Applied Soft Computing Journal

Deep learning and Boosted trees for injuries prediction in power infrastructure projects

--Manuscript Draft--

Manuscript Number:	ASOC-D-19-02540R3
Article Type:	Full Length Article
Keywords:	Deep learning; Boosted trees; predictive analytics; Power infrastructure
Corresponding Author:	Lukumon O. Oyedele, PhD, LLM, MSc, BSc (Hons.) University of West of England, Bristol Bristol, UNITED KINGDOM
First Author:	Ahmed Oyedele
Order of Authors:	Ahmed Oyedele
	Anuoluwapo Ajayi
	Lukumon O. Oyedele, PhD, LLM, MSc, BSc (Hons.)
	Juan Manuel Davila Delgado
	Lukman Akanbi
	Olugbenga Akinade
	Hakeem Owolabi
	Muhammad Bilal
Abstract:	Electrical injury impacts are substantial and massive. Investments in electricity will continue to increase, leading to construction project complexities, which undoubtedly contribute to injuries and associated effects. Machine learning (ML) algorithms are used to quantify and model causes of injuries; however, conventional ML techniques do not produce optimal results since they require careful engineering to transform data into suitable forms. In this study, w e develop and compare state-of-the-art ML algorithms (deep learning and boosted trees) with other conventional methods to overcome this problem by analyzing incident cases obtained from a leading UK power infrastructure company. The predictive performance of the developed models was benchmarked using a statistical comparison between observations and predicted values. Results showed that the implementation of deep feedforward neural networks achieved the best predictions (accuracy= 0.967 and Cohen Kappa measure = 0.964). Furthermore, we conduct a sensitivity analysis to determine the effect of input parameters and data sizes on the mode s’ behavior. The sensitivity analysis results showed strong generalization abilities of the deep learning and boosted tree models and their effectiveness for safety risks management.

Deep learning and Boosted trees: for injuries prediction in power infrastructure projects

Highlights

- Presented deep learning and boosted tree approaches for safety management
- Benchmark deep learning models with other machine learning techniques
- Deep neural networks yield better prediction ability.
- Interpretable models for explaining the deep learning internal workings.

Deep learning and Boosted trees for injuries prediction in power infrastructure projects

Abstract

Electrical injury impacts are substantial and massive. Investments in electricity will continue to increase, leading to construction project complexities, which undoubtedly contribute to injuries and associated effects. Machine learning (ML) algorithms are used to quantify and model causes of injuries; however, conventional ML techniques do not produce optimal results since they require careful engineering to transform data into suitable forms. In this study, we develop and compare state-of-the-art ML algorithms (deep learning and boosted trees) with other conventional methods to overcome this problem by analyzing incident cases obtained from a leading UK power infrastructure company. The predictive performance of the developed models was benchmarked using a statistical comparison between observations and predicted values. Results showed that the implementation of deep feedforward neural networks achieved the best predictions (accuracy= 0.967 and Cohen Kappa measure = 0.964). Furthermore, we conduct a sensitivity analysis to determine the effect of input parameters and data sizes on the modes' behavior. The sensitivity analysis results showed strong generalization abilities of the deep learning and boosted tree models and their effectiveness for safety risks management.

Keywords: Deep learning, Boosted trees, Predictive analytics, Power infrastructure

1. Introduction

Working in power infrastructure sites and maintaining high-voltage overhead power lines is risky, and accidents involving live lines maintenance are lethal [1]. These operations are risky because constructing power infrastructure projects is characterized by continual changes or personnel movement, poor working conditions, and unsteady employment. Other causative factors are extensive use of resources and working in harsh environments (e.g., noise, vibration, dust, and severe weather) [2]. Old and weakened pipelines to facilities such as high-pressure gas mains and electric power substations also increase the potentials for unanticipated injuries, with impacts causing long-term physical and emotional distress to workers, their families and significant economic expenses. It is predicted that electrical utility companies will invest $1.5–$2.0 trillion in the power infrastructure by 2030 to keep up with increasing electricity demands [3]. This investment will increase the volume and complexity of constructing power infrastructure projects, which may lead to high frequency and severity of injuries and their associated monetary costs.

Machine learning (ML) algorithms, due to their flexibility, predictive and interpretative potentials, are used to quantify the contribution of causal factors concerning the occurrence of injuries [4]. However, conventional ML techniques do not produce optimal results. They are limited in processing data in raw forms since careful engineering and considerable domain expertise are required to design feature extractors. Feature extraction algorithms transform raw data into a suitable internal representation to enhance ML models' predictive ability [4], [5]. Studies previously conducted have aimed to predict injuries in construction and infrastructure projects for improved safety-related decisions [6], [7]. This interest is associated with increasing efforts being expended by electrical contractors and utility companies to minimize injuries. However, accident occurrence is still a challenge, as evidenced by several injury prevention studies [2], [8], [9].

Therefore, electrical utilities and contracting companies are in dire need of strategies to reduce injury frequencies and severities resulting from the increased volume and complexity of electrical infrastructure works [7], [10]. Traditional ML techniques required manual extraction of features from the large pool of incident datasets to transform data into internal forms [5] for guaranteed optimal results. Besides, the power infrastructure incident datasets are large, heterogeneous, and characterized by complex interactions between predictors [11], which may be difficult for conventional ML techniques to achieve optimal results. Thus, to optimally manage construction safety, a robust technique is desirable to enhance the predictive accuracy of non-fatal injuries for improved safety management in power infrastructure projects.

We chose the deep learning technique because it is good at discovering intricate structures in high-dimensional data [5] in addition to its remarkable problem-solving success in several domains. Additionally, it has outperformed other traditional ML methods like principal component regression, support vector machines (SVM), and shallow artificial neural networks (ANNs) due to its superior representation ability in speech processing, image recognition, and natural language processing [12]. We also benchmark the predictive accuracies of deep neural networks with two powerful boosted tree techniques- gradient boosted machines (GBM) [13] and extreme gradient boosting (XGB) [14], and two conventional ML techniques, SVM and k-nearest neighbors (KNN). Reviewing the literature on the accident analysis domain, techniques like SVM and KNN are popular because they have a robust theoretical grounding that facilitates learning from data and the capability to handle complexities not related to the computational complexity of the problem at hand [15]. GBM, however, outperforms other conventional ML models due to its ability to capture nonlinear and local relationships among predictors and targets [4]. In contrast, XGB is an accurate and efficient scalable implementation of GBM [16].

The rest of the paper is organized as follows: in the next section, we present a review of the literature. In Section 3, we discuss the methodology, in particular, incident datasets and pre-processing, formal concepts of deep neural networks, boosted trees, and other ML techniques. We also discuss the predictive models' development and parameters tuning. In section 4, we present prediction results and the interpretability of the deep learning model. Finally, Section 5 concludes the study and gives possible future research directions.

2. Related work

In the section, we discuss some pertinent issues in the literature, such as associated risk factors with non-fatal injuries in the power infrastructure industry and ML techniques for injury prediction and safety management.

2.1. Factors associated with non-fatal injuries

In deepening the understanding of accident occurrence, several studies have been conducted to investigate and identify factors causing injuries in the power infrastructure construction. A leading indicator such as workplace activities (electrical works, heavy-lifting operations, manual handling, working at heights, and driving) are a significant cause of injuries [10], [17]. Electrical works, for instance, cause workers to sustain lost work times due to burns injuries [17], and manual handling of tools (i.e., ratchet cutters, manually operated presses, hammers, and ladders) also increases injury risk [18]. These manual tools are also hazardous to linemen due to increased exposure to vibrations, awkward postures, and repetitions [10], [19]. Similarly, heavy equipment operations (soil and material handling tasks) at construction sites produce complexities due to space limitations and constraints caused by the competing project components (i.e., tasks, crews, and materials). These complexities are indicative of the high occurrence of struck-by and crushed-by injuries during work activities involving heavy equipment such as cranes and boom trucks [20].

Also, harsh weather, working environment, inefficient site planning and controls, and improper use of personal protective equipment contribute to injuries [7], [20], [21]. The significance of project-related attributes (project characteristics, cost, and other factors such as location, employee age, employee experience, time of the day, employee type, day of the week) have also been established in past studies [21], [22]. For instance, lack of safety training and unavailability of technical and safety instructions are part of significant reasons for injuries [7]. This study builds on these studies by implementing robust deep learning techniques to discern intricate structures and relationships amongst these factors in a high-dimensional power infrastructure incident dataset.

2.2. Machine learning techniques for predicting injuries

Several ML algorithms (depicted in Table 1) have been employed to analyze and predict construction work-related injuries. Commonly used conventional ML techniques are linear regression, support vector machines, decision trees, and artificial neural networks. Regression-type techniques such as generalized linear, logistic, and probability models have been used to identify factors affecting construction injuries. For instance, a logistic regression model was developed to analyze and predict the risk of roof fall injuries and occupationally induced human injuries [21], [23].

Though the logistic regression model assumes that the independent variables are not dependent on each other, it is sometimes difficult to find such models. Another limitation is its difficulty capturing the nonlinear and local relationships among dependent and independent variables [4]. The KNN method is a well-known classification algorithm used in pattern recognition, and due to its simplicity, it has been used to classify workers according to their risk of suffering musculoskeletal disorders [8]. However, Liu *et al*. [24] revealed its difficulties in classifying close objects originating from different classes correctly. Due to its important characteristics, such as the ability to learn from data and fault tolerances, artificial neural networks (ANN) have been used in construction to analyze and manage safety conditions. [15], [25], [26]. However, ANN still suffers from the uncontrolled convergence speed and local optima.

Table 1. Previous studies using ML approaches for injury analysis

Reference	Algorithms	Aim
[27], [28]	Decision trees (DT) and Associative rules	ML models to identify factors influencing injuries at workplace.
[8]	KNN	Classify workers based on their risk of suffering musculoskeletal disorders.
[6]	SVM, DT, and Bayesian networks	Analyze injury data and identify most relevant variables to improve prediction capability.
[15]	ANN, SVM, Genetic Algorithm (GA)	ML techniques to predict occupational incident outcomes.
[25], [29]	ANN	Neural networks to analyze incident dataset and manage safety conditions.
[26]	ANN	for developing early warning systems for construction workers
[30]	Ontology-based classifier with SVM	Ontology-based text classification with SVM to match safe approaches identified in existing resources with unsafe scenarios.
[4]	Random Forest (RF) and Stochastic Gradient Tree Boosting (SGTB)	ML models to predict safety outcomes from construction attributes.
[7]	GBM; Particle Swarm Optimization	ML models to find complex patterns and reduce unrelated attributes in datasets for optimal future decision-making.
[21], [23]	Logistic Regression, RF, SVM, DT	Compared ML techniques and developed a model that produces safety leading indicators for predicting safety risks.

Also, parameters of neural networks with more than two layers are difficult to optimize using the traditional gradient descent [31]. Furthermore, SVM, due to its low computational costs and accessible optima, is often used for pattern recognition and classification problems [15]. SVM is also efficient for small data sample problems and classifying workers suffering from work-related injuries [21], [30]. However, its computational complexity grows exponentially with the size of training samples [31].

Decision trees [6], [27], [28], have also been used as a non-parametric tool based on the rule induction to analyze the occupational injury data. However, Bengio *et al*. [32] revealed their inability to generalize to examples not seen in the training set. Recently random forest [4], [21], and gradient boosting machines [4] approaches have made great attempts at analyzing construction injury data and predicting injuries, but the error rates are still unsatisfactory.

Apart from attendant limitations confronting these conventional models, a significant problem is their limited ability to process raw data as considerable efforts are needed to transform the raw data into the appropriate internal form [5]. A dataset of carefully selected features has to be manually extracted from a large pool of an incident dataset for these techniques to achieve high prediction accuracy [4]. Data and algorithm-level methods are continually improving, and hybrid approaches are gaining momentum with current researches focusing on computationally efficient methods for analyzing data are evolving. Based on the preceding, state-of-the-art techniques such as deep neural networks are used to analyze incident datasets to improve injury prediction accuracy for safety risk management. Deep learning has gained increasing attention and motivated numerous successful applications [33] [12], where they outperform traditional methods such as principal component regression (PCR), SVM, and ANN [31]. Moreover, unsupervised and supervised learning techniques are appropriately integrated to yield a semi-supervised model [34], which is lacking in traditional ML techniques. Up till now, deep learning applications have not been profound in injury management and safety modeling in the power infrastructure sector with incident datasets characterized by complex interactions of predictors [11].

Thus, the study employs the advantages of deep feedforward neural networks to evolve accurate predictive models to analyze incident datasets for enhanced safety management. We also benchmark the performance of deep feedforward neural networks with boosted tree techniques (GBM and XGB). As reported in the literature, GBM and XGB outperformed other conventional ML techniques in injury outcomes prediction [4], [16].

3. Methodology

In this section, we discuss the methodology adopted, namely the incident dataset relevant to the domain of interest and ML techniques (deep learning and boosted tree techniques) employed. We also discuss the development of predictive models using tested industry-based techniques and adopting rigorous performance testing to guarantee accurate and robust classification.

3.1. Incident datasets

We obtained incident cases related to power infrastructure projects between 2004 and 2016 from a leading UK power infrastructure company. Though the dataset is enormous, there are also challenges such as name disambiguation, spelling errors, duplications, missing entries, and skewed data distribution. For some instances of missing entries, especially text attributes, we derived attributes from existing attributes using string functions in R to extract appropriate attributes. The derived attributes are used as substitutes for unreported instances. For other cases, the mean/median imputation technique (the mean for all samples belonging to the same class) was used in filling missing values. To handle outliers, we used box plots to detect outliers and filter them out appropriately, while sorting and spell checking are used to eliminate duplicates and spelling errors. In solving name disambiguation, we used string processing and pattern matching techniques to find attributes referring to the same entity but written differently. For instance, for the predictor "EQP_T," the equipment named "Mobile elevating work platform" appeared in the dataset as an aerial work platform, or bucket truck, or MEWP, or elevating work platform. All these terms are written as 'MEWP' to improve prediction accuracy.

After the data preprocessing stage, we observe significantly fewer reported cases of certain classes of the dependent variable. This imbalanced data issue is prevalent in many real-world data sets, and if not handled, it will skew prediction results considerably in favor of the majority class [35]. We handle this problem using the under-sampling technique because of its popularity and shorter training times [36]. We also removed from the “Injury type” predictor, attributes such as loss or damage, legal, quality, complaints, and security that have no direct impact on the current study.

As a result of the sensitivity of data involved and in conformity with the EU General Data Protection Regulation (GDPR), we anonymized the data in other not to compromise the privacy of subjects. We followed Sarkar et al. [15] recommendation that ML algorithms produce better results with numerical features than categorical features. After eliminating outliers, we normalized and standardized the dataset using the Z-score method, the regularly used score normalization method [37]. The total number of predictors in the dataset was initially twenty-eight. The selected predictors are based on the Gedeon feature selection method [38], which used the brute force technique to determine the functional contribution of variables to outputs. The predictors derived using a deep learning model with default parameters (trained without parameters tuning) are depicted in Table 2, Also shown in Table 2 are studies employing these predictors in modeling work-related injuries together with the outcome variable (IBP), representing the injured human body parts.

Typically, ML methods require datasets to be split into three subsets (training, validation, and test) for benchmarking. The historical incident dataset (168,574 data cells) was divided into training, validation, and test, with a split ratio of 8:1:1. The training set (80% of the sample) was used for training, the validation set (10%) for tuning hyperparameters, and the test set (10%) for evaluating models. Fig. 1 gives an overview of the steps adopted in this study for managing non-fatal injuries. The steps include data pre-processing, training and testing of models, benchmarking models' performance, and creating interpretable models for the best classification model.

Table 2. Predictor and outcome variables

Variable ID	Min	Mean	Max	Description	Reference
DURATION	-1.36	0.00	1.88	The length of the construction period: very short, short, medium, long, very long	[27]
COST	-1.46	0.00	1.81	The cost of projects, classified as very low, low, moderate, expensive, very expensive	[21], [28]
PROJ_T	-1.28	0.00	1.25	Identifies a specific project type (overhead lines, underground cabling, substation)	[21], [28]
PROJ_C	-0.89	0.00	2.67	Determines the project complexity, i.e., whether the project is a new build, maintenance, or refurbishment.	[39]
REGION	-2.09	0.00	1.08	The five UK regions where projects are constructed.	[28]
LOC	-1.12	0.00	2.64	The project site, location, or work environment.	[22]
CLT	-2.11	0.00	2.65	The client contracting out the project, i.e., Energy companies, communications, digital, and power supplier contractors.	[40]
EQP_T	-1.90	0.00	3.12	Equipment types (Elevator, Drill, Hammer, Haulage.)	[20]
EQP_S	-1.20	0.00	1.19	The state of the equipment (Good, moderately in good condition, not in good condition).	[40]
EMP_A	-0.73	0.00	1.96	Age expressed in a predefined range (16-25,26-44, 45-60).	[41]
EMP_E	-1.40	0.00	1.54	Qualification and the length of time on the job (<1 year, 1-3 years, >3years)	[22]
EMP_C	-0.78	0.00	1.28	Employment contract type, defined as either temporary or permanent.	[22]
YEAR	-3.26	0.00	1.86	The year of project construction.	[29]
SEASON	-1.19	0.00	1.46	Seasons (winter, spring, summer, autumn).	[41]
MONTH	-1.67	0.00	1.55	The month (1-12) of construction.	[41]
TIME	-1.02	0.00	0.98	Time (6 am-12 pm early in the day) or (12 pm-19 pm later in the day).	[7], [41]
DAY	-1.53	0.00	1.63	Day name, i.e., Monday, Tuesday, Wednesday, Thursday	[7]
CS	-1.49	0.00	1.15	The contract status of the employing company i.e. main contractor, or subcontracted, or third-party company.	[8]
TASK	-1.84	0.00	1.32	The specific tasks (i.e., lifting, cutting, loading, pushing, electricals).	[10], [17]
SHIFT_W	-1.03	0.00	0.97	The form of the work shift (fixed or rotating)	[42]
WSLC	-1.09	0.00	1.37	The working surface layout condition, i.e., Good condition, moderately in good condition, not in good condition.	[8]
HSRM	-1.52	0.00	1.13	Safety risk management policies, i.e., risk management with supervision/control, risk management policy but no supervision/control, and no risk management policy/supervision or control.	[21]
IBP (Body *parts*)	Output variable A factor with14 levels (1,2, 3, …14)			Body parts namely, 1-ankle/foot, 2-arm/elbow, 3-back/buttocks, 4-chest/abdomen, 5-ears, 6-eyes, 7-face/shin, 8-groin/hip, 9-fingers, 10-head, 11-knee/leg, 12-multiple parts, 13-neck/shoulder, and 14-no injury	[4]

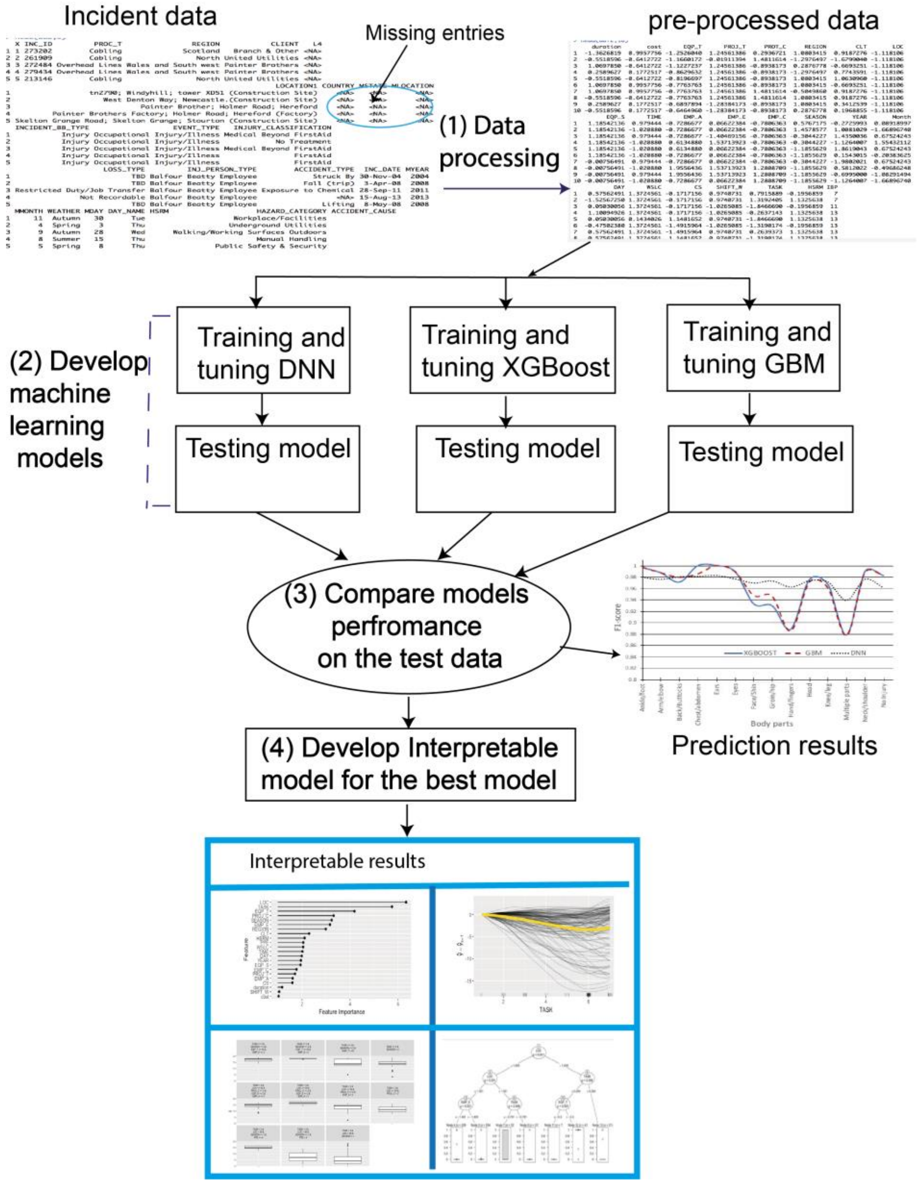


Fig. 1. Steps adopted for managing injuries

3.2. Deep learning

Deep learning is a new ML research area exploiting multiple layers of information processing in a hierarchical architecture for pattern classification and representation learning [34]. Deep feedforward neural networks consist of interconnected neurons, each receiving some inputs and supplying outputs. Each node in the output layer performs weighted sum computation on the values received from input nodes to generate outputs using simple nonlinear transformation functions. Changes to weights are made in response to individual errors encountered by the networks exhibit at the output nodes. Such corrections are usually made using stochastic gradient descent [43]. Deep learning models can achieve accuracy, sometimes exceeding human-level performance. They can derive high-level, complicated abstractions and data representations from massive datasets, making them attractive and suitable for Big Data Analytics. Empirical studies have confirmed the exceptional results of deep learning models in different ML applications, including speech recognition, computer vision, and natural language processing [5], [12], [33].

In this study, we developed deep feedforward networks, a topology in which all nodes are organized into sequential layers, with every node receiving inputs only from nodes in previous layers. Fig. 2 depicts a structure of the deep feedforward neural networks with an input layer (I = 22 neurons), 2 hidden layers (H1 = H2 = 500 neurons), and an output layer (O = 14 neurons). According to Lecun et al. [5], feedforward neural networks have the advantage of automatically discovering representations needed for detecting complex functions in raw data. A DNN is a function that finds a predictor of an output Y given an input X, i.e., $Y = f(X)$. The mapping f(.) is usually parameterized by weights and optimized during the learning process. DNN uses data examples to train a model to make predictions while passing data features through the different hidden layers with many neurons existing in each layer [44].

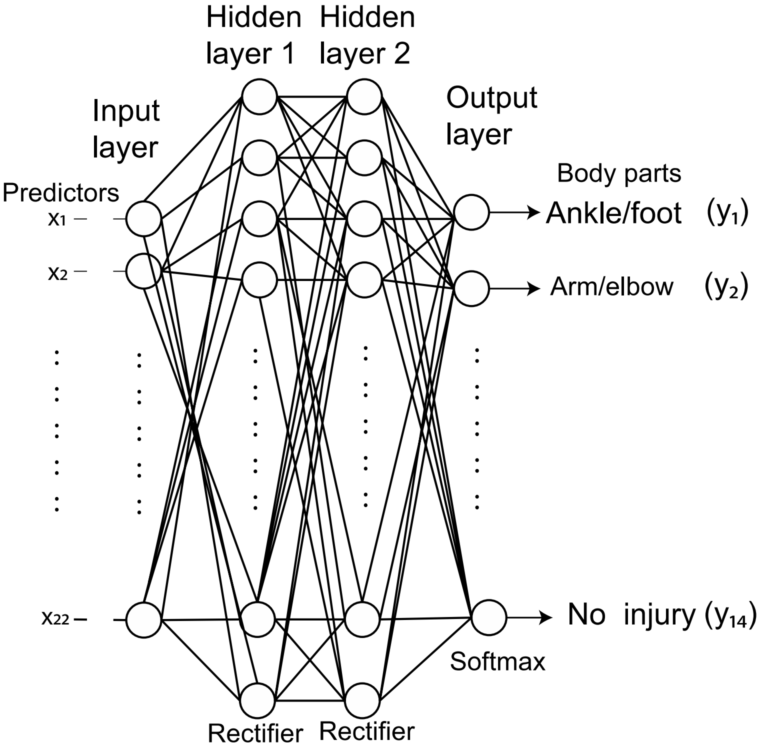


Fig. 2. Feedforward neural networks topology

The existence of multiple levels of representations in DNN distinguishes it from the traditional artificial neural networks that allow the discovery of intricate structures in high-dimensional data rather than learning key features of data designed by human engineers [5].

We formally describe deep learning as follows. Denote the output of a neuron at layer ℓ by h^{ℓ}, and its input vector from a previous layer by $h^{\ell-1}$, then we define the activation of neurons as $h^{\ell} = \sigma\left(b^{\ell} + W^{\ell}h^{\ell-1}\right)$. Where b^{ℓ} is a vector of biases, W^{ℓ} is a matrix of weights and $\sigma(\cdot)$ is the activation function (i.e., tanh, rectifier, maxout) employed to improve the model's training. At the input layer, the input vector, $x = h^{o}$, is analyzed by the network and the output vector h^{ℓ}(in the output layer) is used to make predictions. Stochastic gradient descent is often used to optimize neural networks [5]. This method consists of showing the machine, an input vector of some patterns, computing outputs together with errors, estimating the average gradient for the patterns, and accordingly modifying the weights. The procedure is repeated for many subsets of patterns in the training set until the average of the objective function converges.

Two primary data types are handled by the ML models, the continuous (numerical) and categorical types, described in Table 2. The categorical form is represented internally using the One-Hot encoding, the most common and often recommended approach.

3.2.1 Building feedforward neural networks

Constructing a deep architecture for any problem involves defining the number of layers and neurons in each layer. Neurons determine the complexity of a deep learning model, and the more complex a model is, the more it is prone to overfitting. Deciding the number of neurons in the hidden layers is essential to the architecture of neural networks. We obtain a suitable value for this parameter by adding new neurons incrementally to grow the network, i.e., using the training data and an untuned single-layer network, we varied the number of neurons from 100 to 800 while keeping other parameters at their default values. The resulting value is then used to determine the suitable number of layers by trying out different network topologies (1-layer, 2-layer, and 3-layer, respectively) while keeping other parameters at default values and training the network for 125 epochs.

We address overfitting using Lasso regression (*L1)* and Ridge regression (L2), which are the most common types of regularization [47], [48]. L1 and L2 parameters are used to update the general cost function by adding a regularization term that decreases the values of the weight matrices. L2 is defined in Eq. (1), where k represents the number of cases, l is the sum of squares of residuals, w represents weights, and λ is the regularization parameter whose value is optimized for more reliable results.

$$cost = l + \frac{\lambda}{2k} * \sum \|w\|^2 \qquad (1)$$

L2 regularization forces weights to decay towards zero and is computationally efficient. *T*he absolute value of weights is reduced to zero in L1 *(Eq. (2))*, and it is computationally inefficient since it uses an iterative fitting technique.

$$cost = l + \frac{\lambda}{2k} * \sum [\![w]\!] \qquad (2)$$

It is presumed that deep neural networks with smaller weight matrices lead to a simpler model, which in turn reduces overfitting. In this study, the optimal values of L1 and L2 arrived at using the random search algorithm are 1e-6 and 1e-7, respectively.

We employed Softmax as the output layer's activation function and used ADADELTA [43], [49], an advanced optimization routine to train the network with the adaptive learning rate time smoothing factor (ϵ) set at 1e-8, and the learning rate decay factor (ρ) at 0.9999, as they resulted in improved classification accuracy. Using ADADELTA aims to overcome the sensitivity in selecting hyperparameters and the continual decay of learning rates [49]. The simulation was carried out on a single-core, with a seed parameter at -1 to realize reproducibility. Other potential tuning parameters are set to their H2O default values.

3.3. Boosted trees

This ML algorithm is an ensemble of multiple weak trees to improve robustness over a single predictive model [45]. The idea of a boosted tree algorithm is that many decision trees perform better than a single one. The benefits of using this approach are: first, it is nonparametric (it does not require assumptions about the data), and the number of parameters (parameters related only to algorithms themselves) grows in symmetry to the number of the training set. Second, it is quick and easy to implement. Two examples of boosted tree algorithms considered in this study are Gradient Boosted Machines (GBM) and Extreme Gradient Boosting (XGB). GBM works by sequentially applying weak learners to repeatedly re-weighted versions of the training data. After every boosting iteration, the model increases the weights of misclassified examples and lowers the weights of correctly classified examples. Hence, each successive classifier focuses on examples that are hard to classify in previous steps. After a series of repetitions, a group of weak classifiers' is combined by a weighted majority vote into a final prediction. XGB is an optimized version of GBM, designed for speed and performance. XGB is a scalable end-to-end tree boosting system that handles massive data using the following: cache-aware pre-read technology, distributed memory computing technology, and AllReduce fault-tolerant tools to improve the computation speed of the existing boosting tree algorithm [46].

For boosted tree algorithm, small depth trees (decision trees) are created on a sample of rows and features at each step, and these trees are used to devise a prediction. A decision tree is a flowchart-like structure with each internal node representing a "test" on an attribute (e.g., the lineman's sex or whether a lineman uses personal protective equipment or not), each branch is an outcome of the test, and each leaf node is a class label representing a decision to be taken. The paths from the root to the leaf represent the classification rules.

For instance, Fig. 3 illustrates a simple example of using two Classification and Regression Trees (CART) to predict one of two probable outcomes (“no injury” or “hand injury”) using predictors project type, equipment, and the operation type. The algorithm classifies members of a family into different leaves. In Fig. 3, hands and forks (manual operations) are grouped, while hands, shoulders, and the back are body parts classified together for lifting operations. The algorithm then assigns each leave a score (i.e., -0.2, 0.1, 2, 0.7, and -0.7), as depicted in Fig. 3. The prediction scores of each tree are then summed up to get the final score. The smaller the score, the better the structure is. Here, the algorithm predicts a hand injury since it has a smaller score.

Mathematically, given a dataset with n examples and m features, D=$\{(x_i, y_i)\}(|D| = n, x_i \epsilon R^m)$, a tree ensemble model uses K additive functions to predict the output. The space of regression trees (say F) is defined in Eq. (3), where m denotes the number of features, q denotes the structure of each tree and w denotes the weight vector of each leaf.

$$F = \{f(x) = w_{q(x)}\}(q: R^m \longrightarrow T, w \epsilon R^m) \qquad (3)$$

Decision rules in the trees (q) are used to partition the dataset recursively into leaves, and the final prediction is calculated by summing up scores in corresponding leaves (given by w). We give the equation of this final prediction value in Eq. (4).

$$\hat{y}_i = \Phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \epsilon F \qquad (4)$$

Learning a tree ensemble involves optimizing the regularized objective function in Eq. (5).

$$L(\Phi) = \sum_{k} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k), \Omega(f) \qquad (5)$$

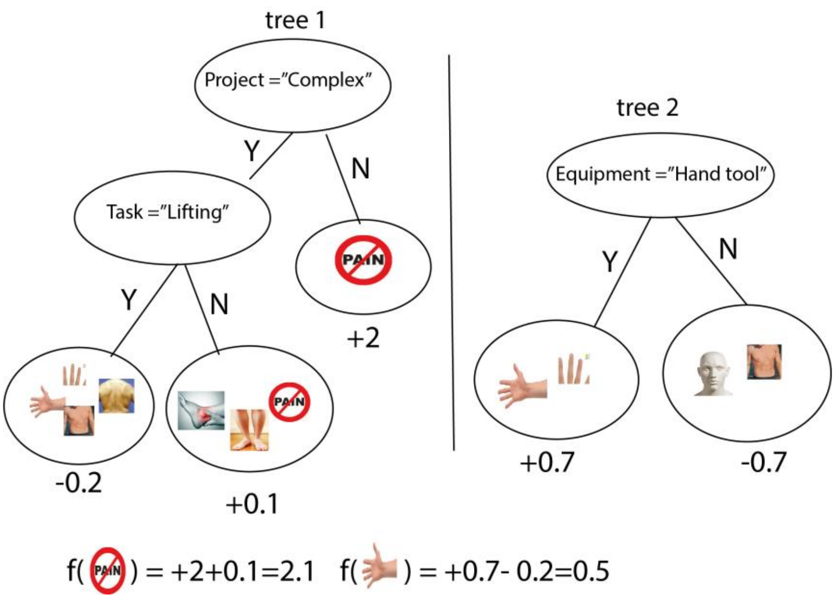


Fig. 3. Injury to body parts classification with boosted trees

The objective function (Eq. 5) calculates the quality of the training set, and Ω calculates the complexity of the model in avoiding overfitting. That is, $\Omega(f) = \alpha T + \frac{1}{2}\beta\|w\|^2$, where α and β are parameters controlling the tree size (or depth) and the minimum number of samples in terminal nodes (leaves), respectively.

3.3.1 Tuning gradient boosted machines (GBM)

For most ML algorithms, GBM inclusive, the issue of overfitting, is a concern. Overfitting is the tendency of a model to fit the training data too well at the expense of generalization. This situation occurs when an unusually high number of trees and tree depths are used. Several approaches, i.e., evolutionary, genetic algorithm, and random search, can be used to determine these parameters' optimum values. In this study, we used the search grid method due to its simplicity. The search grid method involves defining a grid of hyper-parameters combination, building a model for each combination, and selecting the optimal combination using appropriate metrics to quantify the model performance on the testing set. We used AdaBoost [50] to deploy decision trees as weak learners and tuned key four hyper-parameters for enhanced classification performance. The four parameters tuned are the number of trees (*ntree*), tree depth (*max_depth*), the learning rate (*learn_rate*), and the column sample rate per tree (*col_sample_rate*). Boosting may potentially overfit for large *ntree*; hence we limited the number of iterations to 80 - a very conservative value compared to examples provided in the literature [42].

The tree depth between 4 and 8 has been empirically shown to give the best results [42]. Moreover, stumps with only one split allow for no variable interaction effects. Thus, we used a tree depth of 4 to allow for five-way interaction. Hastie et al. [42] recommended learning rates lower than 0.1, but considering the small number of trees used, and a computationally feasible model, we set the learning rate at 0.01. The column sample rate per tree (*col_sample_rate*) is from 0.0 to 1.0, and we used the default value (i.e.,1) since this value reduces the miscalculation error far better than other values. We derived the optimal number of trees (130) using the random search strategy, and set other tuning parameters to their default values. The seed value was set at -1 to support reproducibility.

3.3.2 Tuning extreme gradient boosting (XGB)

XGB is a supervised learning algorithm that predicts an outcome by combining the estimates of a set of simpler and weaker models [46]. XGB can compute parallel operations on a single machine and learn iteratively from previously built weaker models to minimize error rates. We build the XGB model iteratively by tinkering with three hyperparameters to reduce error rates. The iteration that produces the minimum error rate is selected for the prediction problem. The three crucial hyperparameters of XGB are the number of iterations or the number of trees (*ntree*), the maximum tree depth (*max_depth*), and the learning rate (*learn_rate*) [46]. Similarly, we fixed the maximum tree depth and learning rate values at 4 and 0.01, respectively. Low values are set for these parameters to prevent overfitting and ensure the model's generalization on new data. For the number of iterations, we used the random search strategy to find the optimal setting for this parameter, and 120 was arrived at as the optimal number of iterations.

3.4. Conventional machine learning techniques

3.4.1. Support vector machine (SVM)

SVM is a universal approximator of any multivariate function to any desired level of accuracy. It has been used in different engineering fields with good accuracy, and theoretically, it has lesser overfitting problems and generalizes well. However, the main problem with the SVM model is the selection of the training parameter values. Inappropriate parameter setting often leads to poor prediction accuracy. We refer readers to [30] for the basic understanding of the SVM working principles.

3.4.2. k-Nearest neighbor (KNN)

KNN classifies an observation by looking at the closest k observations. The nearest neighbor decision rule is used to assign a new sample point to the classification based on the nearest of a set of previously classified points. The two decisions needed in the KNN algorithm are the value of k and the distance function. The value of k is determined by trying different values and finding the best with the highest prediction accuracy. The Euclidean distance, interpreted as the physical distance between two-dimensional points, is used for computing the distance function in the KNN. The readers may refer to [8] for a summary of the algorithm.

3.5. Performance evaluation

To evaluate the performance of the classification models, we used per-class metrics such as precision, recall, and F-1 score. Precision is a fraction of correct predictions for a specific class, while recall is the model's ability to classify relevant cases. F-1 score defines the harmonic mean (or a weighted average) of precision and recall, and it reaches its best value at one and its worst at zero. These metrics are defined in Eq. (6), where TP (True Positive) denotes data points correctly labeled or predicted. The false positive (FP) signifies an outcome *incorrectly* predicted as the *positive* class by a model, while true negatives (TN) are data points incorrectly labeled as negatives. False negatives (FN) denote outcomes *wrongly* predicted as the *negative* class. For the overall prediction performance of models, the per-class metrics (precision, recall, and F1-score) are averaged (i.e., $macroPrecision = {\sum_i^n precision_i}/_n$) over all the classes to give the macro-averaged precision, recall, and F-1.

$$\begin{gathered}accuracy = \frac{TP + TN}{TP + TN + FP + FN} \\ precision = \frac{TP}{TP + FP} \\ recall = \frac{TP}{TP + FN} \\ f1_{score} = 2 * \frac{precision * recall}{precision + recall}\end{gathered} \tag{6}$$

Similarly, we used the Cohen Kappa statistic [51], a measure of agreement between the actual and predicted labels. This metric, defined as $k = (t - y)/(1 - y)$, where y is the model output, and t is the desired output, offers a useful measure to handle both multi-class and imbalanced class problems effectively. The closer to 1 a Kappa metric is, the better the classifier when compared to a random chance classifier. For all the models, an oversampling technique to tackle the imbalanced class issues was used. Models for deep feedforward neural networks and boosted tree algorithms are constructed using the training and validation sets described earlier.

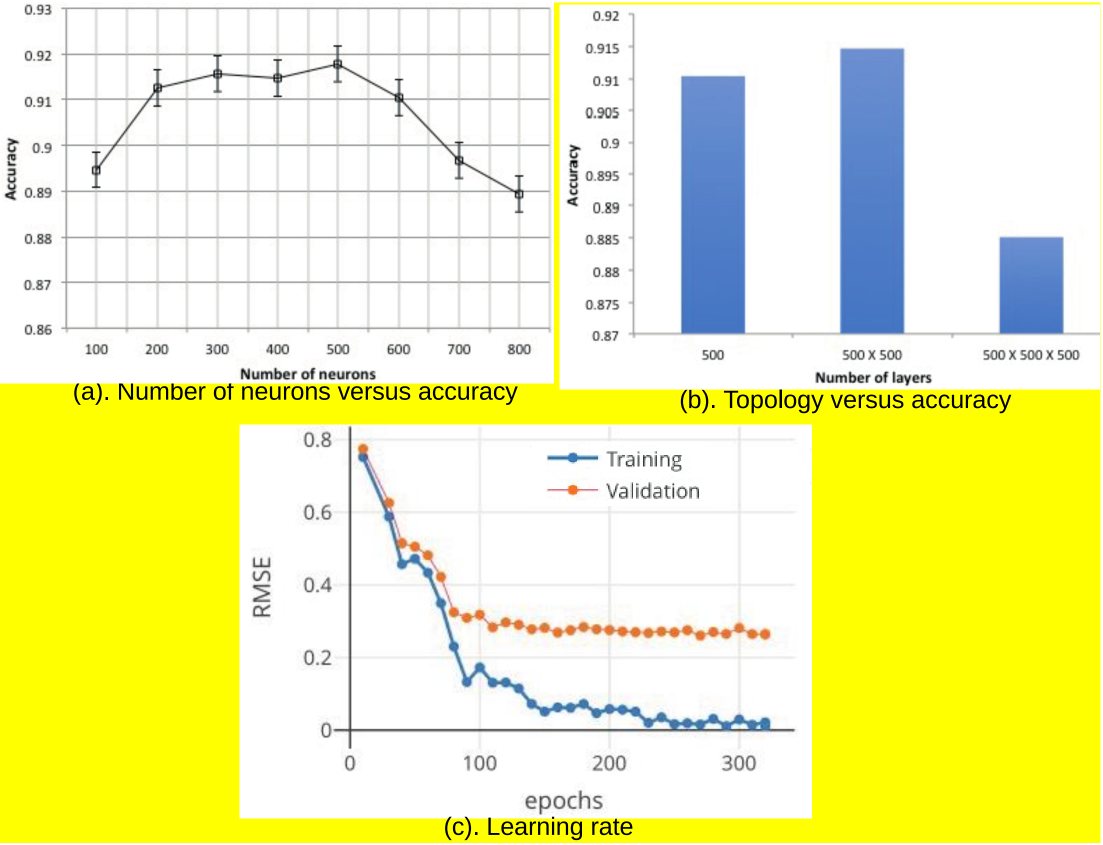
All the development and experimental works are carried out on the Intel Core i5 2.50 GHz with 32GB RAM. Subsequently, we evaluated the models' outputs against the testing sample, and the models' performance metrics are calculated appropriately. The testing set was randomly divided into two subsets (Test A and Test B) in the ratio of 60:40 to determine the effects of different data sizes on the models' performance and sensitivity.

4. Results and discussion

4.1. Determining model's parameters

The optimal number of neurons (500) and network topology (a two-layer network (500 X 500) arrived at with the highest prediction ability is as depicted in Figs 4a and 4b. We settled for this network (2 hidden layers, each with 500 neurons) as the base deep learning model. A topology of '22-500-500-14' with an input layer matching the 22 predictors, two hidden layers (each with 500 neurons), an output layer for the outcome variable (13 body parts representing 13 classes), and the 14th class representing no injury. In arriving at this topology, we followed the recommendation of LeCun et al. [5] by setting the number of neurons in the first hidden layer of the network to correspond to the size of the input and then using regularization techniques to address any possible overfitting.

Fig. 4c shows the root-mean-square error (RMSE) curve for the different iterations on the training and validation sets, with the optimal iteration at 230. This steady slope (at epoch = 230) signifies that the model generalizes well enough on the validation set. The rectified linear was used as the activation function for the two hidden layers because of its popularity, and it has demonstrated high performance in computer vision research [5].



(a). Number of neurons versus accuracy

(b). Topology versus accuracy

(c). Learning rate

Fig. 4. Designing the optimal deep learning architecture

4.2. Feature selection

We show the topmost ten significant predictors (using scaled importance) derived by the three models in Fig. 5. For the DNN model, the significant predictors are selected using the Gedeon feature selection method [38], and by the Gini index for both XGB and GBM, respectively. As shown in Fig. 5, the predictors *LOC* (location), *EQP_T* (equipment), and *TASK* (operations) have the highest explanatory power, as confirmed by all models. After these top three influential predictors, each model finds its unique structure and signals within the data. Other predictors identified are *REGION*, *EMP_E* (employee experience), PROJ_C, SEASON, DAY, HSRM, CLT, REGION, EQP_S, *DURATION* (project duration), and YEAR. The spread between *LOC* (top) and *EQP_S* (bottom) for the GBM model is more pronounced when compared to other models because of its lower tree depth. The first three critical predictors extracted by the three models agree with the literature [28].

The predictor LOC refers to the project site characteristics such as terrains, ground conditions, structures, unsafe conditions, site logistics, and rapidly changing environments. For instance, slippery ground surfaces in winter or rainy periods result in injury to ankles while working or walking outdoors on muddy and unmade surfaces. Also, windy locations can blow grits or dust in the eyes, especially when no or defective safety glasses are worn. Besides, a restricted site location is a significant cause for traps and struck by/against events. These features have a positive association with unsafe behaviors and injuries [28]. Furthermore, equipment and operations have also been identified as critical sources of injuries. According to Hinze and

Teizer [52], outcomes from injuries with different equipment types such as dump trucks, forklifts, excavators, and graders are prevalent in visibility-related injuries.

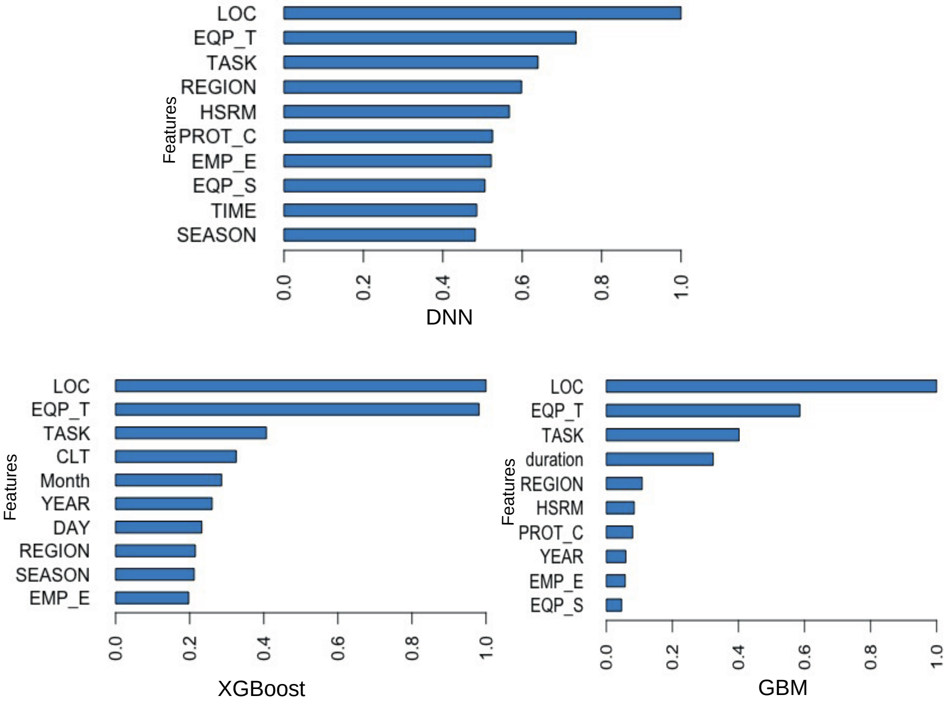


Fig. 5. Top ten predictors by DNN, XGB, and GBM

Being caught in/with equipment or electric shocks is closely related to power machinery [22], including drum machines and mixer-grinders. These injuries are caused mostly by the inappropriate use of safety and protective devices when operating, loading, and transporting power equipment. Also, construction-related tasks, such as plant operations, working at heights, and manual handling operations, coupled with workers' attitudes and behaviors, are sources of injuries [53]. Other predictors (project characteristics, employee experience, day, and season) identified in this study corroborated findings from the literature [21], [22].

4.3. Prediction accuracy of models

The classification accuracy of models for the outcome variable "IBP" is summarized in Table 3. The predictive accuracies and Kappa measures of models on different data sizes (Test A, Test B, and All test) are almost equal irrespective of their size, with the absolute difference ranging from 7e-3 to 1.3e-2. Different data sizes are insignificant for their performance since they produced highly consistent results that are invariant across samples. This attribute makes predictions stable for new examples.

Table 3. Predictive performance of models

Metric	GBM			XGB			DNN		
	Test A	Test B	All test	Test A	Test B	All test	Test A	Test B	All test
Accuracy	0.963	0.950	0.958	0.966	0.961	0.963	0.970	0.963	0.967
Kappa	0.959	0.945	0.954	0.963	0.957	0.961	0.967	0.960	0.964

Accordingly, based on the overall performance, as depicted in Table 3, DNN is considered the best model with the highest predictive accuracy (0.967) and Kappa value (0.964). XGB ranked second with accuracy (0.964) and Kappa value (0.961), while GBM came last with a classification accuracy and Kappa value of 0.958 and 0.954, respectively. Though both XGB and GBM followed the principle of gradient boosting, XGB focuses on the computational power and uses a more regularized model formalization. This formalization makes it resistant to overfitting and robust to noisy feature space, thus, giving it a better performance over the GBM technique [46]. We noted that all the models predicted reasonably well with high accuracy and within the limits of Kappa's substantial agreement.

The precision, recall, and F1-score metrics for all the models are presented in Table 4. The recall measure gives useful insights into the specific performance of a class, and in this study, the predictive performance of models on injured body parts is beneficial. Again, DNN recorded high recall (0.97) for most body parts except "arm/elbow", "hand/fingers", and "Knee/leg"; this means there is a 3% probability of misclassifying an observed injured body part. XGB (overall recall = 0.97) and GBM (overall recall = 0.96), in contrast to DNN, had a slightly higher recall for the classes – "arm/elbow", "chest/abdomen", and "head". The SVM model had an overall recall of 0;91, while KNN had the lowest recall (0.77).

The following body parts: backs/buttocks, ankle/foot, eyes, groin/hip, neck/shoulder, chest/abdomen, arm/elbow, hand/fingers, and head are predicted accurately by the models (high recall values). The primary sources of injuries on these body parts are caught in/between, struck by objects, manual handling operations, working outdoor, and walking on uneven or slippery surfaces. Similarly, the overall precision for DNN, XGB, GBM, SVM, and KNN are 0.97, 0.96, 0.95,0.90, and 0.78, respectively. This high precision is necessary before the commencement of a construction project since a prospective client may want to reduce injuries and consequences to the barest level. For a positive prediction of the head injury, for instance, safety managers will react to this by ensuring the appropriate use of personal protective equipment and other safety measures to mitigate the effects and occurrences of such an injury.

Also, overall f1-scores (Table 4) for the models are 0.97 (DNN), 0.97 (XGB), 0.96 (GBM), 0.91 (SVM), and 0.77 (KNN). The high precision, recall, and f1-score (> 0.75) obtained by the models show they are extremely accurate for prediction problems. However, based on the results obtained here, DNN comes out as the best performing ML algorithm, with its validation also authenticating the reliability of the selected predictors. The deep neural networks support massively scaled models with several features and millions of parameters, offering significant potential for further investigations. However, all the models except SVM and KNN are computationally efficient regarding the training times required to build them. As a result, they can conveniently be built on most machines.

4.4. Sensitivity analysis

Sensitivity analysis is a procedure to identify predictors that potentially influence the outputs of the problem. We carry out sensitivity analysis on the best three models' input parameters (neurons for the feedforward neural networks and number of trees for boosted tree models) to assess their sensitivity to fluctuations. In Table 4, we compare different configurations of models, i.e., baseline models versus models with reduced parameters (configuration A) and models with increased parameters (configuration B) using the accuracy metric. Reducing the input parameters here means splitting the baseline models' parameters into halves, while increasing input parameters means doubling their values.

For instance, we changed the "number of trees" parameter for GBM from 120 (Baseline) to 60 (Configuration A), and from 120 (Baseline) to 240 (Configuration B), respectively. Also, the number of neurons in the first hidden layer of the deep feedforward neural networks was changed appropriately for Configuration A (reduced) and Configuration B (doubled), respectively.

In Table 5, we observe that both XGB and GBM models produced consistent prediction accuracies irrespective of the number of parameters (i.e., trees or neurons) used. A slight improvement is noticeable except for DNN when parameters' values are doubled (i.e., GBM from 0.9579 to 0.9599, and XGB from 0.9642 to 0.9653) or reduced (i.e., GBM from 0.9579 to 0.9484, XGB from 0.9642 to 0.9562; DNN from 0.9673 to 0.9252). This result revealed that boosted tree methods are easier to configure and more robust. The prediction accuracy of DNN, though negligible, dropped by a fraction of 0.04 (in both configurations); that is, adding more neurons than necessary usually improves the performance on the training set but negatively impacts the performance on the test set. On the other hand, pruning many neurons will damage the performance, and performance drops are unrecoverable.

Consequently, the configurations of the models (GBM, XGB, and DNN) performed well (accuracy > 0.90). The three models showed strong generalization abilities (errors between the baseline models and different configurations are negligible), indicating their effectiveness in predicting injuries.

4.5 Model interpretability

We used three global interpretation methods from Interpretable Machine Learning (IML) [54] to explore the internal workings of DNN as the best predictive model. The IML package provides some model-agnostic interpretation methods for ML models. It has internal support for some ML packages (i.e., *mlr*, *caret*, and *KernLab*); however, we carry out a few tinkering to set up an interface with the H2O framework, the popular and state-of-the-art ML package in use today. We first created a custom function to accept a dataset of class (*data.frame*) and designate predicted values as vectors. Then we created a predictor object to hold the model, data, and class labels to be applied to subsequent functions. The three global interpretation methods used are partial dependence, measuring interactions, and surrogate model.

4.5.1 Measuring interactions

Interactions among predictors can be measured to discover how strongly they interact with each other. The IML uses the H-statistic [55] to measure the predicted outcome's dependency on the predictors' interactions. The interaction effect among predictors captured by the DNN model, depicted in Fig. 6a, is strong, with *LOC* exhibiting the strongest interaction signal while SHIFT_W exhibited the least interaction. For a two-way interaction of LOC with other predictors depicted in Fig. 6b, the *EQP_T:LOC* (location and equipment) interaction had the most substantial influence on body parts injuries, confirming findings from [22].

Table 4. Precision, recall, F1-score of models

	Recall					Precision					F1-score				
Body parts	XGB	GBM	DNN	SVM	KNN	XGB	GBM	DNN	SVM	KNN	XGB	GBM	DNN	SVM	KNN
Ankle/foot	0.98	0.97	1.00	0.96	0.90	1.00	0.97	1.00	0.98	0.97	0.99	0.96	1.00	0.97	0.93
Arm/elbow	0.98	0.96	0.93	0.95	0.83	0.95	0.95	0.97	0.96	0.98	0.96	0.95	0.95	0.95	0.90
Back/Buttocks	0.99	1.00	1.00	0.96	0.82	0.94	0.93	0.96	1.00	1.00	0.96	0.96	0.98	0.97	0.90
Chest/abdomen	1.00	0.94	0.94	1.00	0.89	1.00	1.00	1.00	0.97	0.97	1.00	0.97	0.97	0.98	0.93
Ears	1.00	1.00	1.00	1.00	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97
Eyes	1.00	1.00	1.00	1.00	0.90	1.00	0.95	0.97	0.97	0.97	1.00	0.97	0.98	0.98	0.94
Face/Shin	0.96	0.97	1.00	0.84	0.76	0.96	0.96	0.86	0.90	0.68	0.96	0.96	0.93	0.87	0.72
Groin/hip	1.00	1.00	1.00	0.98	0.91	0.98	1.00	0.98	0.98	0.98	0.99	1.00	0.99	0.98	0.95
Hand/fingers	0.91	0.91	0.93	0.82	0.45	0.86	0.89	0.98	0.73	0.56	0.88	0.90	0.95	0.77	0.50
Head	1.00	1.00	0.98	0.86	0.68	0.96	0.95	0.96	0.88	0.62	0.98	0.97	0.97	0.87	0.65
Knee/leg	0.90	0.90	0.92	0.63	0.48	0.96	0.93	0.93	0.66	0.33	0.92	0.91	0.92	0.64	0.39
Multiple parts	0.85	0.83	0.95	0.83	0.53	0.95	0.91	0.95	0.75	0.38	0.89	0.87	0.95	0.79	0.42
Neck/shoulder	0.98	0.94	0.98	0.86	0.67	0.97	0.98	0.98	0.90	0.64	0.97	0.96	0.98	0.88	0.65
No Injury	0.98	0.98	0.98	0.99	0.95	0.99	0.99	1.00	0.96	0.92	0.98	0.98	0.99	0.97	0.93
Overall	0.97	0.96	**0.97**	0.91	0.77	0.96	0.95	**0.97**	0.90	0.78	0.97	0.96	0.97	0.91	0.77

Table 5. Sensitivity analysis

Model	Baseline (B)	Configuration A (CA)	Configuration B (CB)	Error	
XGB: no of trees	120	60	240		
GBM: no of trees	130	65	260		
DNN: no of neurons	22-500-500-14	22-250-500-14	22-1000-500-14		
Prediction accuracy on test data				$\|B-CA\|$	$\|B-CB\|$
XGB	0.9642	0.9562	0.9653	0.008	0.001
GBM	0.9579	0.9484	0.9599	0.010	0.002
DNN	0.9673	0.9252	0.9262	0.042	0.041

The TASK:LOC interaction was the second strongest predictor, significantly contributing to body parts injuries. Other significant interactions contributing to injuries include *EMP_E:LOC*, *EMP_C:LOC*, *PROJ_C:LOC*, *WSLC:LOC*, and *EMP_A:LOC*. The *YEAR:LOC* interaction had the least influence on injuries to body parts.

This result suggests that the condition of site locations and work surface layout designs are essential contributors to injuries. Linemen spend most of their time outdoors on unmade roads and in restricted workspaces while undertaking construction works. Planning of workplace layout, good access to sites, and safety controls may influence good safety practices, which will reduce injuries on sites. Also, work activities carried out by linemen are considered instrumental in determining whether or not injuries will occur. For instance, the more repetitive a task is, the more likely a lineman being injured. An automated system is, therefore, recommended to automate some of these operations to reduce injuries.

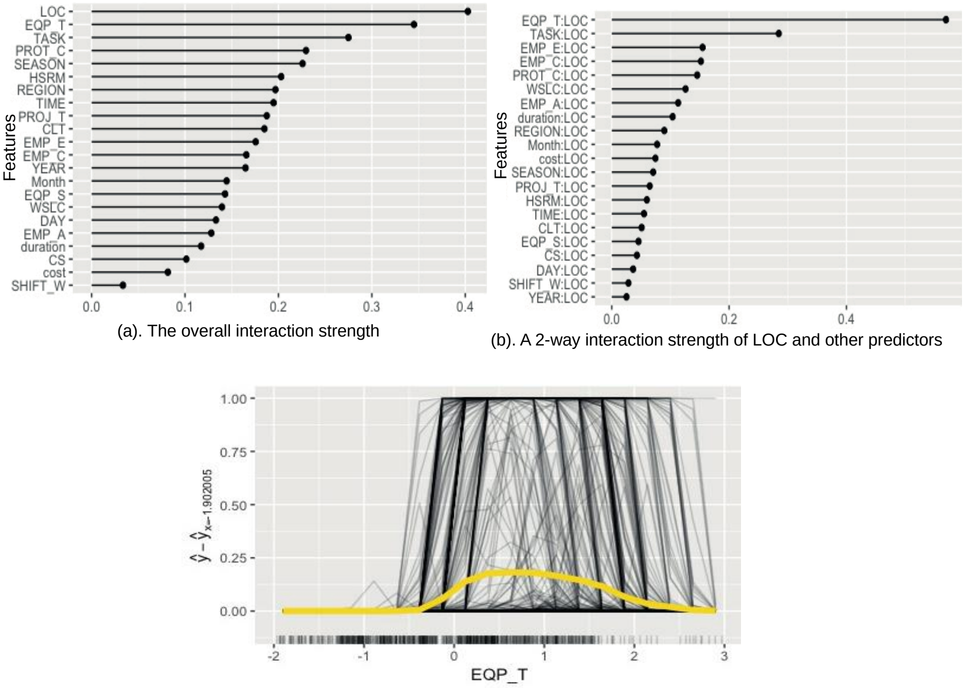
4.5.2. Partial dependence

The partial dependence plot (PDP) shows the marginal effect of one or two predictors on the predicted outcome, indicating whether the relationship between them is linear, monotonous, or complicated. A Partial class [54] was used to illustrate this dependence. We implemented the PDP and individual conditional expectation (ICE) curves following the methodology documented in the literature [55].

Fig. 6c depicts ICE curves and the PDP curve (in yellow) for comparing the marginal impact of the predictor (EQP_T) on the likelihood of an injury. The relationship captured by DNN is nonlinear, and the PDP curve (thick yellow line) represents the average prediction across all observations. This nonlinear model exhibits behavior that seems intuitively reasonable, i.e., equipment when newly acquired functions optimally and efficiently (likelihoods of causing injuries is low), but they begin to malfunction at a later stage (likelihoods of causing injuries is high). However, they begin to operate optimally again when repaired. The ability of deep neural networks to capture this behavior partially explains their superior prediction performance.

4.5.3. Surrogate models

A surrogate model enables a sophisticated model to be more interpretable by replacing the “black box” model with a simpler one (i.e., a “white box” model such as decision trees). Surrogate models help in explaining complicated deep learning models to decision-makers and helping them identify risks accurately to provide appropriate mitigation plans. A decision tree algorithm trained using a maximum tree depth of 4 was used to interpret the internal workings of the DNN model. The decision tree explains the DNN predictions with a correlation coefficient value ($R^2 = 0.73$).



(a). The overall interaction strength

(b). A 2-way interaction strength of LOC and other predictors

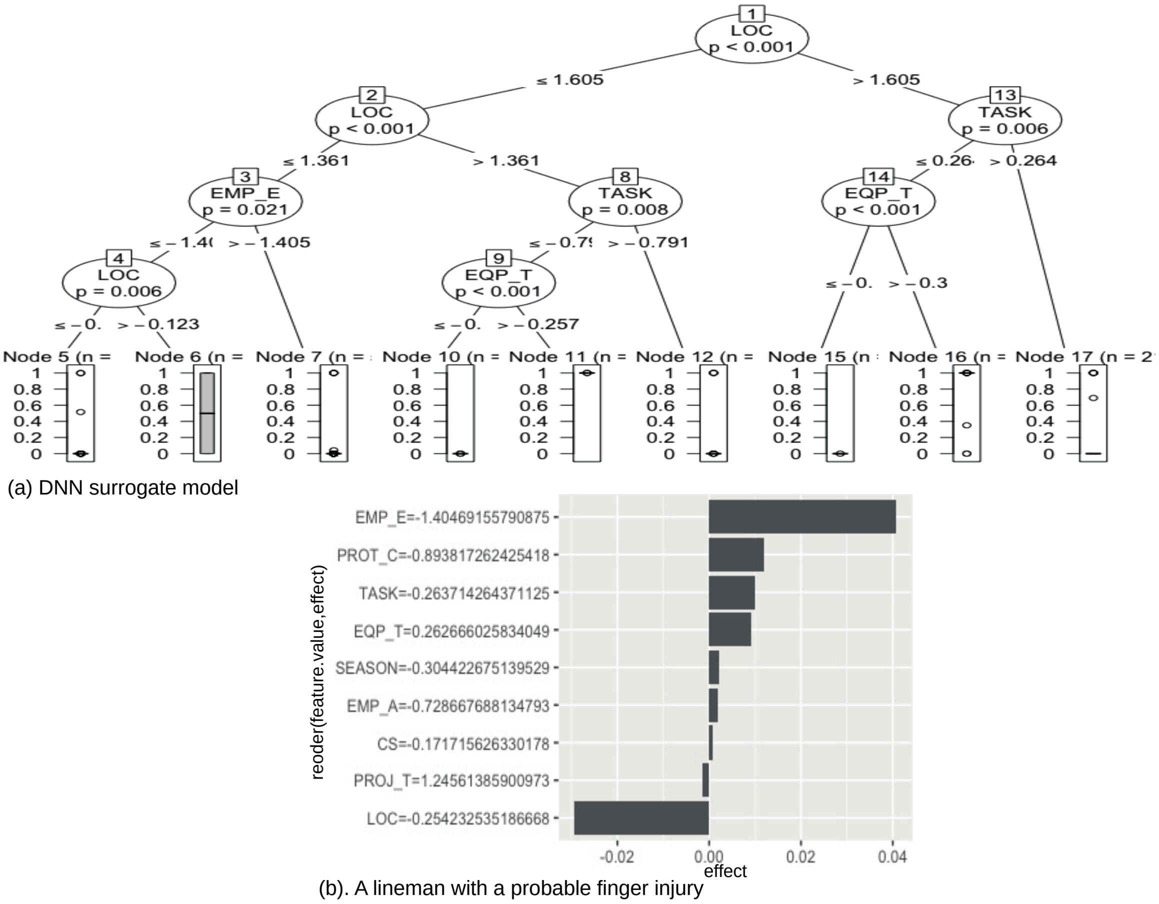
(c). Partial dependence plots for predictor EQP_T

Fig. 6. Explaining DNN with interaction strength and partial dependence plots

Fig. 7(a) illustrates some rules describing the occurrence of injuries and the associations between the predictors determined from such rules. For instance, the rule (LOC>1.605 AND TASK>0.264) means certain tasks conducted in some locations will result in a 0.62 likelihood of ankle, 0.23 likelihood of head, 0.1 likelihood of eye, and 0.05 likelihood of face/shin injuries. Also, the rule (LOC > 1.605 AND TASK<=0.264 AND EQP_T <=-0.3) implies a 0.86 probability of a lineman getting multiple injuries and a 0.14 probability of injuring the ankle for specific tasks performed with some equipment in that location. Similarly, another rule (LOC > 1.605 AND TASK <= 0.264) AND EQP_T >-0.3) implies there is a 0.93 probability of a lineman injuring the head, 0.02 injuring eyes, and 0.05 injuring multiple body parts for certain operations executed with some equipment in certain locations.

4.5.4. Local interpretation

A Local Interpretable Model-agnostic Explanations (LIME) was also implemented to provide local explanations for the deep learning models. For instance, Fig 7b fits a local deep learning model for a lineman on the likelihood of a hand injury by checking the ten most influential predictors in the model. As shown in Fig 7b, predictors such as EMP_E (an inexperienced technician), PROJ_C (a new build), TASK (i.e., wiring), and EQP_T (i.e., manual tools) have a sizeable influence on this lineman having a hand injury. However, the predictor LOC does not influence the occurrence of a hand injury for this observation. A literal interpretation might be the lineman working on or near a wire that is thought to be dead but live and mistakenly touching the wire using specific equipment. Thus, new workers or inexperienced personnel need adequate training in equipment handling and operations.



(a) DNN surrogate model

(b). A lineman with a probable finger injury

Fig. 7. Explaining DNN with a surrogate model and LIME

4.6. Implications of the study

This study has attempted to comprehensively consider current work safety situations in the power infrastructure viz-a-viz using more explanatory variables compared with other studies, developing classification models with state-of-the-art prediction techniques, and drawing on historical data to have a robust understanding of causes of injuries in power infrastructure projects.

The models developed identified key predictors that contributed to injuries and showed how their combinations could lead to human body part injuries and serve as an essential reference to preventing injuries from occurring. For instance, before embarking on a power infrastructure project, critical features related to the project (location, equipment, tasks, regions, lineman's experience, and duration) will be fed into predictive models to simulate various scenarios.

The resulting output from this study can trigger proactive safety precautions to mitigate injuries and their consequences. Therefore, administrators having this pre-knowledge will ensure that preventive and safety measures are implemented to avoid or minimize injuries. Also, the sensitivity analysis results validated the stability of the models for consistent predictions irrespective of the testing data size and the number of parameters. Again, we used more predictors for work-related non-fatal injuries than most related works, and to the best of our knowledge, this is the first time the predictive performance of deep neural networks and boosted trees are applied to predict injuries within the power infrastructure domain. Besides, the developed DNN model is sufficient, less computational, and can be adapted in related fields because it exhibits state-of-the-art features such as dropout regularization and ADADELTA optimization.

5. Conclusions

As demonstrated in this study, the application of robust ML techniques in the power infrastructure industry is timely. This industry is becoming conscious of the need to collect massive unstructured data and elicit meaningful value for decision-making. In this study, we adopted deep learning and boosted tree methods to analyze the power infrastructure incident datasets, and improve accuracies of injury prediction for safety risks management. We benchmarked the predictive ability of deep neural networks with boosted trees on testing data using appropriate metrics. The deep learning outperformed boosted trees and other conventional ML techniques with an accuracy of 0.967 and a Kappa measure of 0.964. The sensitivity analysis of the deep neural networks and boosted tree models revealed their robustness and stability to changes in data sizes and architectures. This study has implications both in academia and industrial practice, mainly the revelation of critical predictors that can result in injuries and how they can serve as crucial reference points in injury prevention. Also, the computationally efficient models developed can be adapted in related fields.

This study was limited in a way that it focuses on one construction company. Thus, the generalizability of these research findings to other companies can only be validated through additional research by collecting data from several organizations to study the organizational complexity effects on culture. We also desire to implement robust interface techniques (Generative Adverbial Networks and Convolutional Neural Networks) for near real-time video processing and prescriptive analytics to the developed deep feedforward neural networks for holistic safety management.

Acknowledgements

The authors sincerely appreciate Innovate UK (Grant Ref: 54832-413479) and EPSRC for the support.

References

[1] A. A. Garcia, I. G. Bobadilla, G. A. Figueroa, M. P. Ramirez, and J. M. Roman, “Virtual reality training system for maintenance and operation of high-voltage overhead power lines,” *Virtual Real.*, pp. 27–40, 2016.

[2] A. Pinto, I. Nunes, and R. Ribeiro, “Occupational risk assessment in construction industry – Overview and reflection,” *Saf. Sci.*, vol. 49, pp. 616–624, 2011.

[3] M. W. Chupka, R. Earle, and P. H. Fox-Penner, *Transforming America’s Power Industry: the Investment Challenge 2010–2030*. Washington, D.C: Edison Foundation, 2008.

[4] A. J. P. Tixier, M. R. Hallowell, B. Rajagopalan, and D. Bowman, “Application of machine learning to construction injury prediction,” *Autom. Constr.*, vol. 69, pp. 102–114, 2016.

[5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.

[6] T. Rivas, M. Paz, J. E. Martín, J. M. Matías, J. F. García, and J. Taboada, “Explaining and predicting workplace accidents using data-mining techniques,” *Reliab. Eng. Syst. Saf.*, vol. 96, no. 7, pp. 739–747, 2011.

[7] A. Ajayi *et al.*, “Optimised Big Data analytics for health and safety hazards prediction in power infrastructure operations,” *Saf. Sci.*, vol. 125, 2020.

[8] S. Sanchez, F. Iglesias-Rodríguez, F. Riesgo, and F. de Cos Juez, “Applying the K-nearest neighbor technique to the classification of workers according to their risk of suffering musculoskeletal disorders,” *Int. J. Ind. Ergon.*, vol. 52, pp. 92–99, 2014.

[9] M. R. D. Stackhouse and R. Stewart, “Failing to fix what is found: Risk accommodation in the oil and gas industry,” *Risk Anal.*, vol. 37, no. 1, pp. 130–146, 2017.

[10] A. Albert and M. Hallowell, “Safety risk management for electrical transmission and distribution line construction,” *Saf. Sci.*, vol. 51, no. 51, pp. 118–126, 2013.

[11] H. Lingard, M. Hallowell, R. Salas, and P. Pirzadeh, “Leading or lagging? Temporal analysis of safety indicators on a large infrastructure construction project,” *Saf. Sci.*, pp. 206–220, 2017.

[12] P. Li and K. Mao, “Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts,” *Expert Syst. Appl.*, vol. 115, pp. 512–523, 2019.

[13] B. Greenwell, B. Boehmke, and J. Cunningham, “gbm: Generalized Boosted Regression Models. R package version 2.1.5.” pp. 1–39, 2019.

[14] T. Chen *et al.*, “xgboost: Extreme Gradient Boosting. R package version 1.0.0.2.” pp. 1–57, 2020.

[15] S. Sarkar, S. Vinay, R. Raj, J. Maiti, and P. Mitra, “Application of optimized machine learning techniques for prediction of occupational accidents,” *Comput. Oper. Res.*, vol. In press, pp. 1–15, 2018.

[16] R. Sheridan, W. Wang, A. Liaw, J. Ma, and E. Gifford, “Extreme Gradient Boosting as a Method for Quantitative Structure-Activity Relationships,” *J. Chem. Inf. Model.*, vol. 56, no. 12, pp. 2353–2360, 2016.

[17] S. H. Salehi, M. J. Fatemi, K. Aśadi, S. Shoar, A. DerGhazarian, and R. Samimi, “Electrical injury in construction workers: A special focus on injury with electrical power,” *Burns*, vol. 40, no. 2, pp. 300–304, 2014.

[18] C. Moriguchi, J. Alencar, L. Miranda-Júnior, and H. Coury, “Musculoskeletal symptoms among energy distribution network linemen,” *Brazilian J. Phys. Ther.*, pp. 123–129, 2009.

[19] M. Yu, L. Sun, I. Du, and F. Wu, “Ergonomics Hazards Analysis of Linemen’s Power Line Fixing Work in China,” *Int. J. Occup. Saf. Ergon.*, vol. 15, no. 3, pp. 309–317, 2009.

[20] E. Kazan and M. A. Usmen, “Worker safety and injury severity analysis of earthmoving equipment accidents,” *J. Safety Res.*, vol. 65, pp. 73–81, 2018.

[21] C. Q. . Poh, C. U. Ubeynarayana, and Y. M. Goh, “Safety leading indicators for construction sites: A machine learning approach,” *Autom. Constr.*, vol. 93, pp. 375–386, 2018.

[22] C. Cheng, S. Leu, Y. Cheng, T. Wu, and C. Lin, “Applying data mining techniques to explore factors contributing to occupational injuries in Taiwan’s construction industry,” *Accid. Anal. Prev.*, vol. 48, pp. 214–222, 2012.

[23] S. K. Palei and S. K. Das, “Logistic regression model for prediction of roof fall risks in bord and pillar workings in coal mines: An approach,” *Saf. Sci.*, vol. 47, no. 1, pp. 89–96, 2009.

[24] Z. Liu, Q. Pan, and J. Dezert, “A new belief-based K-nearest neighbor classification method,” *Pattern Recognit.*, vol. 46, no. 3, pp. 834–844, 2013.

[25] F. E. Ciarapica and G. Giacchetta, “Classification and prediction of occupational injury risk using soft computing techniques : An Italian study,” *Saf. Sci.*, vol. 47, no. 1, pp. 36–49, 2009.

[26] W. Yi, A. P. C. Chan, X. Wang, and J. Wang, “Development of an early-warning system for site work in hot and humid environments: A case study,” *Autom. Constr.*, vol. 62, pp. 101–113, 2016.

[27] N. Nenonen, “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.*, vol. 44, no. 2, pp. 215–224, 2013.

[28] G. Mistikoglu, I. H. Gerek, E. Erdis, P. E. M. Usmen, H. Cakan, and E. E. Kazan, “Decision tree analysis of construction fall accidents involving roofers,” *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2256–2263, 2015.

[29] Y. M. Goh and D. Chua, “Neural network analysis of construction safety management systems: a case study in Singapore,” *Constr. Manag. Econ.*, vol. 31, no. 5, pp. 460–470, 2013.

[30] N. W. Chi, K. Y. Lin, and S. H. Hsieh, “Using ontology-based text classification to assist Job Hazard Analysis,” *Adv. Eng. Informatics*, vol. 28, no. 4, pp. 381–394, 2014.

[31] C. Shang, F. Yang, D. Huang, and W. Lyu, “Data-driven soft sensor development based on deep learning technique,” *J. Process Control*, vol. 24, no. 3, pp. 223–233, 2014.

[32] Y. Bengio, O. Delalleau, and C. Simard, “Decision Trees do not Generalize to New Variations,” *Comput. Intell.*, vol. 26, no. 4, pp. 449–467, 2010.

[33] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Trans. Audio. Speech. Lang.*

Processing, vol. 20, no. 1, pp. 30–42, 2012.

[34] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 35, no. 8, pp. 1798–1828, 2013.

[35] D. Thammasiri, D. Delen, P. Meesad, and N. Kasap, “A critical assessment of imbalanced class distribution problem: the case of predicting freshmen studen,” *Expert Syst. Appl.*, pp. 321–330, 2014.

[36] Z.-H. Liu, Xu-Ying; Wu, Jianxin; Zhou, “Exploratory Undersampling for Class-Imbalance Learning,” *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 39, no. 2, pp. 539–550, 2009.

[37] A. Jain, K. Nandakumar, and A. Ross, “Score normalization in multimodal biometric systems,” *Pattern Recognit.*, pp. 2270–2285, 2005.

[38] T. Gedeon, “Data mining of inputs: analysing magnitude and functional measures,” *Int. J. Neural Syst.*, pp. 209–218, 1997.

[39] M. Törner and A. Pousette, “Safety in construction - a comprehensive description of the characteristics of high safety standards in construction work, from the combined perspective of supervisors and experienced workers,” *J. Safety Res.*, vol. 40, no. 6, pp. 399–409, 2009.

[40] H.-T. Liu and Y. Tsai, “A fuzzy risk assessment approach for occupational hazards in the construction industry,” *Saf. Sci.*, vol. 50, no. 4, pp. 1067–1078, 2012.

[41] C. W. Liao and Y. H. Perng, “Data mining for occupational injuries in the Taiwan construction industry,” *Saf. Sci.*, vol. 46, pp. 1091–102, 2008.

[42] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer-Verlag, 2001.

[43] A. Candel and E. LeDell, *Deep learning with H2O*, 6th ed. Mountain View, CA, USA: H2O.ai, Inc, 2016.

[44] J. Schmidhuber, “Deep learning in neural networks : An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.

[45] L. Li, Y. Li, Y. Qin, J. Chen, L. Wang, and D. Yi, “Adaptive stochastic gradient boosting tree with composite criterion,” *J. Stat. Comput. Simul.*, vol. 86, no. 10, pp. 1901–1911, 2016.

[46] T. Chen and C. Guestrin, “XGBoost,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, 2016, pp. 785–794.

[47] S. Zeng, J. Gou, and L. Deng, “An antinoise sparse representation method for robust face recognition via joint l1 and l2 regularization,” *Expert Syst. Appl.*, vol. 82, pp. 1–9, 2017.

[48] Y. Bengio, “Practical Recommendations for Gradient-Based Training of Deep Architectures,” in *Lecture Notes in Computer Science*, G. Montavon, G. B. Orr, and K. Müller, Eds. Springer Berlin Heidelberg, 2012, pp. 437–478.

[49] M. D. Zeiler, “ADADELTA: an adaptive learning rate method,” *arXiv Prepr.*, vol. arXiv:1212, 2012.

[50] J. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann. Stat.*, 2001.

[51] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[52] J. W. Hinze and J. Teizer, “Visibility-related fatalities related to construction equipment,” *Saf. Sci.*, vol. 49, no. 5, pp. 709–718, 2011.

[53] W. Fang *et al.*, “A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network,” *Adv. Eng. Informatics*, vol. 39, pp. 170–177, 2019.

[54] C. Molnar, G. Casalicchio, and and B. Bischl, “iml: An R package for Interpretable Machine Learning,” *J. Open Source Softw.*, vol. 3, no. 27, pp. 1–2, 2018.

[55] J. Friedman and B. Popescu, “Predictive learning via rule ensembles,” *Ann. Appl. Stat.*, vol. 2, no. 3, pp. 916–954, 2008.