart

optimal number of topics, four metrics have been used in this study. The first metric (say, 'Metric1'), has been developed by Griffiths and Steyvers [69], which helps to determine the optimal number of topics by the calculation of the maximum log-likelihood of the data. Cao et al. [70] have developed another metric (say, 'Metric2') to compute the stability of the structure of a topic using mean cosine distance between every pair of topics. Using this, it has been found that the stability increases with the decrease in mean distance. Likewise, Arun et al. [71] have developed a Kullback–Leibler (KL) divergence-based metric (say, 'Metric3'). The choice of the optimal number of topics depends on the minimum divergence. Of late, Deveaud et al. [72] have developed a heuristic search-based metric (say, 'Metric4') to determine the number of latent concepts within the user's query. It is basically done by maximizing the intra-topics information divergence. choice of the optimal number of topics depends on the minimum divergence. Of late, Deveaud et al. [72] have developed a heuristic search-based metric (say, 'Metric4') to determine the number of latent concepts within the user's query. It is basically done by maximizing the intra-topics information divergence. Metric 1 and 4 are based on word-coherence and Metric 2 and 3 are based on word-log-perplexity. Therefore, considering the Therefore, considering the aforementioned metrics simultaneously, two metrics, i.e., Metric2 and Metric3 are to be minimized, whereas the other two, i.e., Metric1 and Metric4 are to be maximized.

After obtaining the optimal number of topics, LDA-based topic modeling is used on unstructured text. In this process, it is assumed that both document and words are obtained from a generative probability model [73]. Each document is obtained by the model given below.

(i) $N_i \sim$ Poisson distribution (where N_i is a random variable representing the number of words in i-th document)

(ii) $\theta_i \sim$ Dirichlet distribution (α) (where θ_i is a random variable denoting per document-topic proportion, α is a proportion parameter, $\alpha < 1$)

(iii) $\beta_{T_j} \sim$ Dirichlet (η) (where β_{T_j} is a per-topic (say, T_j) word proportion parameter, such that $T_j \in \{T\}$

(iv) $z_{i,w_l} \sim$ Multinomial (θ_i) (where z_{i,w_l} implies the topic of each word $w_l \in \{w\}$ in the i-th document, D_i)

(v) $w_{i,w_l} \sim$ Multinomial $p(w_{i,w_l} \mid z_{i,w_l}, \beta_{T_j})$ (where w_{i,w_l} is the observed word in z_{i,w_l} topic)

According to the LDA theory proposed by [73], the words are generated from the distribution of the topic. Different topics are able to produce similar words. Based on the words generated within each topic, an intuitive meaning can be ascribed to the topic. To estimate the parameters, a joint distribution of observed and latent random variables is used, which can be expressed in the following Eq. (1):

$$p(\beta, \theta, z, w \mid \alpha, \eta) = \prod_{j=1}^{m} p(\beta_{T_j} \mid \eta) \prod_{i=1}^{n} p(\theta_{D_i} \mid \alpha) \left(\prod_{l=1}^{p} p(z_{i,w_l} \mid \theta_{D_i}) p(W_{D_i,w_l} \mid \beta_{1:m}, Z_{i,w_l}) \right) \tag{1}$$

Using LDA topic modeling, a set of optimal number of topics are generated. Here, each topic consists of eight key words that are used as predictors and added with the other conditional predictors. The algorithm for the extraction of topics in terms of key words is defined in Algorithm 1.

Algorithm 1: Key term extraction from unstructured texts.

```
Input: $[T]_{n\times 1}$: The set contains $n$ number of unstructured texts; $t_1$ and $t_2$: The lower and upper limit
    of topic number, respectively.
Output: $[KT]_{m\times 10}$: The set contains 8 key terms under each topic and $m$ is the optimal number of
    topics.
Initialize $[KT] = 0$
Define lemmatized and stemmed word vector $[W]_{n\times 1}$ from $[T]_{n\times 1}$
Arrange $[W]_{n\times 1}$ in ascending order
Define term frequency $[TF]_{n\times 1}$ from $[W]_{n\times 1}$
Define $[TF - IDF]_{n\times 1}$ from $[TF]_{n\times 1}$
Set log perplexity $[LP] \leftarrow \phi$
Set coherence value $[CV] \leftarrow \phi$
for $i = t_1$ to $t_2$ do
    for $j=1$ to $i$ do
        Random selection of $TOPIC[j]$ containing 8 words from $[w]$
    end
    Define log perplexity $(lp)$ using $TOPIC[j]_1^i$
    $LP[i] = lp$
    Define coherence value $(cv)$ using $TOPIC[j]_1^i$
    $CV[i] = cv$
end
for $i = t_1$ to $t_2$ do
    if $LP[i] ==$ min $LP$ and $CV[i]$ = max $CV$ then
        $[KT] = TOPIC_1^i$
    end
end
return $[KT]$
```

2.1.2 Missing data handling

For missing data handling, random forest classifier is used. It uses bootstrap sampling. The observations having missing values in dependent attribute are imputed by randomly drawing from independent attributes. This algorithm is used to fit individual regression tree to a bootstrapped sample and impute or predict each missing value as the prediction of a randomly selected decision tree. Consider a universe $S = <U, A \cup C>$, where U is the information table, A is the set of predictor attributes, and C is the response attribute. Here, U consists of samples: $U = \{x_1, x_2, ..., x_Q\}$, where Q represents the total number of instances in U, $A = \{a_1, a_2, ..., a_p\}$. Here, p denotes the number of predictors in U. For an arbitrary attribute $a_s(s \in$

p) having missing values at entries $i_{mis}^{(s)} \leq \{1, 2, ..., Q\}$, the dataset is separated into four parts, which are as follows:

(i) Observed values of attribute a_s are denoted as $y_{obs}^{(s)}$.

(ii) Missing values of attribute a_s are denoted as $y_{mis}^{(s)}$.

(iii) The attributes other than a_s with observations $i_{obs}^{(s)} = \{1, 2, ..., Q\} \setminus i_{mis}^{(s)}$ are denoted as $x_{obs}^{(s)}$.

(iv) The attributes other than a_s with observations $i_{mis}^{(s)}$ are denoted as $x_{mis}^{(s)}$.

Now, under such condition, missing value imputation is performed using the random forest algorithm. The steps of this process are given below.

- *Step 1:* First, find out the percentage of missing values for each a_s in the dataset.
- *Step 2:* Sort a_s according to the ascending order of the percentage of missing values.
- *Step 3:* For each a_s, the random forest algorithm is trained with predictors $x_{obs}^{(s)}$ and the response attribute $y_{obs}^{(s)}$. Then, the missing values $y_{mis}^{(s)}$ are predicted or imputed using this algorithm, which is tested on $x_{mis}^{(s)}$. This imputation process is repeated until a termination criterion is satisfied. In this study, the user-defined maximum number of iterations is considered as the stopping criterion.

2.1.3 Class imbalance handling

To handle the class imbalance issue in data, Synthetic Minority Over-sampling Technique (SMOTE) algorithm is used. It was proposed by Chawla et al. [74]. It is an oversampling technique. It oversamples the minority class by generating synthetic samples in the imbalanced dataset. Each sample of minority class is considered initially and the new samples are generated along the line segment which connects this sample with its minority nearest neighbor. It usually works by oversampling minority class and undersampling the majority class simultaneously. That is why it can produce better classification performance than only undersampling. The steps of this algorithm used in this study are displayed in Fig. 2.

2.1.4 Outlier handling

The word ‘outlier’ indicates a data object which deviates significantly from the rest of the data. Handling of such data is very important as the existence of such data may negatively impact on the classifier’s performance. To handle this issue, k-modes clustering algorithm has been used in this study [75]. It extends the k-means algorithm to categorical domain by using a suitable dissimilarity measure defined over categorical attributes. The pseudo-code for outlier detection using k-modes algorithm is presented in Algorithm 2.

Algorithm 2: The pseudo-code for outlier detection.

```
Input: $[x]_{m\times n}$: Information table; $\{c_1, c_2, ..., c_n\}$: A set of attributes; $\{CL\}^{\alpha} = \{cl_1, cl_2, ..., cl_k\}$:
       Cluster's center at the iteration $\alpha$; $N$: Total number of iteration taken for the operation
Output: $[OCK]_k$: Cluster center with samples optimum.
Initialize $[K] = 2$
Initialize $[OCK]_k = \phi$
for $\alpha = 2$ to $N$ do
    Initialize $[OK]_{\alpha} = \{cl_1, cl_2, ..., cl_{\alpha}\}$ set of cluster's center
    Add samples from $[x]_{m\times n}$ to $[OK]_{\alpha}$
    $[OK]_{\alpha}^{'}$= Updated cluster's center based on sample information
    Define error $E(OK)_{\alpha}$ for $[OK]_{\alpha}$
    Define error $E(OK')_{\alpha}$ for $[OK]_{\alpha}^{'}$
    if $E(OK)_{\alpha} > E(OK')_{\alpha}$ then
        $[OCK]_{k=\alpha} = [OK]_{\alpha}^{'}$
    end
    else
        $[OCK]_{k=\alpha}$ is the set of optimum cluster's center
    end
    Store all the samples corresponding to the nearest cluster center of $[OCK]_{k=\alpha} = [S_{OCK}]$
end
return $[OCK]_k$
```

2.1.5 Data transformation

After the outlier detection and reduction, the reduced dataset is transformed from its categorical form to the continuous form using a gravity factor (GF)-based normalization technique [6]. In this data processing stage, uncertainty arises as the values of the decision classes are user-defined. GF is a normalized value, which is calculated from the frequency of each of the categories in each attribute. The following Eq. (2) is used to estimate the GF values of the data:

$$GF = \frac{\sum_{i=1}^{n} x_i y_i}{n \times \sum_{i=1}^{n} x_i} \tag{2}$$

, where x_i and y_i denote the percentage of each category of each predictor and a corresponding normalization factor for risk, respectively. The value of n is taken as equal to three since the response attribute 'Incident outcomes' has three classes (i.e., injury, nearmiss, and property damage). The normalization makes the GF values scaled from 0 to 1. The pseudo-code for computing GF values is shown in Algorithm 3.

negative integers. The left and right portions of an AE network are called 'encoder' and 'decoder,' respectively. The inputs of the encoder are the inputs of the AE and its outputs are the inputs of the decoder. The output of the decoder is the output of the AE. If there are outputs $c = [c_1\ c_2\ c_3\ \ldots\ c_N]^T$, activation function (usually, sigmoidal) f, inputs $x = [x_1\ x_2\ x_3\ \ldots\ x_M]^T$, biases $b = [b_1\ b_2\ b_3\ \ldots\ b_N]^T$, and the weights $W = [w_1\ w_2\ w_3\ \ldots\ w_N]^T$, the relationship between input and output in the encoder can be denoted by $c = g_E(b + W^T x)$ and can be expressed as the following Eq. (3) [43]:

$$c = f(b + W^T x) \tag{3}$$

Algorithm 3: Categorical to continuous conversion.

```
Input: $[I]_{n\times m}$: A set of categorical data; $n$: The number of rows; $m$: The number of columns; $C$: The
    number of decision classes in $I$; $\lambda$ = A variable varied from 1 to C, $l$= The $l$-th decision class.
Output: $CONT_{n\times m}$: A set of continuous data
Initialize $CONT_{n\times m} \leftarrow \phi$ for $i = 1$ to $m$ do
    Define pivot table $P[i]_{B\times 1}$ for the $i$-th attribute for $j = 1$ to $m$ do
        $GF[j] = 0$
        $SUM[j] = P[i][j]$
        for $\lambda = 1$ to $C$ do
            $COUNT = P[i][j]$ for $l$-th decision class
            $FREQ = COUNT/SUM[j]$
            $RSK = FREQ + l$
            $GF[j] = GF[j] + RSK$
        end
        $GF[j] = \frac{GF[j]}{C}$
        $[CONT] = GF[j]$
    end
end
return $[CONT]_{n\times m}$
```

2.2 Classification and optimization algorithms

The classification algorithm used in this study is DNN, which is basically a stacked auto-encoder (SAE), consisting of an auto-encoder and a softmax classifier [43]. A brief description of the auto-encoder, SAE, and softmax classifier is given below. In addition, three optimization algorithms, namely SGD, RMSProp, and ADAM are also discussed in this section.

2.2.1 The auto-encoder

The auto-encoder (AE) is basically a feed-forward ANN. It consists of three layers; one input and one output layer, and a hidden layer in between them. The AE is trained in such a manner that the number of nodes at the output layer becomes equal to that of the input layer to map the input space to feature space. In Fig. 3, a network structure with a single hidden layer of ANN is displayed. The number of inputs and outputs are equal, which is equal to M, and the number of hidden nodes is N, where both M and N are non-

Similarly, for the decoder, if there are outputs $\widehat{x} = [\widehat{x}_1\ \widehat{x}_2\ \widehat{x}_3\ \ldots,\ \widehat{x}_M]^T$, activation function (usually, sigmoidal) $\widehat{f}$ at the output layer, biases $\widehat{b} = [\widehat{b}_1\ \widehat{b}_2\ \widehat{b}_3\ \ldots,\ \widehat{b}_M]^T$, and the weights $\widehat{W} = [\widehat{w}_1\ \widehat{w}_2\ \widehat{w}_3\ \ldots,\ \widehat{w}_M]^T$, the relationship between input and output of the decoder is denoted as $\widehat{x} = g_D(\widehat{b} + \widehat{W}^T c)$ and expressed as the following Eq. (4):

$$\widehat{x} = \widehat{f}(\widehat{b} + \widehat{W}^T c) \tag{4}$$

As stated earlier, an auto-encoder consists of two parts: encoder and decoder. Therefore, the input–output relationship of an auto-encoder is denoted as Eq. (5):

$$g_{AE}^1 = g_E \circ g_D \tag{5}$$

, where g_E and g_D denote the function of encoder and decoder, respectively, $\circ$ represents the output of an encoder is fed to a decoder as input.

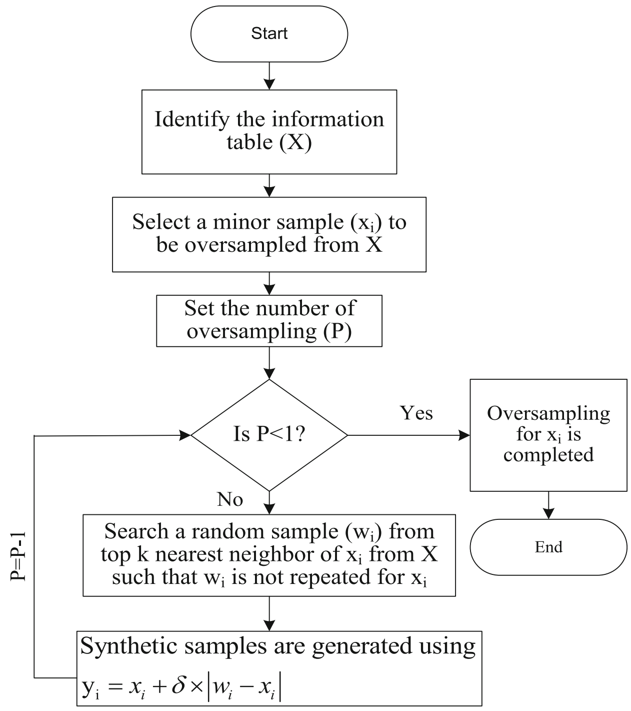

Start
Identify the information table (X)
Select a minor sample (x_i) to be oversampled from X
Set the number of oversampling (P)
Is P<1?
Yes
Oversampling for x_i is completed
End
No
Search a random sample (w_i) from top k nearest neighbor of x_i from X such that w_i is not repeated for x_i
Synthetic samples are generated using
$y_i = x_i + \delta \times |w_i - x_i|$
P=P-1

Fig. 2 Algorithmic flowchart of SMOTE algorithm

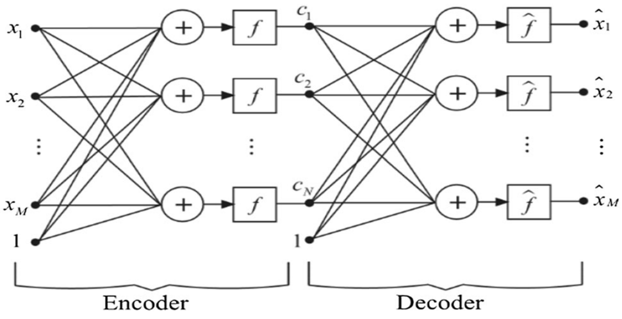


Fig. 3 Auto-encoder network

2.2.2 The structure of a stacked auto-encoder (SAE)

The structure of an SAE is developed by cascading operation that helps to generate a number of AEs (refer to Fig. 4). Let L be number of AEs that are cascaded to form stacked auto-encoder (SAE), and let $g^1_{AE}, g^2_{AE}, \ldots, g^L_{AE}$ be the function of the aforesaid L auto-encoders. Therefore, the operation of SAE can be expressed as the following Eq. (6):

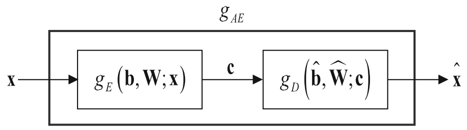

g_{AE}
$\mathbf{x}$
$g_E(\mathbf{b}, \mathbf{W}; \mathbf{x})$
$\mathbf{c}$
$g_D(\hat{\mathbf{b}}, \widehat{\mathbf{W}}; \mathbf{c})$
$\hat{\mathbf{x}}$

Fig. 4 Cascading-based auto-encoder network

$$g_{SAE} = g^1_{AE} \circ g^2_{AE} \circ g^3_{AE} \circ \ldots \circ g^L_{AE} \tag{6}$$

2.2.3 The softmax classifier

This classifier is a linear classifier, which can classify the multiple classes. It is used to handle two classes. The working principle of a softmax classifier depends on the principle of logistic regression [76]. After the development of a DNN classifier, the training is done by using optimization algorithms, namely SGD, RMSProp, and ADAM algorithms, which are described below.

2.2.4 SGD algorithm

SGD is a popular optimization algorithm, which is used to minimize the objective function with model parameters θ. The parameters are updated in the opposite direction of the gradient of $\bigtriangledown_\theta J(\theta)$ [65]. The size of the steps is determined by the learning rate, η. Let θ^{t-1} be the value of the parameters for the $(t-1)$-th iteration. Then, the updated value of the parameters for t-th iteration (where input is x and output is y), defined as:

$$\theta^t = \theta^{t-1} - \eta \times \bigtriangledown_{\theta^{t-1}} J(\theta^{t-1}; x; y) \tag{7}$$

, where $t = 1, 2, \ldots, T$. Here, T represents the total number of iterations. At each iteration, the values of the parameters are updated for every sample present in the dataset. Due to the capability of the SGD of updating the parameters one at a time, it works very faster and can be used in an online settings. Since it shows the frequent update with high variance in the objective function, a high fluctuation is observed, which may help to get the better local minima. This characteristic does not help the algorithm to converge at an exact minimum point. However, decreasing the learning rate slowly may help the algorithm converge to a local or global minimum. The pseudo-code of SGD algorithm is given in Algorithm 4.

Algorithm 4: Pseudo-code of the SGD algorithm.

```
Input: $\epsilon_k$
Output: $\theta$
while termination criterion does not satisfy do
    sample $m$ from $\left\{x^{(1)}, x^{(2)}, ..., x^{(m)}\right\}$ with related to the target $y^{(i)}$
    $\widehat{g} \leftarrow +\frac{1}{m}\nabla_\theta \sum_i J(f(x^{(i)};\theta), y^{(i)})$
    $\theta \leftarrow \theta - \epsilon\widehat{g}$
end
return $\theta$
```

2.2.5 RMSProp algorithm

RMSProp is a GD-based optimization algorithm. The learning rate of this algorithm is adapted for each parameter. To resolve the issues of the *'vanishing gradient'* and entrapment of solution into local optimum, RMSProp algorithm is used for training a DNN algorithm [77]. It uses a moving average of squared gradients. It can balance the step size by decreasing the steps for the large gradient to avoid 'exploding' and by increasing the steps for the small gradient to avoid 'vanishing.' The algorithm weighs the recent past more heavily as compared to distant past. As a consequence, it explores the effectiveness of the optimization algorithm for DNNs. The pseudo-code of RMSProp algorithm is given in Algorithm 5.

Algorithm 5: Pseudo-code of the RMSProp algorithm.

```
Input: $\epsilon, \rho, \theta, \delta$
Output: $\theta$
Initialize $r = 0$
while termination criterion does not satisfy do
    sample $m$ from the training set $\left\{x^{(1)}, x^{(2)}, ..., x^{(m)}\right\}$ with related to the target $y^{(i)}$
    $g \leftarrow +\frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i)};\theta), y^{(i)})$
    $r \leftarrow \rho r + (1-\rho) g \cdot g$
    $\Delta\theta = -\frac{\epsilon}{\sqrt{\delta + r}} \cdot g$
    $\theta \leftarrow \theta + \Delta\theta$
end
return $\theta$
```

2.2.6 ADAM algorithm

ADAM is a gradient-based first-order stochastic optimizer [64]. From the first and second moment estimates of the gradients, it calculates the adaptive learning rates (ALR) of different parameters. The main advantage of this method includes the capability of handling the issue of sparse gradients. There are a few more advantages, for example, it works without a stationary objective, the parameters are updated without depending on the rescaling of the gradient, it has little memory requirement, etc. The algorithm starts with initializing the moving averages (MAs) by setting them at zeros. Let g_t be the gradient of a stochastic objective function J at t-th iteration, and let m_t be the first moment (i.e., the mean of gradients) and v_t be the second moment (i.e., variance of the gradients) at t-th iteration. m_t and v_t are defined as follows:

$$g_t = \bigtriangledown_{\theta^t} J(\theta^t) \tag{8}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{9}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{10}$$

, where m_{t-1} and v_{t-1} denote the first and second moment of gradients at $(t-1)$-th iteration, respectively. The hyperparameters β_1 and β_2 indicate the first and second exponential decay rates for the moment estimates, respectively, and $\beta_1, \beta_2 \in [0,1)$. These parameters basically control the decay rates of the MAs. Since the initialization of MAs starts with zeros, the moment estimates become biased toward zero. In order to counteract the initialization bias, bias-corrected first and second-moment estimates are computed using the following Eqs. (11) and (12), respectively:

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{11}$$

$$\widehat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{12}$$

Later, the parameters are updated using the update rule in ADAM, as given in Eq. (13):

$$\theta^{t+1} = \theta^t - \frac{\alpha}{\sqrt{\widehat{v}_t} + \epsilon}\widehat{m}_t \tag{13}$$

, where ϵ is a very small value. The pseudo-code of the ADAM algorithm is provided in Algorithm 6. The pseudo-code of ADAM-based DNN, i.e., ADNN is provided in Algorithm 7.

Algorithm 6: Pseudo-code of the ADAM algorithm.

```
Input: $\alpha, \beta_1, \beta_2 \in [0,1), f(\theta), \theta_0$
Output: $\theta_t$
$m_0 \leftarrow 0$
$\nu_0 \leftarrow 0$
$t \leftarrow 0$
while $\theta_t$ does not converge do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t$
    $\nu_t \leftarrow \beta_2 \cdot \nu_{t-1} + (1-\beta_2) \cdot g_{t^2}$
    $\widehat{m_t} \leftarrow \frac{m_t}{(1-\beta_1 t)}$
    $\widehat{\nu_t} \leftarrow \frac{\nu_t}{(1-\beta_2 t)}$
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \frac{\widehat{m_t}}{(\sqrt{\widehat{\nu_t}}+\epsilon)}$
end
return $\theta_t$
```

Algorithm 7: The pseudo-code of ADNN algorithm.

```
Input: $X_{M\times N}$: Feature space having $M$ dimension and the number of samples; $N$: The number of
       samples present in the input space for encoder; $b = [b_1, b_2, ..., b_N]^T$: Biases;
       $W = [w_1, w_2, ..., w_N]^T$: Weights;
Output: $\widehat{X}_{M\times N}$: Output of the decoder.
Initialize $[\widehat{X}_{M\times N}] \leftarrow \phi$
Initialize $[C_{M\times N}] \leftarrow \phi$
Train the model parameters of encoder using ADM represented in Algorithm 6
Define $[C_{M\times N}]$ using Eq. (3)
Train the model parameters of decoder using ADM represented in Algorithm 6
Define $[\widehat{X}_{M\times N}]$ using Eq. (4) based on the trained model parameter.
return $[\widehat{X}_{M\times N}]$
```

3 Case study

The data consisting of a total of 9473 incident records have been retrieved from a steel manufacturing plant over the period of 2010 to 2013. After collection of data, they are preprocessed. The dataset comprises a total of thirteen attributes (11 categorical, and two free unstructured text attributes), of which 'Incident outcome' is deemed as the response attribute. A short description of attributes with their percentage of occurrence in the dataset is provided in Table 1. After the data collection, they have been preprocessed. In pre-processing, some basic tasks have been

Table 1 Attributes with percentage of occurrence in the dataset.

Attribute	Description	Category (%)
Day (DOI)	The attribute denoting the day of the week when the incident took place, has a total of seven classes, i.e., Sunday to Saturday	Sun (8.3%), Mon (15.7%), Tue (15.3%), Wed (15.1%), Thu (15.5%), Fri (15.5%), and Sat (14.5%)
Month (MOI)	The attribute indicating the month of a year when the incident occurred, has twelve classes, i.e., January to December	Jan (5.5%), Feb (5.3%), Mar (6%), Apr (10.8%), May (10.5%), Jun (10.9%), Jul (11.3%), Aug (9.3%), Sep (7.7%), Oct (7.3%), Nov (7.6%), and Dec (7.7%)
Divisions (Div.)	It implies the location in the plant where the incident occurred. A total of fourteen categories of this attribute, i.e., Div1, Div2, Div3, ... Div13 and Div14 are available in the dataset	Div1 (0.3%), Div2 (10.6%), Div3 (2.4%), Div4 (6.6%), Div5 (0.01%), Div6 (15.3%), Div7 (8.5%), Div8 (2.1%), Div9 (3.7%), Div10 (1.7%), Div11 (2.9%), Div12 (28%), Div13 (0.2%), and Div14 (17.7%).
Incident outcome (IO)	This attribute refers to the category of the incident occurred. The attribute IO has three classes: (i) Injury (Incident 1), (ii) near miss (Incident 2), and (iii) property damage (Incident 3)	Injury (34.9%), near miss (40.3%), and property damage (24.8%)
Incident events (PC)	This attribute refers to the event of an incident which occurred. It may be crane dashing, dashing/ collision, derailment, slip trip fall (STF), and so forth. In the dataset, this attribute has eleven classes, namely Cause 1, Cause 2, ... Cause 10, and Cause 11	Cause 1 (6.3%), Cause 2 (4.7%), Cause 3 (1.7%), Cause 4 (12.1%), Cause 5 (14.9%), Cause 6 (4.1%), Cause 7 (18.6%), Cause 8 (4.7%), Cause 9 (11.2%), Cause 10 (20.7%), and Cause 11 (0.9%)
Condition of the work (WC)	The attribute refers to the status of the work while the incident took place. There are three classes of this attribute, i.e., working in a group (GW), working a single (SW), and not applicable (NA)	Single working (SW) (36.7%), Group working (GW) (48.7%), and not applicable (14.6%)
Condition of machine (MC)	This attribute refers to the condition of the machine while the incident happened. It describes the machine whether it was idle (MI) or working (MW), or not applicable (NA)	Working (W) (43.8%), idle (I) (13.6%), and not related (N) (42.6%)
Types of observation (OT)	It indicates the basic or root causes of incidents. It is categorized into four classes: (i) unsafe act (UA), which means a human, due to his/her own fault, is accountable for the occurrence of incident; (ii) unsafe condition as well as unsafe act (UAC), which denotes the both factors, hazardous condition, and human fault are responsible, (iii) unsafe act by other (UAO), which implies that incident happens because of someone's mistakes, and (iv) unsafe condition (UC), which indicates a state with potential leading to the occurrence of an incident.	Unsafe act (UA) (46.6%), Unsafe condition (UC) (32.7%), Unsafe act and condition (UAC) (12.2%), and Unsafe act by others (8.4%)
Employee Type (ET)	It indicates the type of worker/employee. It may be either an employee or a contractor	Permanent employee (E) (28.3%), and Contractor (C) (71.7%)
Incident type (IT)	It denotes the types of the incidents; either human behavior (Bhv.) or process type (Pro.)	Behavioral (Bhv) (75.2%), and Process-related (Pro) (24.8%)
Standard operating procedure (SOP)	This attribute implicates a guideline, which should be maintained or followed during work. There are a total of six classes in this attribute, namely SOP is available and followed (SAF), SOP is available but not followed (SANF), SOP is inadequate but is followed (SIF), SOP is inadequate and is not followed (SINF), SOP is not available and is not required (SNNR), and SOP is not available but is required (SNR)	SAF (19.5%), SINF (7.6%), SANF (32.4%), SIF (10.4%), SNN (19.6%), and SNR (10.6%)
Brief description of the incident (BD)	It is an unstructured text data, which narrates how and why the incident took place	Text data
Event leading to the incident (EL)	This attribute contains a detailed description of circumstances and conditions that led to the happening of the incident	Text data

done, for example, removal of inconsistencies manually, and missing data by random forest algorithm. Then, unstructured text attributes have been converted into a categorical attribute using topic modeling. Thereafter, class imbalance problem is handled using SMOTE algorithm.

4 Results and discussions

In this section, the generation of a new categorical attribute using topic modeling, evaluation of the importance of attributes using chi-square approach, hyper-parameter study, and prediction of incident outcomes are discussed in details.

4.1 Generation of a categorical attribute using topic modeling

There are two text attributes, called ‘BD’ and ‘EL’ within the dataset, which comprise the description of incidents. LDA topic modeling has been used to create a categorical attribute from the texts. Four metrics have been used simultaneously to determine the number of topics optimally from the attributes of the unstructured text. From the topic modeling, a total of nine topics are extracted (refer to Fig. 5). With each of the topics, an exhaustive list of terms, based on the probability of occurrence, is generated. Using the list, a meaningful event is obtained for each topic. For the purpose of visualization, the top eight terms per topic with its corresponding probability of occurrence are shown in Table 2. For instance, in Topic1, the top eight terms, ‘road,’ ‘shift,’ ‘near,’ ‘injury,’ ‘come,’ ‘sudden,’ ‘duty,’ and ‘fell’ are found. From the list of the terms extracted, it can be inferred that Topic1 describes ‘Falling’ as a meaningful event, which has been later validated by five domain experts.

4.2 Evaluation of feature importance using Chi-square approach

Once the dataset has been pre-processed, chi-square test is conducted for the evaluation of the importance of attributes. The higher values of chi-square in Fig. 6 suggest that the attributes, such as ‘Employee types,’ ‘Topic,’ ‘Incident events,’ and ‘Machine condition’ are the significant predictors for the prediction of incident outcomes.

4.3 Hyper-parametric study

A tree-based regression model is used to find the optimal values of the hyper-parameters of DNN for producing the best accuracy. The hyper-parameters of a DNN include learning rates, activation function, the number of hidden layers, and the number of neurons in each hidden layer. First, the model is evaluated based on the values of the parameters, which are initially set at random. The model is then improved by sequentially evaluating the cost function for a number of evaluations, which is set equal to 10 (i.e., ‘n_calls=10’). This is performed for all the three optimized DNNs. For ADNN, the best results are obtained with a single layer of the six hidden layers with 5, 7, 7, 6, 4, and 5 neurons, respectively, a learning rate of 0.061, and ‘rectified linear unit (ReLU)’ activation function. For RMSProp-DNN, the best results are achieved with a setting of five hidden layers with 3, 3, 5, 4, and 4 neurons, respectively, the learning rate of 0.0041, and ‘ReLU’ activation function. Similarly, for SGD-DNN, the best results are obtained with the five hidden layers with 6, 4, 3, 4, and 3 neurons, respectively, the learning rate of 0.00001, and ‘tanh’ activation function. The ranges and the optimal values of the optimized classifiers are listed in Table 3. The convergence plots of all the three cases are also recorded and depicted in Fig. 7a–c.

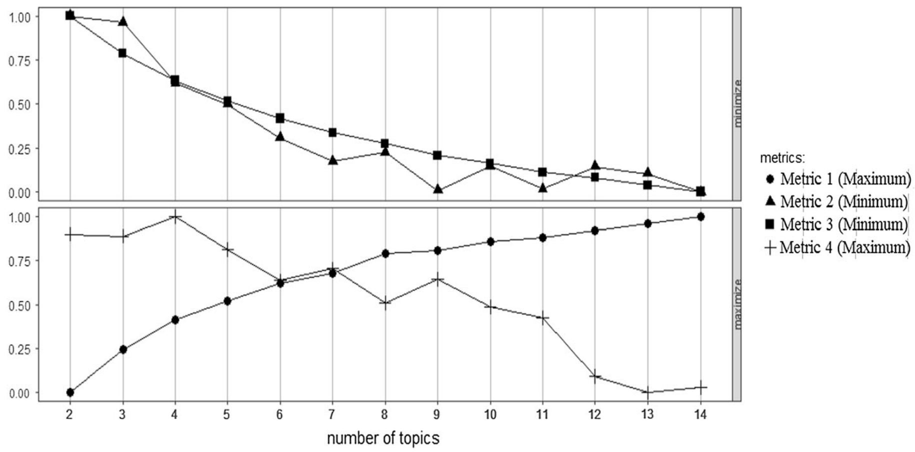


Fig. 5 Optimal number of topics from incident narratives

Table 2 Eight top terms with probabilities for each of the nine topics

Topic no.	Top eight terms	Meaningful event
1	0.0419*road + 0.039189*shift + 0.035124*near + 0.033509*injury + 0.031548* come + 0.028174*sudden + 0.026819*duty + 0.025522*fell	Injuries due to fall
2	0.042011*one + 0.041117*work + 0.03016+8*job + 0.026867*person + 0.025942*area + 0.0211*material + 0.014962*plate + 0.013081*kept	Material Handling
3	0.075108*got + 0.03764*load + 0.028079*line + 0.027372*place + 0.027069*wagon + 0.025925*loco + 0.020101*due + 0.019865*point	Locomotive failures
4	0.053615*side + 0.027446*dumper + 0.025663*car + 0.025211*driver + 0.023513*gate + 0.020741*vehicle + 0.017911*dash + 0.015393*truck	Vehicle hitting/collision
5	0.051178*hand + 0.036216*left + 0.036216*right + 0.031686*cut + 0.025129*got + 0.024831*hit + 0.023966*first + 0.020867*finger	Finger related injuries
6	0.038966*slip + 0.03491*fall + 0.031652*leg + 0.028139*fell + 0.022263*floor + 0.021784*machine + 0.02156*one + 0.019963*person	Slipping
7	0.030726*water + 0.025716*due + 0.024613*pipe + 0.021178*open + 0.018783*gas + 0.014876*slag + 0.014813*door + 0.01434*start	Pipe Leakage
8	0.069761*operation + 0.060601*crane + 0.033244*coil + 0.026268*lift + 0.018292*roll + 0.015865*position + 0.013166*broken + 0.0124378*damage	Crane operation failure
9	0.48791*fire + 0.026413*cable + 0.022651*found + 0.015609*damage + 0.1552*electric + 0.015012*control + 0.014237*power + 0.012924*room	Control failures

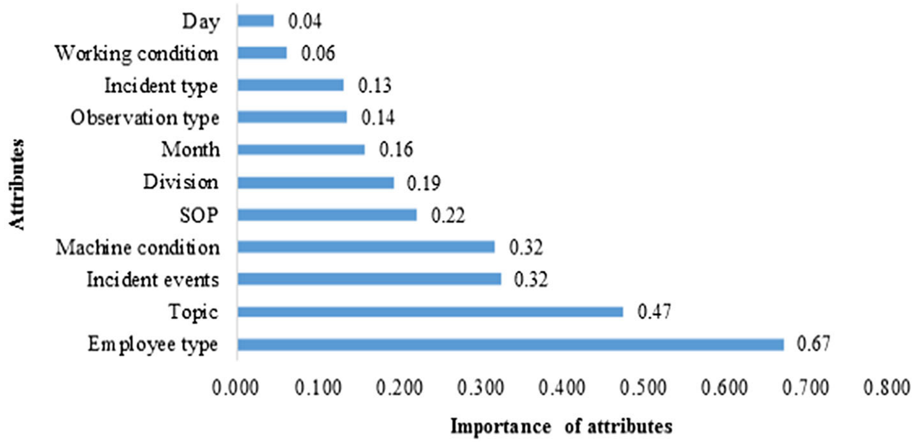


Fig. 6 Importance of attributes in prediction

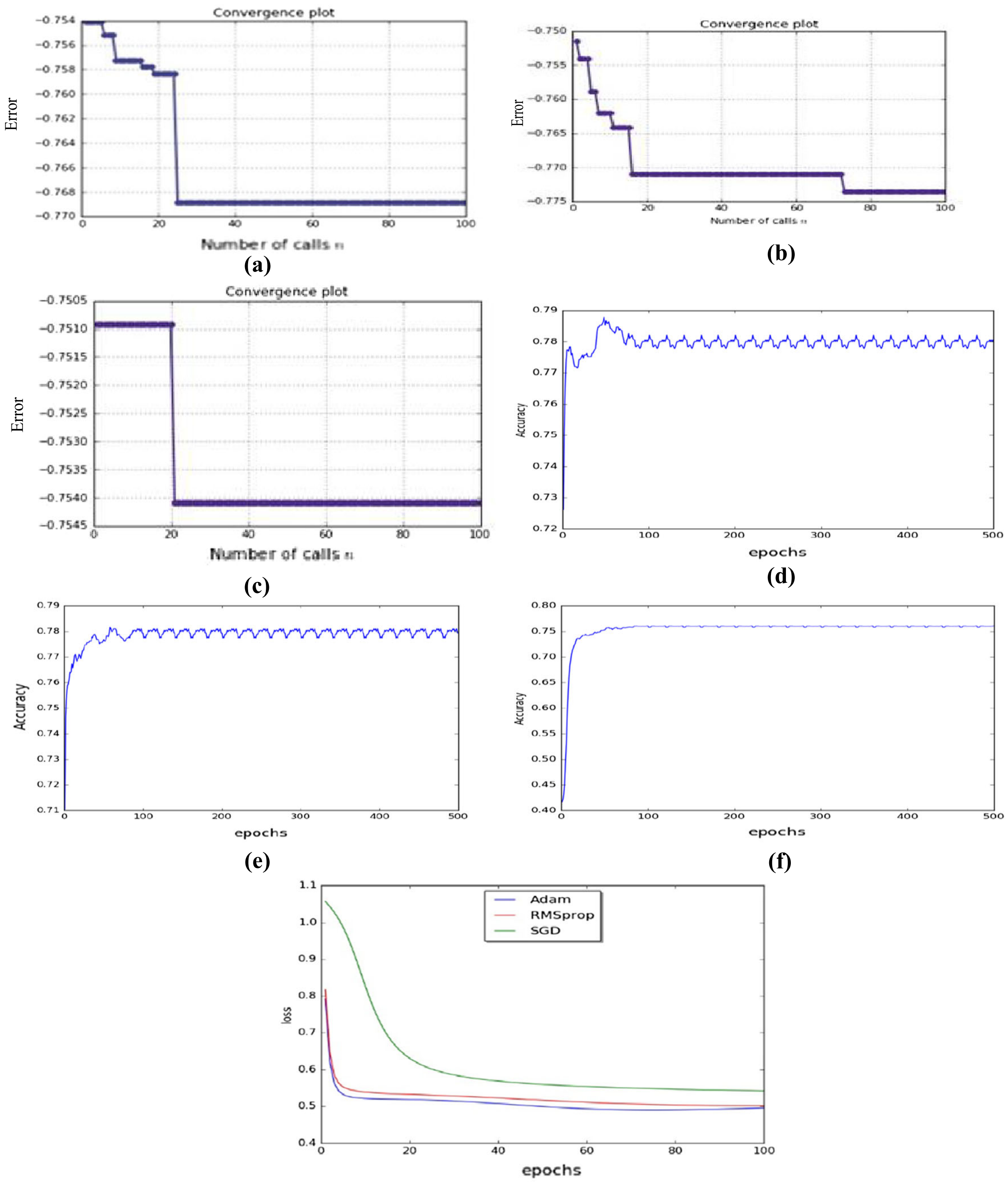
Table 3 Hyper-parameters of the optimized classifiers

Hyper-parameters	Range	Classifiers		
		ADNN	RMSProp-DNN	SGD-DNN
Learning rate	10^{-6}–10^{-2}	0.061	0.0041	0.00001
Activation functions	'ReLU,' 'tanh,' and 'sigmoid'	'ReLU'	'ReLU'	'tanh'
Number of layers	0–10	6	5	5
Number of neurons	5–25	20	17	12

4.4 Prediction

This section demonstrates the classification performances of the three optimized DNNs, namely ADNN, RMSProp-DNN, and SGD-DNN. Evaluation of the performances is done based on incident data and five other benchmark datasets, namely 'Breast cancer,' 'Iris,' 'PID,' 'Hungarian,' and 'Cleveland' retrieved from UCI Machine Learning Repository[1]. Besides these, other three state-of-the-art classifiers, namely k-NN, SVM, and RF are also employed

[1] https://archive.ics.uci.edu/ml/index.php.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig. 7 Convergence plots: **a** Error plot of ADNN, **b** error plot of RMSProp-DNN, **c** error plot of SGD-DNN, **d** accuracy plot of ADNN, **e** accuracy plot of RMSProp-DNN, **f** accuracy plot of SGD-DNN, and **g** loss vs epochs plot of ADAM, RMSProp, and SGD-based DNN classifiers

using 10-fold cross-validation to perform further checking of the performance. From Table 4, it can be seen that the maximum accuracy 78.8% is generated by ADNN, whereas RMSProp-DNN and SGD-DNN produce the accuracies 78.1%, and 76.1%, respectively. Other algorithms, RF, k-NN, and SVM produce the best accuracy of 73.28%, 70.76%, and 65.11%, respectively. With the experiments, the corresponding graphs, i.e., accuracy versus epochs are also depicted for the three optimized algorithms in Fig. 7d–f. In addition, for comparative study, the analysis related to 'loss versus epochs' is plotted for the three models (refer to Fig. 7g). The trend of accuracies obtained by the six

Table 4 Best accuracy of different classifiers on different datasets using 10-fold cross-validation

Classifiers	Datasets					
	Incident data	Breast cancer	Iris	PID	Cleveland	Hungarian
ADNN	0.788	0.938	0.967	0.715	0.826	0.814
RMSProp-DNN	0.781	0.9298	0.9666	0.704	0.817	0.792
SGD-DNN	0.761	0.64	0.933	0.684	0.795	0.783
SVM	0.6511	0.6228	0.9333	0.657	0.768	0.754
k-NN	0.7076	0.8859	0.9333	0.662	0.771	0.775
RF	0.7328	0.891	0.944	0.69	0.803	0.784

models, namely ADNN, RMSProp-DNN, SGD-DNN, SVM, k-NN, and RF is somewhat similar in nature in terms of the order of best accuracies on five benchmark test datasets. Hence, from the experimental results reported in Table 4, it is explored that ADNN classifier produces the highest accuracy for all the datasets. Although k-NN, RF, SGD-DNN, RMSProp-DNN, and SVM are compared with the proposed ADNN, DNN parameters are tuned using SGD, RMSProp, and ADAM optimizers to determine the best one since these optimizers are best suited for tuning deep learning model parameters rather than others. Therefore, convergence plots are exhibited for these three optimizers only.

It is to be noted that in all datasets, information related to attributes and their respective classes are given. Using this information, important attributes are extracted and used in analyses for better classification performance. The extraction of attributes is done by using a stacked auto encoder. For example, the stacked auto encoder is applied over the data, ‘Iris,’ which has four attributes, namely, ‘sepal length,’ ‘sepal width,’ ‘petal length,’ and ‘petal width,’ and three decision or response classes, namely ‘setosa,’ ‘virginica,’ and ‘versicolour.’ Now, the dataset is divided into two sets: training set and test set. The number of inputs used in an auto encoder is same as the number of attributes in the data. Hyper-parameters are multiplied with attribute values for each input sample to extract its (e.g., input sample) feature value. This feature extraction is done in hidden layer of auto encoder. For all the samples in input data, a feature map is generated. The feature map constitutes all the feature values generated from the input data. In stacked auto encoder, the generated feature map is fed to the next auto encoder to get its reduced feature map. In this way the feature map is passed through all the auto encoders, and finally generate more reduced feature map. Input data can be represented concisely using its reduced feature map. Instead of taking the entire input data, its corresponding reduced feature map is used for softmax classification; thereby increasing the classification accuracy as well as reducing the computational time. Therefore, useful feature map extraction and its softmax classification is more advantageous and necessary than just only using softmax classification on the entire input data.

4.4.1 Performance evaluation and comparison

Besides the accuracy, other performance measures including sensitivity (i.e., recall), F–measure, and precision are also evaluated to compare the classification models. The results of ADNN, RMSProp-DNN, SGD-DNN, SVM, k-NN, and RF are summarized in Table 5.

4.4.2 Statistical test for significance

Following the strategy adopted by [78–80], two nonparametric tests, Wilcoxon signed-ranks test and Mann–Whitney U test are carried out with 95% confidence interval (i.e., significance level, $\alpha = 0.05$) for comparison of the performance of ADNN with each of the other two models, i.e., RMSProp-DNN and SGD-DNN. Results reveal that there exist significant differences between ADNN and the other two models since $p < 0.05$ (refer to Table 6). In addition, the results of the Mann–Whitney U test also support the findings of the previous test (refer to Table 7).

4.4.3 Robustness checking of the classifiers

Robustness checking of the three optimized classifiers is carried out using five independent runs with 10-fold cross-validation for every run. Adopting the process of Oztekin et al. [81], seeds are randomly selected for splitting the dataset into training and testing. Five different numbers, i.e., 221, 223, 225, 227, and 229 are assigned to seeds for five different runs, which, in turn, produces a set of 50 cross-validation accuracies (i.e., 10-fold cross-validation accuracies per seed × 5 seeds). Based on these values, a box plot is generated for each of the six classifiers (i.e., ADAM-DNN, RMSProp_DNN, SGD-DNN, SVM, k-NN, and RF) (refer to Fig. 8). From the figure, it is unveiled that the ADNN algorithm shows the maximum accuracy values with the least range of dispersion; whereas the minimum

Table 5 Performance metrics of the six classifiers

Performance metrices	Classes	ADNN			RMSProp-DNN			SGD-DNN			k-NN			SVM			RF		
		Min.	Max.	Mean	Min.	Max.	Mean	Min.	Max.	Mean	Min.	Max.	Mean	Min.	Max.	Mean	Min.	Max.	Mean
F-measure (F1-score)	Class 1	0.702	0.914	**0.901**	0.535	0.885	0.873	0.752	0.861	0.791	0.82	0.91	0.88	0.86	0.93	**0.91**	0.814	0.85	0.832
	Class 2	0.182	0.754	**0.714**	0.619	0.754	0.738	0.624	0.707	0.649	0.71	0.8	**0.76**	0.69	0.83	0.77	0.725	0.784	0.759
	Class 3	0.647	0.689	**0.66**	0.054	0.657	0.632	0.528	0.697	0.567	0.6	0.88	0.81	0.59	0.89	**0.82**	0.62	0.782	0.71
Recall	Class 1	0.588	0.897	**0.884**	0.373	0.884	0.866	0.684	0.82	0.737	0.78	0.93	0.88	0.81	0.96	**0.91**	0.8	0.92	0.875
	Class 2	0.1	0.708	0.641	0.512	0.748	**0.722**	0.524	0.658	0.588	0.71	0.8	0.76	0.63	0.88	0.78	0.73	0.827	**0.786**
	Class 3	0.681	0.757	**0.718**	0.028	0.624	0.583	0.538	0.76	0.577	0.59	0.88	0.81	0.5	0.93	**0.83**	0.621	0.882	0.76
Precision	Class 1	0.902	0.937	**0.92**	0.869	0.946	0.882	0.817	1.00	0.883	0.87	0.88	0.88	0.9	0.92	**0.91**	0.86	0.89	0.885
	Class 2	0.794	0.963	**0.816**	0.742	0.783	0.756	0.657	1.00	0.756	0.7	0.8	0.76	0.78	0.78	**0.78**	0.71	0.78	0.758
	Class 3	0.584	0.646	0.61	0.676	0.813	**0.699**	0.521	0.842	0.608	0.61	0.87	0.81	0.71	0.86	**0.82**	0.62	0.852	0.819

accuracies are yielded by the SVM algorithm. Maximum dispersion of accuracies is observed for k-NN algorithm. Therefore, from the comparative study, the ADNN classifier can be deemed as the robust model.

From the experimental analyses, it is to be noted that one or more hyper-parameters are set to a particular value in ML-approach, which influences the testing accuracy of the classification algorithms. With the proper selection of the hyper-parameter values, the ML algorithm performs with the optimum accuracy. Therefore, parameter tuning is a very important task in ML approach. This tuning process comprises three steps: *Step 1:* Parameters are randomly initialized with some weight values; *Step 2:* error/loss is calculated based on the weight values; and *Step 3:* the error is propagated back to update the weight values such that the error should be minimized. In such cases, backpropagation of the errors can be made using several optimization algorithms, such as gradient descent (GD) and stochastic gradient descent (SGD) algorithms. In GD-based optimization algorithm, first, the error is calculated for the entire dataset and then the error is back propagated for weight updation. Therefore, it takes a lot of time to move even a single step closer to the optimum weight value/cost. This problem is solved in SGD, where the entire dataset is divided into several mini batches of size one. After passing through one mini batch, the parameters are updated; thereby speed up the system. These two optimization algorithms are further speeded up by incorporating momentum that defines the desired direction of the learning process so that parameters can reach to optimality with comparatively shorter time. As stated before, SGD is better than GD and hence, SGD with momentum is superior to GD with momentum. RMSProp is a GD-based optimization algorithm. It combines GD with momentum. Therefore, it takes less time than GD to achieve optimality. Whereas, in SGD, the hyper-parameters are updated after passing through one mini batch, having the size equal to one. ADAM combines the SGD with momentum. SGD reduces the searching space and momentum increases the learning rate. Therefore, ADAM is superior to GD, SGD, and RMSprop algorithms in terms of computation time.

5 Conclusions

The present study proposes a new methodology for the development of a prediction model which enables us to predict the incident outcomes using the hidden information underlying the data. DNN, an effective and powerful classifier, has been used in this task. The findings of the study allow us to draw some useful insights regarding the handling of unstructured text data and parameter optimization of the classifiers. For instances, topic modeling is

Table 6 Results of the Wilcoxon Signed Rank test for ADNN, RMSProp-DNN, and SGD-DNN for the incident dataset

	Z-value	Standard	*p*-value deviation	Mean difference	Sig. ($p < 0.05$)
RMSProp-DNN and ADNN	−7.2965	286.51	0	0.02	ADAM
SGD-DNN and ADNN	−8.6818	290.84	0	−0.02	ADAM

Table 7 Results of the Mann–Whitney U test for ADAM-DNN (ADNN), RMSProp-DNN, and SGD-DNN for each dataset

Algorithms	Datasets	Incident dataset	Breast cancer	Iris	PID	Cleveland	Hungarian
SGD and ADAM-DNN	Z-Value	−11.876	−11.971	11.926	−12.841	−10.325	−11.301
	P-Value	0	0	0	0.00002	0	0
	Mean Diff.	100.5	100.5	101	102	105	103
	U-Value	139	100	132	125	134	162
	Significance ($p < 0.05$)	ADAM-DNN	ADAM-DNN	ADAM-DNN	ADAM-DNN	ADAM-DNN	ADAM-DNN
RMSProp and ADAM-DNN	Z-Value	−4.421	−3.45	−10.8	−6.547	−8.258	−11.302
	P-Value	0	0	0	0	0	0
	Mean Diff.	100.5	100.5	125.5	113	142	118
	U-Value	3190	3587.5	1450	2145	2467	1548
	Significance ($p < 0.05$)	ADAM-DNN	ADAM-DNN	ADAM-DNN	ADAM-DNN	ADAM-DNN	ADAM-DNN

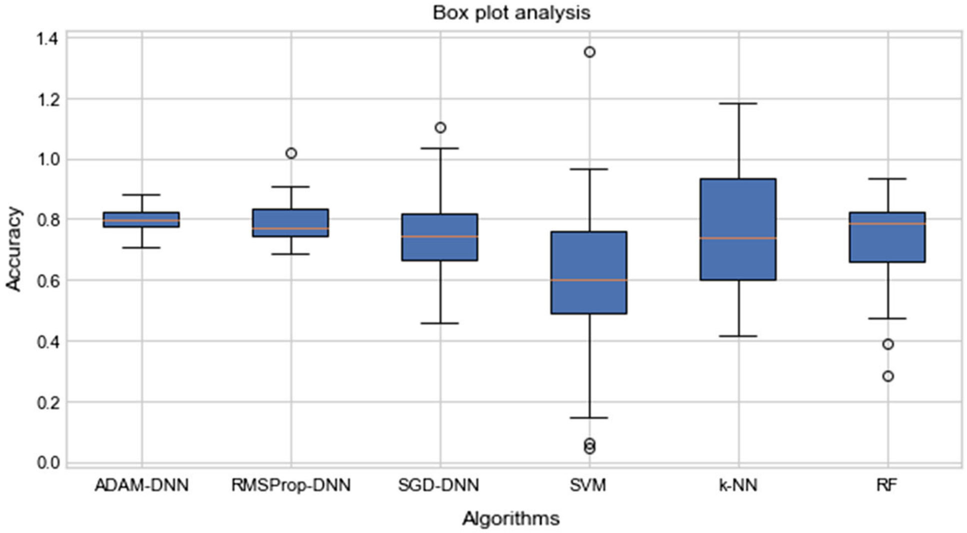


Fig. 8 Box plot analysis for robustness checking of different classifiers

found to be a very effective tool used for the analysis of unstructured texts. Moreover, using chi-square testing, the ‘topic’ is also found to be one of the important predictors of incident outcomes. Besides this, other attributes including ‘Employee types,’ ‘Incident events,’ and ‘Machine condition’ are found to be the important determinants of the response attribute. Key words related to each topic are added with other categorical predictors to form the input feature space. This input feature space is fed to the DNN for prediction. In order to achieve the improved accuracy in classification, the parameters of DNN are tuned by the three optimization algorithms, namely RMSProp, ADAM, and SGD, separately. From this study, it is evident that the proposed approach ADNN is found to be the best classifier with the highest accuracy. In support of the findings from the experiments, other algorithms, namely SGD-DNN, RMSProp-DNN, SVM, k-NN, and RF have also been applied to the incident dataset. The results reveal that the ADNN classifier outperforms others in all cases. Further, all the algorithms used in this study have been tested using five different available benchmark datasets. In all these cases, ADNN algorithm performs the best. In order to check whether the performance of ADNN algorithm differs significantly or not from other algorithms (i.e., SGD-DNN, and RMSProp-DNN), two statistical tests, namely Wilcoxon signed ranked test and Mann–Whitney U test have been carried out. Results reveal that our proposed algorithm significantly performs better than the others. Finally, using boxplot analysis, ADNN is found to be the most robust classifier. Therefore, the present study is expected to have potential to contribute both in theoretical and practical aspects.

5.1 Theoretical contributions

From the theoretical point of view, the study offers a number of contributions. First, the proposed methodology shows a new way to handle issue of the use of unstructured texts in analysis using LDA-based topic modeling. Second, the methodology explores a strategy of using parameter optimization of classifiers for increasing prediction performances. Third, the higher predictive accuracy of the optimized classifiers reveals that incidents do not occur in a chaotic fashion, but hidden patterns do exist. Therefore, these patterns can be explored and captured with the use of machine learning techniques. This finding suggests that occupational safety should be studied empirically in a systematic way rather than strictly following a qualitative approach through subjective, expert-opinion-based data analysis.

5.2 Practical implications

From the practical point of view, the study has some real implications. The study can help decision-makers like safety professionals to predict the possible outcomes of incidents. It may be either injury or near-miss, or property damage. Based on this predicted outcome, safety-related decisions can be undertaken, such as working places should be cleaned and free of spillage of oil or any liquid, proper illumination level at working places should be maintained, unexposed cables to be removed from working places, and others. In addition, it can help decision-makers pre-process data by addressing the issues of handling of unstructured text in analysis. The use of LDA helps in automatic text classification which enables safety managers to identify useful information (such as probable accidents with severities) by extracting and relating relevant data present in documents. It is useful when it is used on proactive data (i.e., information that lead to incident). The proactive data indicates the data collected prior to the occurrence of any incident, for example, an inspection report. This report usually narrates the date, time, location, machine condition, working nature of a worker, type of activity being performed by workers, pre-incident working conditions, etc. Using both incident and inspection data, one attribute ‘Incident’ can be generated which has two classes, ‘Yes,’ or ‘No.’ The ‘Yes’ means the incident has occurred; on the other hand, ‘No’ means the incident has not occurred. Using automatic text classification on this information, a safety manager can at least classify the documents as either ‘Yes’ or ‘No.’ If any new document (from inspection) is classified as ‘Yes’ by the classification algorithm, it means that there is a possibility of the occurrence of an incident. With the help of this information, the safety manager can take proactive measures to prevent this occurrence. Moreover, the evaluation of the importance of attributes toward incident outcome prediction can help the decision-makers identify the important and unimportant attributes. Therefore, they can put more focus on the important attributes or factors responsible for incidents and accordingly, the factors can be improved or eliminated from the system to prevent the occurrences of incidents.

However, like other studies, this study has also some limitations. The study suffers from the issue that demands an extensive human labor, which is necessary to sanitize the data prior to analysis. This is a time-consuming and less effective process. Further, the dataset consists of a limited number of incident records. It is noteworthy to mention that using a substantial amount of data in the analysis is necessary for achieving the model’s generality. Based on the study carried out, some interesting avenues could be explored for the future research. For examples, the study could be expanded to develop an ADNN-based automated decision support system (ADSS) [82] which not only enables us for prediction but also facilitates smart decision-making based on the generation of rules. Therefore, the present study can be useful for academics and researchers through the development of a new methodology to overcome the issues of unstructured and hidden information in data. Moreover, to resolve the issues, the study could be expanded beyond the manufacturing industry, such as construction, process industry, aviation, and so forth.

Appendix

A notations used

See Table 8.

Table 8 Notations used in the study

Notations	Meaning
N_i	Random variable representing the number of words in i-th document, where $i = 1, 2, ..., n$
θ_i	Random variable denoting per document-topic proportion
α	Proportion parameter, where $\alpha < 1$
β_{T_j}	Per-topic word proportion parameter
T_j	j-th topic, where $j = 1, 2, ..., m$
η	Random variable related to Dirichlet distribution
z_{i,w_l}	The topic of each word $w_l \in \{w\}$ in the i-th document, where $l = 1, 2, ..., p$
D_i	i-th document
w_{i,w_l}	Observed word in a topic
β	Per-topic word proportion parameter
S	A universe, where $\left\{x_1, x_2, ..., x_Q\right\}$
Q	The total number of instances in U
A	A set of predictor attributes, where $\left\{a_1, a_2, ..., a_p\right\}$
p	The number of predictors in U
a_s	An arbitrary attribute, where $s \in p$
$i_{mis}^{(s)}$	The entries in a dataset having missing values, where $i_{mis}^{(s)} \leq Q$
$y_{obs}^{(s)}$	Observed values of attribute a_s
$y_{mis}^{(s)}$	Missing values of attribute a_s
$x_{obs}^{(s)}$	The attributes other than a_s with observations $i_{obs}^{(s)} = \{1, 2, ..., Q\} \setminus i_{mis}^{(s)}$
$x_{mis}^{(s)}$	The attributes other than a_s with observations $i_{mis}^{(s)}$
X	Information table used in SMOTE
P	Number of oversampling used in SMOTE
w_i	Random sample used in SMOTE
y_i	Synthetic samples generated by SMOTE
δ	A random value (0, 1) used in SMOTE
NS	Total number of samples/instances used in SMOTE
q	Total number of minor samples used in SMOTE
M	The number of inputs or outputs used in an auto-encoder
N	The number of hidden nodes used in the auto-encoder
$c = [c_1\ c_2\ c_3\ ...\ c_N]^T$	A set of outputs from the auto-encoder
$x = [x_1\ x_2\ x_3\ ...\ x_M]^T$	A set of inputs used in the auto-encoder
f	The activation function (usually, sigmoidal) for the auto-encoder
$b = [b_1\ b_2\ b_3\ ...\ b_N]^T$	Biases associated with the auto-encoder
$W = [w_1\ w_2\ w_3\ ...\ w_N]^T$	The weights associated with the auto-encoder
c	The relationship between input and output in the encoder
$g_{AE}^1 = g_E \circ g_D$	The input–output relationship of an auto-encoder, where g_E and g_D denote
L	The number of auto-encoders cascaded to form a stacked auto-encoder
ϵ	Learning rate
θ	Initial parameter
m	Number of samples in a minibatch
$\left\{X^{(1)}, X^{(2)}, X^{(3)}, ..., X^{(m)}\right\}$	Training set
$y^{(i)}$	Target
$\widehat{g}$	Estimate of gradient
∇_θ	Gradient
ρ	Decay rate
δ	Constant of smaller value

Table 8 (continued)

Notations	Meaning
r	Accumulation parameter
$\Delta\theta$	Difference in parameter θ
α	Step size
β_1, β_2	Exponential decay rate for the moment estimates, where $\beta_1, \beta_2 \in [0, 1)$
$f(\theta)$	Stochastic objective function with parameter θ
θ_0	Initial parameter vector
m_0	First moment vector
v_0	Second moment vector
t	Time step
g_t	Gradient at time step t
m_t	First-moment biased estimate at time step t
v_t	Second-moment biased estimate at time step t
$\widehat{m_t}$	Bias-corrected first-moment estimate
$\widehat{v_t}$	Bias-corrected second-moment estimate

Acknowledgements The paper is original in its contents and is not under consideration for publication in any other journals/proceedings. We would like to express our special appreciation and thanks to Ms. Anima Pramanik, Research Scholar of the Department of Industrial & Systems Engineering, IIT Kharagpur, for her assistance in this study. We deeply acknowledge “Safety Analytics & Virtual Reality (SAVR) Laboratory” (www.savr.iitkgp.ac.in), Department of Industrial & Systems Engineering, IIT Kharagpur, for experimental/computational and research facilities for this work. We would also like to thank the management of the plant for providing relevant data and their support and cooperation during the study.

Funding There is no funding received by this work. The authors have no financial or proprietary interests in any material discussed in this article.

Declarations

Conflict of interest There is no potential conflict of interest to disclose, such as employment, financial or non-financial interest.

References

1. Sánchez AS, Fernández PR, Lasheras FS, de Cos Juez FJ, Nieto PG (2011) Prediction of work-related accidents according to working conditions using support vector machines. Appl Math Comput 218(7):3539–3552
2. ILO, Promoting safe and healthy jobs : the ILO global programme on safety, health and the environment (Safework), Tech. rep., In: World of Work (2008)
3. EUROSTAT (2009) Labour force survey 2007 ad hoc module on accidents at work and work-related health problems, Tech Rep, In: European communities
4. Chi N-W, Lin K-Y, El-Gohary N, Hsieh S-H (2016) Evaluating the strength of text classification categories for supporting construction field inspection. Autom Constr 64:78–88
5. Chen WT, Chang P-Y, Chou K, Mortis LE (2010) Developing a cbr-based adjudication system for fatal construction industry occupational accidents. Part i: building the system framework. Exp Syst Appl 37(7):4867–4880
6. Fragiadakis N, Tsoukalas V, Papazoglou V (2014) An adaptive neuro-fuzzy inference system (anfis) model for assessing occupational risk in the shipbuilding industry. Saf Sci 63:226–235
7. Goh YM, Chua D (2013) Neural network analysis of construction safety management systems: a case study in singapore. Constr Manag Econ 31(5):460–470
8. Khakzad N, Khan F, Amyotte P (2011) Safety analysis in process facilities: comparison of fault tree and bayesian network approaches. Reliab Eng Syst Saf 96(8):925–932
9. Sorock GS, Ranney TA, Lehto MR (1996) Motor vehicle crashes in roadway construction workzones: an analysis using narrative text from insurance claims. Accid Anal Prevent 28(1):131–138
10. Lehto MR, Sorock GS (1996) Machine learning of motor vehicle accident categories from narrative data. Methods Inf Med 35(04/05):309–316
11. Wellman HM, Lehto MR, Sorock GS, Smith GS (2004) Computerized coding of injury narrative data from the national health interview survey. Accid Anal Prevent 36(2):165–171
12. Noorinaeini A, Lehto MR (2006) Hybrid singular value decomposition; a model of human text classification. Int J Human Factors Model Simul 1(1):95–118
13. Pons-Porrata A, Berlanga-Llavori R, Ruiz-Shulcloper J (2007) Topic discovery based on text mining techniques. Inf Process Manag 43(3):752–768
14. Brooks B (2008) Shifting the focus of strategic occupational injury prevention: mining free-text, workers compensation claims data. Saf Sci 46(1):1–21

15. Fan H, Li H (2013) Retrieving similar cases for alternative dispute resolution in construction accidents using text mining techniques. Autom Constr 34:85–91
16. Abdat F, Leclercq S, Cuny X, Tissot C (2014) Extracting recurrent scenarios from narrative texts using a bayesian network: application to serious occupational accidents with movement disturbance. Accid Anal Prevent 70:155–166
17. Sanchez-Pi N, Martí L, Garcia ACB (2014) Text classification techniques in oil industry applications. In: international joint conference SOCO'13-CISIS'13-ICEUTE'13, Springer, 2014, pp. 211–220
18. Sanchez-Pi N, Martí L, Garcia ACB (2016) Improving ontology-based text classification: an occupational health and security application. J Appl Logic 17:48–58
19. Goh YM, Ubeynarayana C (2017) Construction accident narrative classification: an evaluation of text mining techniques. Accid Anal Prevent 108:122–130
20. Zhang Z, He Q, Gao J, Ni M (2018) A deep learning approach for detecting traffic accidents from social media data. Transp Res Part C Emerg Technol 86:580–596
21. Song B, Suh Y (2019) Identifying convergence fields and technologies for industrial safety: Lda-based network analysis. Technol Forecast Soc Change 138:115–126
22. Suh Y (2021) Sectoral patterns of accident process for occupational safety using narrative texts of osha database. Saf Sci 142:105363
23. Zhong B, Pan X, Love PE, Ding L, Fang W (2020) Deep learning and network analysis: classifying and visualizing accident narratives in construction. Autom Const 113:103089
24. Sarkar S, Vinay S, Pateshwari V, Maiti J (2016) Study of optimized svm for incident prediction of a steel plant in India. In: 2016 IEEE annual India conference (INDICON), IEEE, 2016, pp. 1–6
25. Sarkar S, Pramanik A, Maiti J, Reniers G (2020) Predicting and analyzing injury severity: a machine learning-based approach using class-imbalanced proactive and reactive data. Saf Sci 125:104616
26. Brown DE (2016) Text mining the contributors to rail accidents. IEEE Trans Intell Transp Syst 17(2):346–355
27. Nenonen N (2013) Analysing factors related to slipping, stumbling, and falling accidents at work: application of data mining methods to finnish occupational accidents and diseases statistics database. Appl Ergon 44(2):215–224
28. Bevilacqua M, Ciarapica F, Giacchetta G (2008) Industrial and occupational ergonomics in the petrochemical process industry: a regression trees approach. Accid Anal Prevent 40(4):1468–1479
29. Cheng C-W, Yao H-Q, Wu T-C (2013) Applying data mining techniques to analyze the causes of major occupational accidents in the petrochemical industry. J Loss Prevent Process Ind 26(6):1269–1278
30. Rungskunroch P, Jack A, Kaewunruen S (2021) Benchmarking on railway safety performance using bayesian inference, decision tree and petri-net techniques based on long-term accidental data sets. Reliab Eng Syst Saf 213:107684
31. 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. Reliab Eng Syst Saf 200:106931
32. Ghasemzadeh A, Hammit BE, Ahmed MM, Young RK (2018) Parametric ordinal logistic regression and non-parametric decision tree approaches for assessing the impact of weather conditions on driver speed selection using naturalistic driving data. Transport Res Record 2672(12):137–147
33. Babič F, Lukáčová A, Paralič J (2015) Descriptive and predictive analyses of data representing aviation accidents. New research in multimedia and internet systems. Springer, Cham, pp 181–190
34. Rivas T, Paz M, Martín J, Matías JM, García J, Taboada J (2011) Explaining and predicting workplace accidents using data-mining techniques. Reliab Eng Syst Saf 96(7):739–747
35. Matías J, Rivas T, Martín J, Taboada J (2008) A machine learning methodology for the analysis of workplace accidents. Int J Comput Math 85(3–4):559–578
36. He X, Chen W, Nie B, Zhang M (2010) Classification technique for danger classes of coal and gas outburst in deep coal mines. Saf Sci 48(2):173–178. https://doi.org/10.1016/j.ssci.2009.07.007
37. Yi W, Chan AP, Wang X, Wang J (2016) Development of an early-warning system for site work in hot and humid environments: a case study. Autom Const 62:101–113
38. Sarkar S, Patel A, Madaan S, Maiti J (2016) Prediction of occupational accidents using decision tree approach. In: 2016 IEEE annual India conference (INDICON). IEEE, pp 1–6
39. Sobhan S, Sammangi V, Rahul R, Maiti J, Mitra P (2019) Application of optimized machine learning techniques for prediction of occupational accidents. Comput Oper Res 106:210–224. https://doi.org/10.1016/j.cor.2018.02.021
40. Wang Y, Xu W (2018) Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud. Decis Support Syst 105:87–95
41. Wang Z, Ren J, Zhang D, Sun M, Jiang J (2018) A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos. Neurocomputing 287:68–83
42. Jiang M, Liang Y, Feng X, Fan X, Pei Z, Xue Y, Guan R (2018) Text classification based on deep belief network and softmax regression. Neural Comput Appl 29(1):61–70
43. Caliskan A, Yuksel ME, Badem H, Basturk A (2018) Performance improvement of deep neural network classifiers by a simple training strategy. Eng Appl Artif Intell 67:14–23
44. Hinton GE (1990) Connectionist learning procedures. Machine learning, vol 3. Elsevier, London, pp 555–610
45. Utgoff PE, Stracuzzi DJ (2002) Many-layered learning. Neural Comput 14(10):2497–2529
46. Chen J, Li K, Li K, Yu PS, Zeng Z (2021) Dynamic planning of bicycle stations in dockless public bicycle-sharing system using gated graph neural network. ACM Trans Intell Syst Technol (TIST) 12(2):1–22
47. Chen J, Li K, Rong H, Bilal K, Li K, Philip SY (2019) A periodicity-based parallel time series prediction algorithm in cloud computing environments. Inf Sci 496:506–537
48. Yu J, Hu B (2020) Influence of the combination of big data technology on the spark platform with deep learning on elevator safety monitoring efficiency. PloS One 15(6):e0234824
49. Chen J, Li K, Tang Z, Bilal K, Yu S, Weng C, Li K (2016) A parallel random forest algorithm for big data in a spark cloud computing environment. IEEE Trans Parallel Distrib Syst 28(4):919–933
50. Wen L, Li X, Gao L, Zhang Y (2017) A new convolutional neural network based data-driven fault diagnosis method. IEEE Trans Ind Electron 65(7):5990–5998
51. Hinton G, Deng L, Yu D, Dahl GE, Mohamed A-R, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN et al (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 29(6):82–97
52. Muhammad K, Ahmad J, Baik SW (2018) Early fire detection using convolutional neural networks during surveillance for effective disaster management. Neurocomputing 288:30–42
53. Uzair M, Shafait F, Ghanem B, Mian A (2018) Representation learning with deep extreme learning machines for efficient image set classification. Neural Comput Appl 30(4):1211–1223
54. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: semantic image segmentation with deep

convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848

55. Yan C, Hu J, Zhang C, (2018) Deep transformer: A framework for 2d text image rectification from planar transformations. Neurocomputing 288: 32–43
56. Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS (2016) Deep learning for visual understanding: a review. Neurocomputing 187:27–48
57. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE (2017) A survey of deep neural network architectures and their applications. Neurocomputing 234:11–26
58. Badem H, Basturk A, Caliskan A, Yuksel ME (2017) A new efficient training strategy for deep neural networks by hybridization of artificial bee colony and limited-memory bfgs optimization algorithms. Neurocomputing 266:506–526
59. Ng A, Autoencoder S, Cs294a lecture notes, Dosegljivo: https://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf.[Dostopano 20. 7. 2016]
60. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
61. Nesterov Y (2012) Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J Optim 22(2):341–362
62. Tan DS, Chen W-Y, Hua K-L (2018) Deepdemosaicking: adaptive image demosaicking via multiple deep fully convolutional networks. IEEE Trans Image Process 27(5):2408–2419
63. Le QV, Ngiam J, Coates A, Lahiri A, Prochnow B, Ng AY (2011) On optimization methods for deep learning. In: proceedings of the 28th international conference on machine learning, Omnipress, pp. 265–272
64. Kingma DP, Ba J, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980
65. Ruder S, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747
66. Jung Y (2018) Multiple predicting k-fold cross-validation for model selection. J Nonparamet Stat 30(1):197–215
67. Subasi A, Kevric J, Canbaz MA (2019) Epileptic seizure detection using hybrid machine learning methods. Neural Comput Appl 31(1):317–325
68. Maniruzzaman M, Kumar N, Abedin MM, Islam MS, Suri HS, El-Baz AS, Suri JS (2017) Comparative approaches for classification of diabetes mellitus data: machine learning paradigm. Comput Methods Progr Biomed 152:23–34
69. Griffiths TL, Steyvers M (2004) Finding scientific topics. Proc Natl Acad Sci 101:5228–5235
70. Cao J, Xia T, Li J, Zhang Y, Tang S (2009) A density-based method for adaptive LDA model selection. Neurocomputing 72(7):1775–1781. https://doi.org/10.1016/j.neucom.2008.06.011
71. Arun R, Suresh V, Madhavan CEV, Murty MN (2010) On finding the natural number of topics with latent dirichlet allocation: some observations. In: Zaki MJ, Yu JX, Ravindran B, Pudi V (eds) Advances in knowledge discovery and data mining. Springer, Cham, pp 391–402
72. Deveaud R, SanJuan E, Bellot P (2014) Accurate and effective latent concept modeling for Ad Hoc information retrieval. Document Numerique 17(1):61–84. https://doi.org/10.3166/DN.17.1.61-84
73. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
74. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
75. Suri NMR, Athithan G (2019) Outlier detection: techniques and applications. Springer, Cham
76. Zhang Y, Zhang E, Chen W (2016) Deep neural network for halftone image classification based on sparse auto-encoder. Eng Appl Artif Intell 50:245–255
77. Kurbiel T, Khaleghian S, Training of deep neural networks based on distance measures using rmsprop, arXiv preprint arXiv:1708.01911
78. Huber M, Imhof D (2019) Machine learning with screens for detecting bid-rigging cartels. Int J Ind Org 65:277–301
79. Xu X, Wang J, Peng H, Wu R (2019) Prediction of academic performance associated with internet usage behaviors using machine learning algorithms. Comput Human Behav 98:166–173
80. Li Z, Wu Q, Ci Y, Chen C, Chen X, Zhang G (2019) Using latent class analysis and mixed logit model to explore risk factors on driver injury severity in single-vehicle crashes. Accid Anal Prevent 129:230–240
81. Oztekin A, Al-Ebbini L, Sevkli Z, Delen D (2018) A decision analytic approach to predicting quality of life for lung transplant recipients: a hybrid genetic algorithms-based methodology. Eur J Oper Res 266(2):639–651
82. Sarkar S, Chain M, Nayak S, Maiti J (2019) Decision support system for prediction of occupational accident: a case study from a steel plant. In: Emerging technologies in data mining and information security. Springer, Singapore, pp 787–796