Automation in Construction 116 (2020) 103202

Contents lists available at ScienceDirect

Automation in Construction

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

Maintenance intervention predictions using entity-embedding neural networks

Zaharah Allah Bukhsh[a,*], Irina Stipanovic[a], Aaqib Saeed[b], Andre G. Doree[a]

[a] Department of Construction Management and Engineering, University of Twente, Enschede, the Netherlands
[b] Department of Mathematics and Computer Science, Technical University of Eindhoven, Eindhoven, the Netherlands

ARTICLE INFO

Keywords:
Maintenance decisions
Bridges
Maintenance prediction
Machine learning
Deep neural networks
Decision-support
Multi-task learning
Entity embedding

ABSTRACT

Data-driven decision support can substantially aid in smart and efficient maintenance planning of road bridges. However, many infrastructure managers primary rely on information obtained during visual inspection to subjectively decide on the follow-up maintenance actions. The subjective approach is likely to lack the appropriate use of inspection data and does not promise cost-effective maintenance plans. In this paper, we show that the historical and operational data, readily available at the agencies, is of vital importance and can be used effectively for the recommendations of maintenance advises for bridges. This is achieved by developing a machine learning system that is trained on the past asset management data and provide support to the decision-makers in the condition assessment, risk analysis, and maintenance planning tasks. We have evaluated several traditional learning algorithms as well as the deep neural networks with entity embedding to find the optimal predictive models in terms of predictive capability. Additionally, we have explored the multi-task learning framework that has a shared representation of related prediction tasks to develop a powerful unified model. The analysis of results shows that a unified multi-task learning model performed best for the considered problems followed by task-specific neural networks with entity embedding and class weights. The results of models are further evaluated by instance-level explanations, which provide insights about essential features and explain the importance of data attributes for a particular task.

1. Introduction

Functional and serviceable transport infrastructure presents one of the essential predispositions for the economic growth of a country. Among other infrastructure objects, bridges represent a vital link in any roadway network. They provide the crossings at critical locations, reduce the travel times, and maintain the traffic flow [1]. Under limited financial resources [2], agencies have to take prudent investment and maintenance planning decisions to improve the availability of the bridges, to minimize their life-cycle cost, and to maximize the return on investments. To handle the amount of information required to achieve these objectives, many infrastructure owners use the computerized management systems to manage and process relevant data and to support the decision-making processes [3].

Many agencies have developed the Bridge Management System (BMS) tailored to their specific management needs. Mirzaei et al. [3] provides an overview of BMS being used in sixteen countries. Similarly, Markow and Hyman [4] explore how assets owners use the capabilities of BMS to get support in the decision-making of bridges management programs. Many BMS primarily rely on information obtained during the visual inspection process to decide on the follow-up maintenance actions [5]. These systems prompt inspectors to describe the physical state of the structure, which is quantified based on condition score card [6]. The traditional quantification methods from visual inspection to condition rating rely on a subjective process, with a main assumption that a bridge inspector is experienced and trained personnel and has detailed knowledge of the structure [7]. Since there is often no systematic procedure to record experts' preferences, their comprehension of structures, and related performance objectives, the maintenance decisions become difficult to follow, justify, and replicate in the future.

Several useful reliability assessment and maintenance optimization models have been proposed in the literature [8–12]. However, frequently, the reliability assessment models are not part of BMS; therefore, not every bridge gets an opportunity to have a detailed future performance profile. Likewise, the maintenance optimization models introduce complex mathematical heuristics to formulate and solve the problem. Therefore, the agencies still prefer to use traditional methods based on subjective ranking and preferences of domain experts for the

* Corresponding author at: Drienerlolaan 5, 7522 NB Enschede, the Netherlands.
E-mail address: z.allahbukhsh@utwente.nl (Z. Allah Bukhsh).

https://doi.org/10.1016/j.autcon.2020.103202
Received 17 September 2019; Received in revised form 2 March 2020; Accepted 23 March 2020
Available online 20 May 2020

maintenance decision-making [13–15]. Multiple efforts have been reported in the literature to improve the functionalities of BMS for the decision-making tasks [7,16]. However, the focus has mainly been on extending BMS capabilities to support in long term maintenance planning, and the whole life cycle costing of assets.

The theoretical progress and agency's practices highlighted three key challenges in the context of decision-support for maintenance planning. Firstly, a little attention is paid to investigate the solution that could improve the subjective assessment procedures from visual inspection of assets towards maintenance planning. Secondly, the historical data collected during the past visual inspections are not used for the decision-making process due to data access and analysis limitations [17]. Thirdly, the condition and maintenance optimization models do not scale up to the network-level, and they provide limited support in detailed condition assessment and maintenance planning. To tackle these challenges, we developed a machine learning (ML) system that is trained on historical data and can provide recommendations to the decision-makers on the condition assessment and maintenance planning tasks. ML techniques can solve classification and regression problems by inferring the patterns and rules from the data. The learned model can be used to predict a discrete class (such as condition state) or continuous target class (such as displacements), respectively. The capability of learning discriminative features directly from the data enables the development of systems that can either automate the decisions or provide recommendations to the human decision-maker. The terms *classification* and *predictions* are used interchangeably in this paper. It is because *classification* is a machine learning problem, which entails the *prediction* of a discrete class label. In other words, the *predictive* models are developed to solve the *classification* problem.

We used a large dataset of concrete bridges from the road agency to illustrate the development methodology. The dataset is collected over the years as a result of the Inspection to Maintenance Advice (IMA) process, which is implemented in a BMS. The IMA process collects the data of visual inspection, where a decision-maker assesses the data and decides on the condition state, risk level and recommends maintenance advice on the bases of his/her technical knowledge and judgments. The objective of this study is to develop classification models that can provide support in the subjective assessment procedure of bridge maintenance planning. This work deals with three prediction tasks, namely, assessment of *condition state*, analysis of *risk level*, and recommendation of *maintenance advice*, all by using the damage details noted during visual inspection activity.

The main contributions of this paper are following:

- We have developed several machine learning models and deep neural networks that can learn from the visual inspection data of the bridges. These models are introduced as a tool to support asset-owners in the subjective decision-making process.
- We have presented a generic development methodology that utilizes only the existing data generated from an in-use business process of a transport agency. This results in predictive models that are aligned with the current decision-making practices of the agency. Unlike other studies, this study does not perform additional data collection.
- This study is unique in comparing and applying logistic regression, tree-based models, neural networks with entity embedding and multi-task learning framework to find the best performing predictive model for bridge maintenance planning.
- We also provided the instance-level interpretability to explain the results of the optimal model for each task. The interpretability of the models highlights the important features and explains how a model makes certain predictions.

The paper is structured as follows: Section 2 presents an overview of the studies that utilize the ML techniques for transport infrastructure maintenance. The problem domain and the detailed data description are discussed in Section 3. Section 4 provides an overview of the methodology by highlighting the learning algorithms, the neural network's architectural details, and the evaluation strategy. The details of the experiments and results for each prediction tasks are provided in Section 5. The interpretability of the models' results is explained in Section 6. The key remarks and general observation are provided in Section 7. Section 8 highlights the major outcome of this work and provides a future research agenda.

2. Related work

Machine learning (ML) techniques have achieved significant success in many industries ranging from health-care, finance, manufacturing, marketing, transport, and agriculture. Due to advancements in communication and sensor technologies, machine learning algorithms are increasingly being adopted for the management of economic infrastructures, including transportation [18,19], energy [20–22], water management [23,24] and smart city services [25,26]. In this section, we discuss the studies that apply the ML techniques for the management of road and railway structures.

Masino et al. [27] proposed an infrastructure monitoring system based on vehicle sensors and supervised learning algorithms in order to estimate the road quality. Similarly, Souza et al. [28] introduced a low-cost system to evaluate the pavement condition by using the vibration readings from the accelerometer sensor of smartphones. Morales et al. [29] proposed a methodology to automate the prediction of maintenance interventions for the road pavements using the operational and historical maintenance data. From the railway domain, few notable studies are failure prediction models using heterogeneous data from multiple-detectors systems [30], predictive models to detect metro door failure [31], recurrent neural networks to identify and capture the failures in railway track circuits [32], finding and localizing damages in railway bridges [33], assessment of remaining useful lifetime of an electrical power switch [34], and maintenance need and type prediction for switches and crossings [35].

It can be noted that the majority of these studies employ additional data collection means to continuously monitor the assets for predictive modeling. Though useful for experimentation, it is expensive and impractical to mount monitoring devices on multiple assets across the network and continuously collect the data for a longer period of time [32]. Additionally, these studies do not address the topic of interpretability to explain the decision logic of the models. Therefore, the focus of this paper is on utilizing only historical data and proposing a methodology that can be implemented within a transport agency for the decision-making of bridge maintenance planning. Furthermore, special attention is given towards the model interpretability in order to avoid developing black-box models by elaborating on the results of the model at an instance.

3. Problem domain and data description

A road agency has shared a large dataset of concrete bridges to analyze the applicability of machine learning approaches for providing decision-support in the assessment of condition states, risk levels, and maintenance actions. Here the data and name of the agency are anonymized owing to the confidentiality agreement.

The agency uses a customized BMS to store inventory, condition states, risk profiles, and maintenance plans of road bridges. In total, the highway network consists of approximately 3800 bridges. All the civil structures (physical objects) are introduced with standard decomposition to support the network-oriented asset management approach [36].

An example of the decomposition of a road network is presented in Table 1. The focus of this study is on the *object* and sub-levels specifically for the bridges, as depicted with bold text. Depending on the structural details, a bridge consists of several *elements* and *components*. It is important to note that not all the given elements of the bridge are equally important in the overall structural integrity of a bridge. A

Table 1
Example of decomposition of road network [36].

Level	Example
Network	Highway network
Sub network	Ring road system
Network branch	Highway between interchanges
Object	Bridge, tunnel, road section
Element	Superstructure, abutment, bearing, pavement
Component	Top layer, seal of joints

weighted average method, conducted to elicit the relative importance of bridge components, reveals the superstructure, bearing, abutment, and joints as the most relevant elements for structural performance of the bridge (see Table 3 of Allah Bukhsh et al. [37]). Therefore, the problem domain of this study covers the predictive modeling of bridges having superstructure, bearing, abutment, and joints as the main elements. The amount of available data also motivated the selection of the named elements. However, the authors do acknowledge the importance of foundation as one of the most critical bridge elements. In our case, bridges are crossing over still water (canals) and are not exposed to the risk of scour failure. In many countries, due to climate change impacts and extreme rainfall events, bridge foundations may be exposed to high risk of scour [38]. Therefore special attention should be given to the foundation condition assessment and inclusion of this data in the analysis [39].

3.1. The process of Inspection to Maintenance Advice (IMA)

Inspection is an integral tool for the infrastructure asset management. The inspection framework of the considered road agency consists of three types of inspection, namely routine, general and principal inspections. Routine and general inspections are aimed at the detection of unexpected failures. The principal inspection is targeted towards prognosis of future maintenance need of the infrastructure.

A detailed process of the BMS from inspection of an asset to the maintenance advice is outlined in Fig. 1. The details of inspection are recorded at the *element* level, whereas any noted damages, their cause, type of damage, and its extent are noted at the *component* level. Afterward, the *condition score of a bridge component is quantified* on a standard scorecard based on subjective analysis, quantitative standards, and service level agreements. Next, the desk study is performed in which the noted damages and condition states are accessed for their probability of failures *to quantify the level of risks*. The noted risk on the element of the bridge is controlled by taking certain maintenance measures. The asset owners and inspection managers issue *maintenance advice* from a standard list to trigger the maintenance actions. However, the process from Inspection to Maintenance Advice is subjective, where due to risk consideration, a direct link between damage, condition, and risk level may not be established. The asset owners have to conduct many interviews and consult quantitative and quantitative standards along with the performance requirements to support the decision aspects of the IMA process [40].

Essentially, IMA follows the risk-based inspection procedure, where the performance of the structure is the main focus. This implies that even when a component has significant damage and poor condition state, but with no impact on its performance the risk is regarded as negligible [40]. This procedure ensures that the maintenance actions are not driven by condition scores only; instead the estimated risk profiles and the future maintenance plans are taken into account. The IMA process is an initial step in a holistic asset management approach followed by the agency. An interested reader may refer to [36,41] for a broader understanding of assets management and life cycle costing method of the agency. The output from the IMA process is used for the maintenance planning based on reliability, availability, maintainability and safety aspects. The optimal planning of maintenance of the assets is out of the scope of this study, though the aforementioned approach can be found in [37].

In this paper, we aim to develop classification models that could learn only from historical data of the IMA process to assist in the decision-making of condition assessment, risk level, and maintenance advice. Fig. 1 depicts the decision aspects of the IMA process with a diamond shape that will be supported by the ML classification models. Additionally, for the development of predictive models, we utilized the basic details of a bridge such as age, route, type, material, and the noted damage as depicted with *rectangle shape* in Fig. 1. Further discussion on

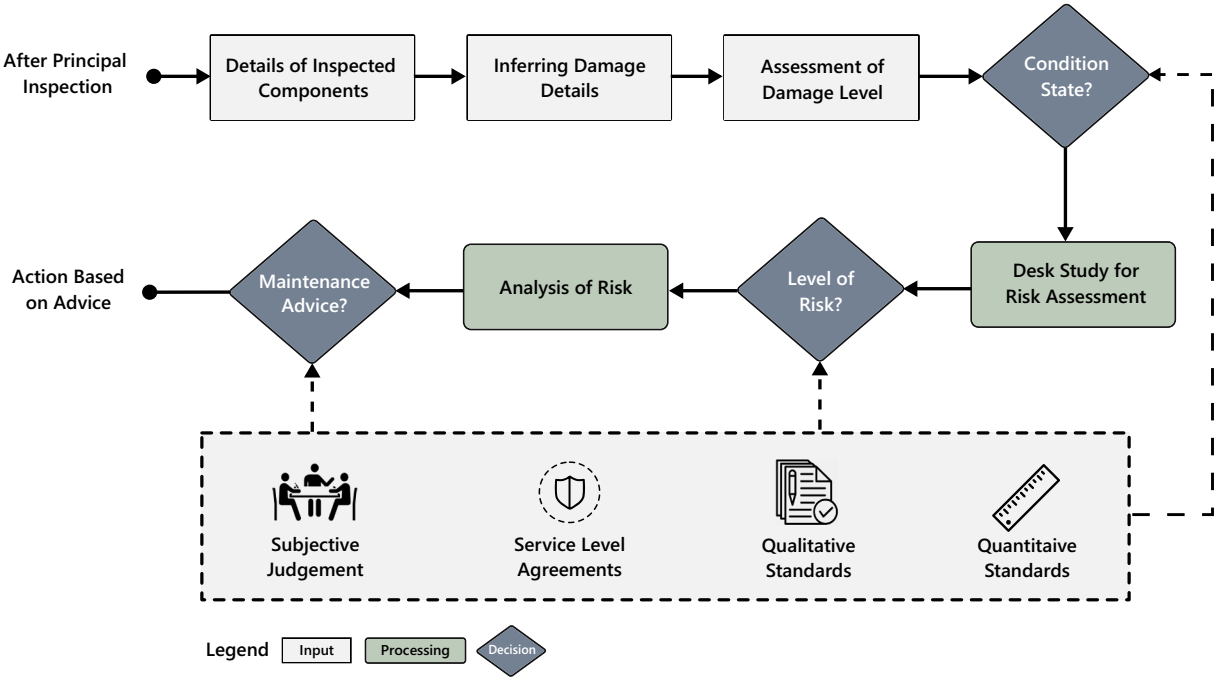


Fig. 1. Process of Inspection to Maintenance Advice (IMA).

Table 2
Selected features set from IMA dataset.

No.	Feature name	Type
1	Bridge-code	Discrete
2	Element-code	Discrete
3	Component-code	Discrete
4	Bridge-material	Categorical
5	Segment-material	Categorical
6	Element-material	Categorical
7	Component-material	Categorical
8	Bridge-nature	Categorical
9	Bridge-Age	Continuous
10	Bridge-route	Categorical
11	Element-name	Categorical
12	Component-name	Categorical
13	Inspection-point	Categorical
14	Inspection-detail	Categorical
15	Temperature-insp	Continuous
16	Weather-insp	Categorical
17	Damage-category	Categorical
18	Damage-cause	Categorical
19	Damage-level	Categorical
20	Damage-type	Categorical

the used data and its characteristics are presented in the following sections.

3.2. Data acquisition from BMS

The data generated from the IMA process is used for the development of classification models. BMS stores all the relevant data in SQL relational database system. Since the different data are recorded based on the decomposition of the road network (as shown in Table 1), we have to execute several SQL queries to obtain all the required data. In the following, a brief detail of the acquired data is provided.

- *Bridge inventory*: It presents the necessary details of the bridges, including their location, construction year, route, and connection to a network branch. Besides, the bridge inventory data also provides the data of sub-levels of a bridge object in the form of related elements and components.
- *Inspection data*: The principal inspection is performed every six years for each element of the bridge. The inspection data file provides the details of the principal inspections conducted from 2007 to 2017. It constitutes features like inspection year, element code, inspection type, inspection location, and temperature on the day of inspection.
- *Damage data*: During the inspection of the elements, the damages are also inspected and recorded at the component level. The damage data file presents component code, damage types, its possible cause, a detailed description, and intensity of the damage. The damage details help in the assessment of the physical state of the element.
- *Risk data*: With the inspection and damage data, the risk of bridge elements is assessed during a desk study. The risk data file outlines the records of all noted risks on elements, their analysis, the risk status, and the risk type. Furthermore, an estimation of the severity of the risk is also noted. To eliminate the observed risk on bridge elements, the asset and inspector manager determine the appropriate maintenance advice.

The aforementioned data sources are interconnected with a unique object identifier (code) for element, component, inspection, damage and risk details. Since, a SQL database consists of a collection of tables, these unique identifiers have enabled us to execute *join operations* and retrieve all the relevant data for each component. The obtained datasets underwent an extensive filtering and cleaning process to extract only those data instances and features that are relevant for the development of classification models.

3.3. Features engineering

Feature engineering is one of the most crucial steps in machine learning (ML) model development pipeline. The feature engineering task constitutes of preparing data for learning algorithms through extracting useful features from the given data, combining similar features, and eliminating the least relevant features. The quality and quantity of features play an intrinsic role in the predictive ability of the model. Feature engineering mainly takes into account the domain knowledge of experts who decide about the relevance of features for the dependent variable (i.e., class label).

We performed the feature engineering task on the data collected from the IMA process during the period from 2007 to 2017. Guided by several interactive sessions with experts, we eliminated duplicated features such as the condition codes and their explanations and other irrelevant features, e.g., the location coordinates, the dimensional properties, and unique identifiers related to the inspection activities. We also eliminated those instances for the bridges that were constructed before the year 1900, as they follow special maintenance procedures and have a lot of missing data. Additionally, we eliminated all the data instances that do not have any noted damages, relevant risk details, or maintenance advice. Without the specific damage details, the condition assessment is purely subjective process with no available data for the ML model development.

The decision-makers and experts of BMS also facilitated in determining the relevance of features for the classification tasks. The irrelevant features such as coordinates, bridge name, descriptions are eliminated, whereas other features like bridge-age and bridge-route are elicited from data. The exhaustive feature selection procedure reduced the total number of features (data columns) from 69 to 20, excluding the class labels. These final set of twenty features are referred as *selected features* since they are diligently chosen by experts. The tasks of condition state, risk level, and maintenance advice prediction are sequential, as depicted in Fig. 1, which means the output of one task may be used for the prediction of the other task. However, for the sake of robust classifier and to avoid data leakage problem [42], all prediction tasks are trained on the same set of features data (see Table 2), where the output of one model is not used as a feature for learning the other task.

Table 2 provides the selected features set along with their data type. *Bridge, element and component code* express a tree-like data structure to present the decomposition of a bridge in the dataset as shown in Table 1. The example of materials of a bridge and its respective elements include concrete, steel, asphalt, rubber, etc. Further details related to inspection activity are also included in terms of its specific location over the bridge, description of notable aspects, temperature, and weather during the inspection activity. Additionally, damage details in terms of its category (e.g., normal aging, construction error, etc.), its cause, type, and intensity are also considered for predictive modeling. Further insights on causes of damages are provided in the following Section.

The selected features are then pre-processed based on their data types. The features with categorical types are assigned with representative numerical codes while considering their ordinal properties. For example, *bridge nature* as wet and dry are numerically encoded as 1 and 2, respectively. For the continuous features, we performed feature scaling depending on the requirements of the used algorithm, as discussed in Section 4. For instance, tree-based algorithms such as decision tree, random forest are invariant to feature scales [43], whereas the neural network demands the data normalization. For a neural network, the z-score data normalization is applied to the numeric features using the following equation:

$$z = \frac{x - \mu}{\sigma}$$

where μ is the mean (average), and σ is the standard deviation from the mean. The standardization normalizes the numeric feature values

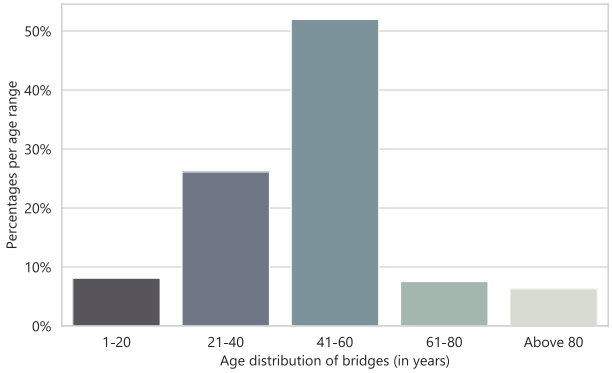


Fig. 2. Age range of the bridges in years.

around 0 with a standard deviation of 1. Z-score normalization coverts all features into a single scale, which enables comparison among them. For textual features, we initially calculated the term frequency-inverse document frequency (tf-idf) [44]. However, during the earlier exploration, we noted that the text features do not contribute towards the models' performance. Therefore, these attributes were removed from further analysis.

3.4. Visual analytics

This section provides some useful insights about the IMA dataset. The visual analytics helps in interpreting and analyzing the characteristics and distribution of data. Fig. 2 presents the age distribution of 2960 bridges that are part of the IMA dataset. Approximately 75% of all bridges are between the age of 21 to 60 years, whereas less than 10% of bridges are within 1 to 20 years of age.

The principal inspection of the bridges leads to the identification of multiple damages. To provide an overview of the noted damages, Fig. 3 present 15 most occurring damage causes. Aging is one of the most frequent causes of bridge damage. It can be further verified by the age distribution of the bridges, where at least 50% of them are older than 40 years.

The supervised learning algorithms require the labeled data for model training and testing purposes. In our case, the condition state, risk level, and maintenance advice are the class labels for which the distinct models are developed. The predictive models learn well (i.e., generalize to unseen data instances) when each class has at least one-tenth representation in the overall dataset [45]. A balanced dataset has an equal representation of all classes. However, in the case of an imbalanced dataset, the model tends to be biased towards the class having major representation, thus performing poorly for the minority classes. We present the frequency distribution and the percentage distribution for all three classification problems in Figs. 4, 5 and 6. Besides, these figures also introduce the class labels of predictive tasks, where the meaning of the labels is self-explanatory.

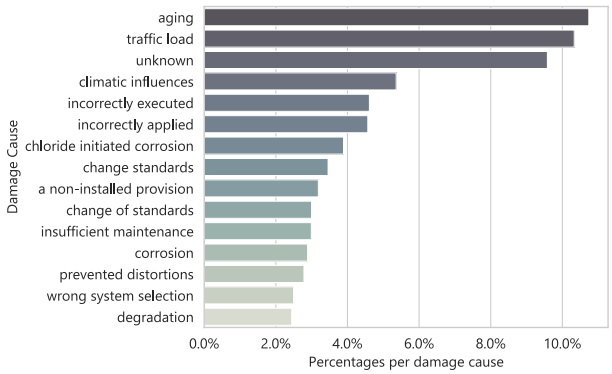


Fig. 3. Most frequently identified damages causes on elements and components of bridges.

The visual analysis presented in Figs. 4a, 5a, 6a reveals that the class distributions of condition state, risk level and maintenance advice is highly imbalanced. The *good condition state* class has more than 40,000 instances and represents more than 50% of the overall dataset as shown in Fig. 4b. The risk level classes have an even higher imbalance, where the majority class (i.e., *limited*) represents 64% of the overall dataset (see Fig. 5b). Likewise, the maintenance advice classes are also imbalanced, where *maintenance* class has 14,000 instances, which covers approximately 59% of all the data, as depicted in Fig. 6b.

In addition to using the complete (imbalanced) dataset, we also performed random under-sampling of the majority classes iteratively to determine the optimal under-sampling ratio that will improve the learning ability of the predictive models. In random sampling, each data point has an equal probability of selection, when the data instances are independent and identically distributed [45]. For condition state, risk-level, and maintenance advice prediction, a majority class is under-sampled, where only 35%–40% of its instances are randomly selected. Though the under-sampling does not balance the dataset, it improves the representation of the minority classes to a certain extent. To put this in perspective, Figs. 4c, 5c and 6c provides the under-sampled class distributions. It is essential to note that even though each element has a certain condition state, not every element has an associated risk. Hence, the number of data instances available for developing condition state models is higher than the other classification tasks.

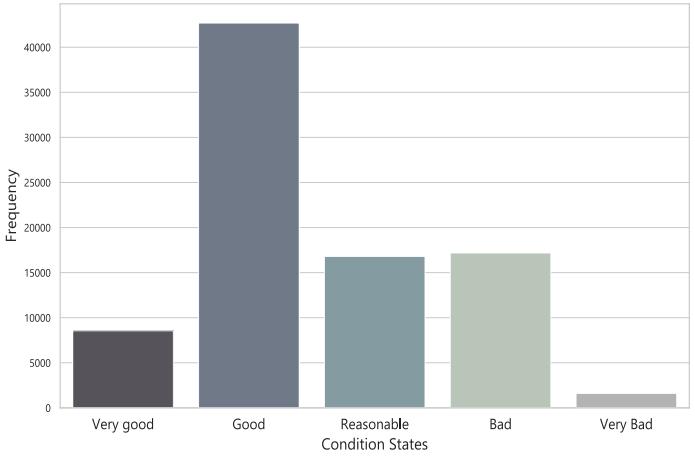
4. Methodology for prediction of maintenance related tasks

Machine Learning (ML) is a scientific study of algorithms that can extract useful patterns from the raw data in order to facilitate data-driven decision-making. The ML techniques have enabled computers to tackle complex problems using real-world data. In supervised learning, a ML model learns from the labeled training data to find the relationship between x features and y target class. A well-trained model $\widehat{f}$ must be able to make accurate prediction $\hat{y}$, given unseen future data instance $\bar{x}$. Depending on the specific learning problem, there are number of algorithms to elicit $\widehat{f}$ from the dataset [46]. The choice of an optimal algorithm depends on the target output and the size and format of the available dataset. According to the *No free lunch* theorem, no single algorithm is significantly superior to others [46]. In practice, we have to try a handful of different algorithms to train, evaluate, and select the best performing model. This section presents the overall methodology to develop accurate predictive models for the IMA dataset in order to support the subjective decision-making process of bridge maintenance planning.

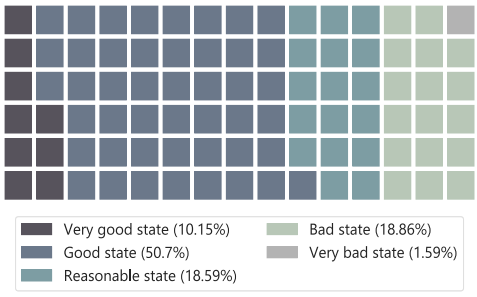
The data generated from the IMA process is annotated and has a structured nature. We selected supervised algorithms from traditional machine learning and from deep learning paradigm to find the best performing model for the prediction of condition state, risk level, and maintenance advice. In the following sections, we first briefly introduce the ML algorithms that are used for the development of predictive models. Next, we present the deep learning paradigm and also motivate our choice of utilizing neural networks for the structured dataset. The detailed explanation of each algorithm is out of the scope of this study; instead, an interested reader may refer to Trevor et al. [47]. Finally, we discuss the various evaluation approaches and performance measures that are applied to gauge the predictive ability of the developed models.

4.1. Machine learning techniques

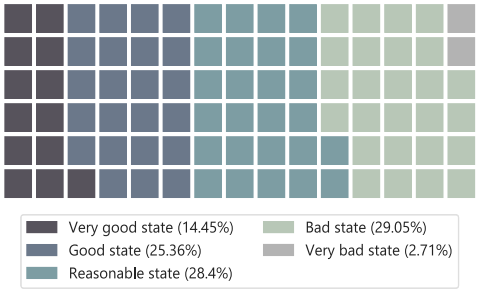
Among the several ML techniques, the utilized algorithms include logistic regression, decision tree, random forest, and gradient boosting trees. The logistic regression algorithm is selected to establish the baseline performance. The tree-based algorithms consisting of decision-



(a) Frequency distribution of condition state classes.



(b) Complete dataset.



(c) Undersampled dataset.

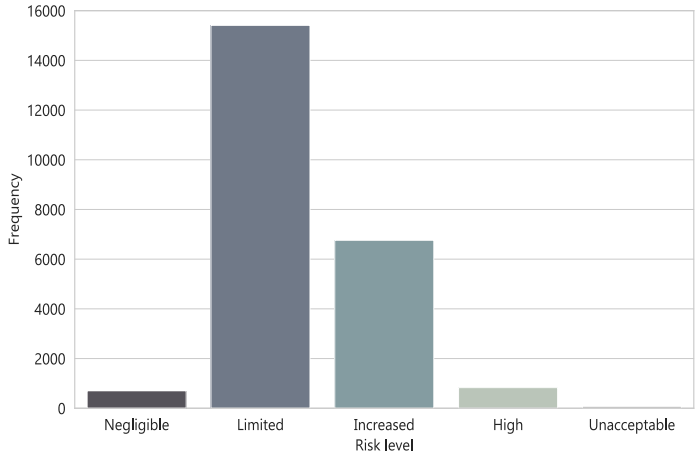
Fig. 4. Representation of condition state classes in the overall data set.

tree, random forest, and gradient boosting trees are chosen because of their proven prediction performance on structured datasets in many industry challenges and academic literature [48,49]. This section briefly introduces the ML techniques that will be applied to develop predictive models. Furthermore, we explain the details of the development and hyper-parameters tuning of each model.

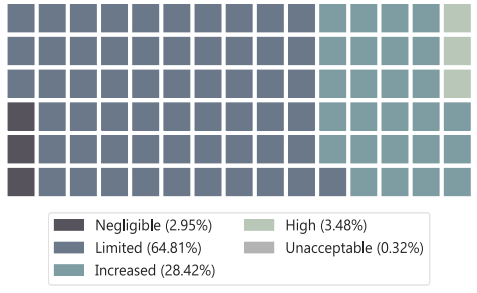
Logistic regression is a classification algorithm that performs well for the linearly separable classes. It takes a linear combination of weights and values, maps them to real-valued number, and outputs the predicted probabilities. The resulting probability presents a likelihood that a particular sample belongs to a specific class. By applying a binary threshold function, the discrete classification can also be obtained from the predicted probability.

Decision Tree works on a simple strategy of divide-and-conquer by employing the recursive partitioning of the data. The key idea is to split the dataset number of times, where the resulting sets are homogeneous and belong to the same target class. The algorithm applies the top-down greedy search to determine the best split of nodes until the maximum allowable length of a decision tree is reached, and the terminal nodes are the target classes [50].

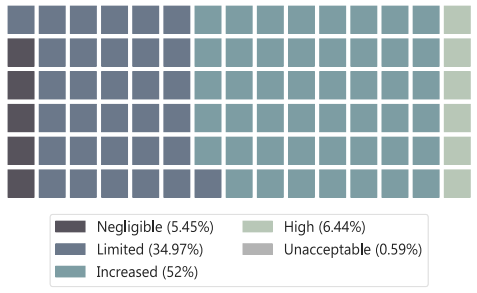
Random Forest is an ensemble approach. Unlike the decision tree that builds a single tree for an entire dataset, random forest randomly selects the instances and features of data to construct multiple trees in a parallel fashion. The central idea of random forest is to average the



(a) Frequency distribution of risk level classes.

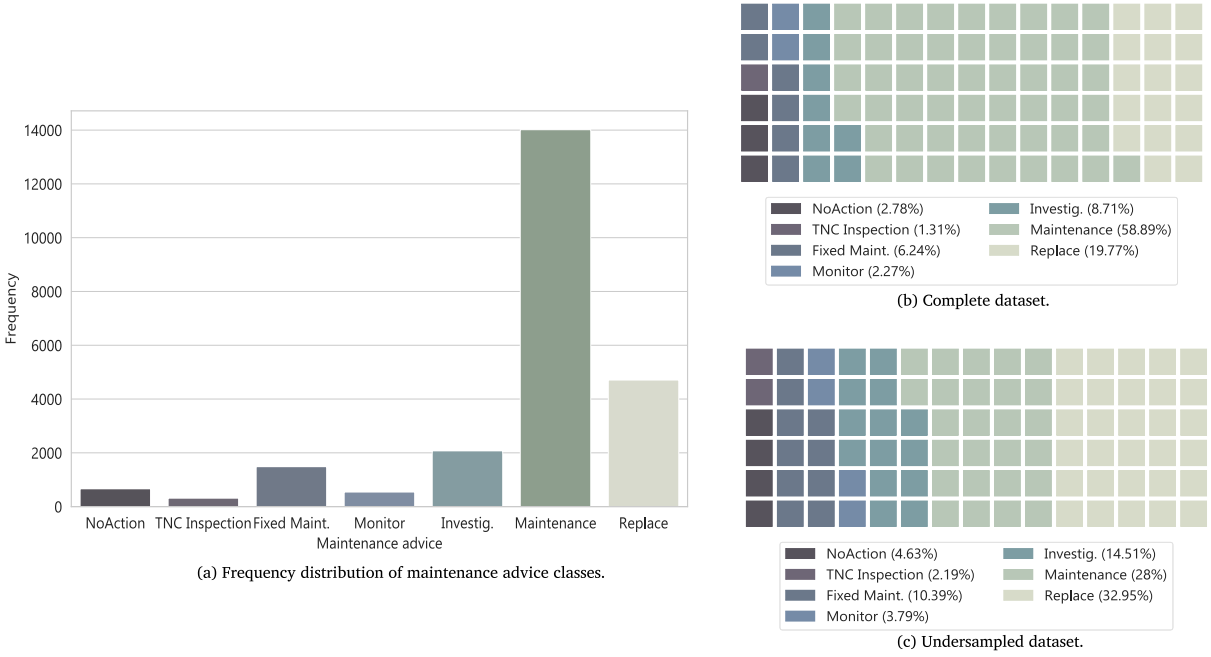


(b) Complete dataset.



(c) Undersampled dataset.

Fig. 5. Representation of risk level classes in the overall data set.



(a) Frequency distribution of maintenance advice classes.

(b) Complete dataset.

(c) Undersampled dataset.

Fig. 6. Representation of maintenance advice classes in the overall data set.

result of many decision trees, which individually suffer from high variance [50]. This ensemble learning approach results in a robust model that is less susceptible to over-fitting.

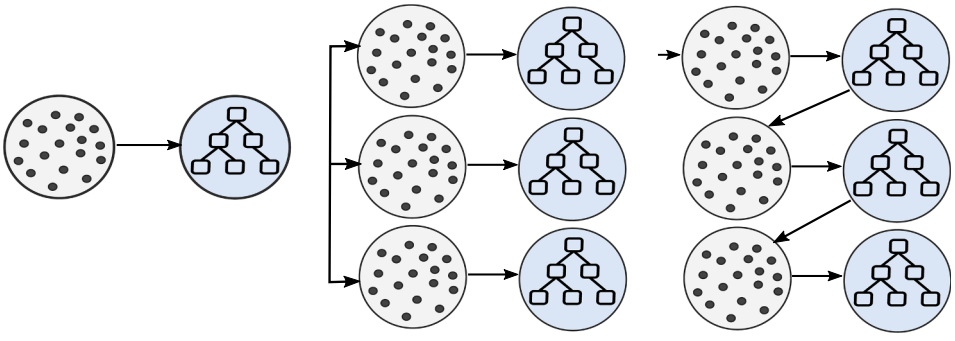
Gradient Boosting Trees is an alternative ensemble learning technique that consecutively produces weak tree classifiers in a stage-wise fashion [43]. The boosting approach strategically resamples and sequentially builds multiple trees for instances that are difficult to estimate with previous ones by minimizing some arbitrary differentiable loss functions, e.g., cross-entropy or sum of squared errors. In other words, the idea is to convert the weak learners into a strong learner sequentially, where each weak learner tries to improve upon its predecessor.

Fig. 7 shows the visual representation of tree-based models. The decision tree develops a single tree of a whole dataset, whereas the random forest develops multiple trees at a time over the randomly selected data. The gradient boosting trees also develops several estimators but in a sequential manner. All the models are applied to the IMA process dataset using the scikit-learn library of python [52]. The hyper-parameters of the models are selected using the random search method [53]. The parameters are tuned over the subset of training data, also called validation set, to empirically optimizing the results of models.

4.2. Deep learning techniques

Deep learning is a subfield of ML whose algorithms are inspired by the structure and function of the brain called Neural Networks (NN). The pioneer researchers of deep learning define NN as “algorithms that seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels” [54]. To put simply, the deep learning algorithm follows several layers of abstraction to learn complex function mappings, rather than direct input to output [55]. The NN extract the useful features from data automatically and can improve themselves without human interventions, whereas ML algorithms require the clear set of manually extracted features and may require additional data to improve its predictive performance [56] (Chapter 1). The motivation to explore NN for the given prediction tasks is due to their state-of-the-art performance in computer vision [57,58], speech recognition [59,60], and natural language processing [61,62].

In this study, we hypothesize that compared to traditional learning algorithms for the structured data, the NN combined with entity embedding can result in better and robust predictive models. The ability of NN to learn complex non-linear representations from the data certainly



(a) Decision Tree (single tree).

(b) Random Forest (parallel trees).

(c) Gradient boosting trees (sequential trees).

Fig. 7. Conceptual representation of tree-based learning techniques [51].

improves the prediction task at hand. Additionally, the learned data representations can also be transferred for the related prediction tasks, where the data is insufficient, outdated, or unlabeled in nature [63]. In this section, we briefly introduce the basic structure of the neural network, followed by explanations of entity embedding for the categorical data. Along the lines, we present the concept of cost-sensitive learning (also called class weights) in order to manage the imbalanced nature of the IMA dataset. Next, we explain the multi-task learning framework of NN, which tries to learn several tasks simultaneously instead of developing a discrete model for each task.

A **neural network** is typically represented by a network diagram consisting of several layers. The basic computation unit is a node (also called neuron), which contains an activation function such as a sigmoid function or a rectified linear unit (ReLU). The supervised learning procedure within NN has a cyclic pattern, where the forward activation flow of output and backward error propagation for the weights adjustments is repeated number of times. The backpropagation is based on a learning rule such as perceptron learning, delta learning, etc., which modifies the weights of the edges based on the input pattern. To put in other words, when the data is presented to the NN for the first time, the output layer provides a mere guess of the output. This procedure is called forward activation flow. Based on the output, appropriate adjustments are made concerning the logic of learning rule and associated weights, which is referred to as backpropagation.

In principle, a neural network can approximate any continuous function since the data continuity guarantees the convergence of the optimization (see Nielsen [64] for an interactive visualization). However, structured data with their categorical features lack the required continuity, which limits the application of NN. Even with coded categorical features, the NN do not work well as the numerical coding eliminates the informative relations among the features. Guo and Berkhahn [65] proposed to use the entity embedding to learn the representation of categorical features in a multi-dimensional space. Given that IMA dataset comprises of categorical and numerical features, we implemented **neural networks with entity embeddings (NN-EE)** in this paper. The architecture of NN with entity embedding is depicted in Fig. 8.

All the categorical features in the dataset (see Table 2) are assigned with numerical codes that are mapped as vectors to develop entity embedding. The mapping is equivalent to an extra layer of neurons, which is added on the top of the input layer and is learned in an end-to-end manner. The numerical features are fed directly to the fully connected layers with 20 hidden units. The output of the embedding layers and the fully connected layer are concatenated and connected to two fully connected layers, each having 128 and 64 hidden neurons. After each dense layer, we applied dropout with 0.1% probability, which randomly drops the neuron from the layers to avoid overfitting and to improve the generalizability of the model. We also applied L2 regularization to the weights of dense layers. At the output layer, the softmax function is applied to obtain the normalized output probabilities, where a class having the highest probability is the predicted class. To tackle the class imbalance problem, mentioned in Section 3.4, we applied cost-sensitive learning by using weighted categorical cross-entropy loss function. In this case, the weights are assigned to the classes based on their distribution in the training set, where the higher weights are assigned to the minority classes and lower weights to the majority class. The **NN-EE with class weights (NN-EE(cw))** handles the data imbalance problem at the algorithmic level without performing any under or oversampling of the actual data. For all the prediction tasks, we perform experiments using NN-EE with and without class weights to analyze the difference in predictive performance.

In the above-mentioned learning settings, there is one task to solve by minimizing a single loss function. Though the prediction of condition state, risk level, and maintenance advice are related, the single-task models treat them independently. In other words, three independent models (i.e., NN-EE) are developed for aforementioned classification tasks, where the representations learned for one task are not shared or used for learning another (similar) task. The framework of multi-task learning argues that single task learning may ignore the potentially useful information that is available from the related tasks [66]. It is inspired by human-learning principles, where we use the knowledge obtained from previous tasks to learn related tasks efficiently. Multi-task learning aims to develop a unified model by using shared hidden

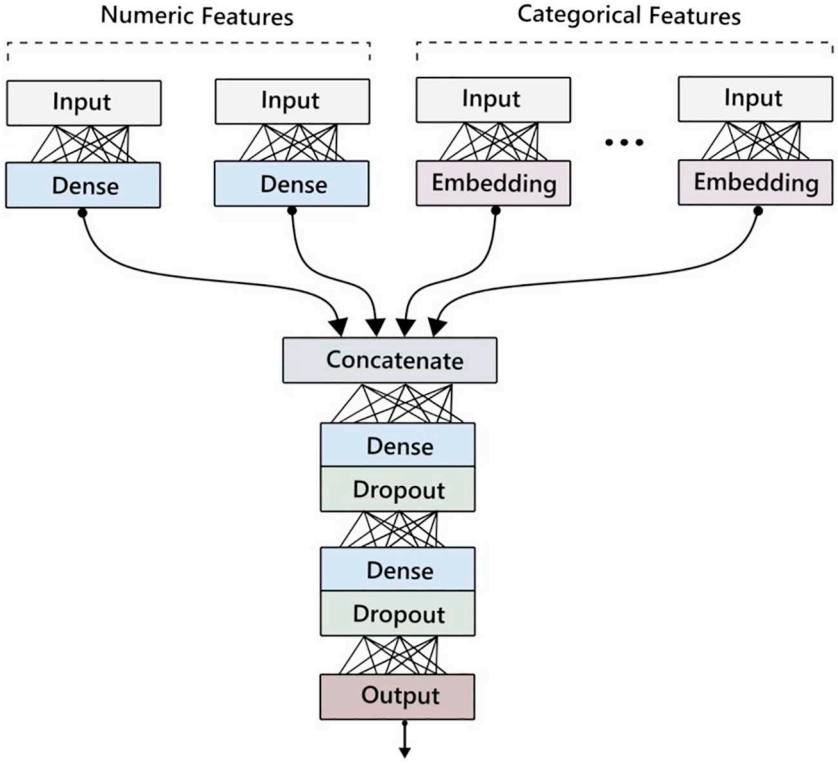


Fig. 8. Architecture of neural network with entity embeddings (NN-EE).

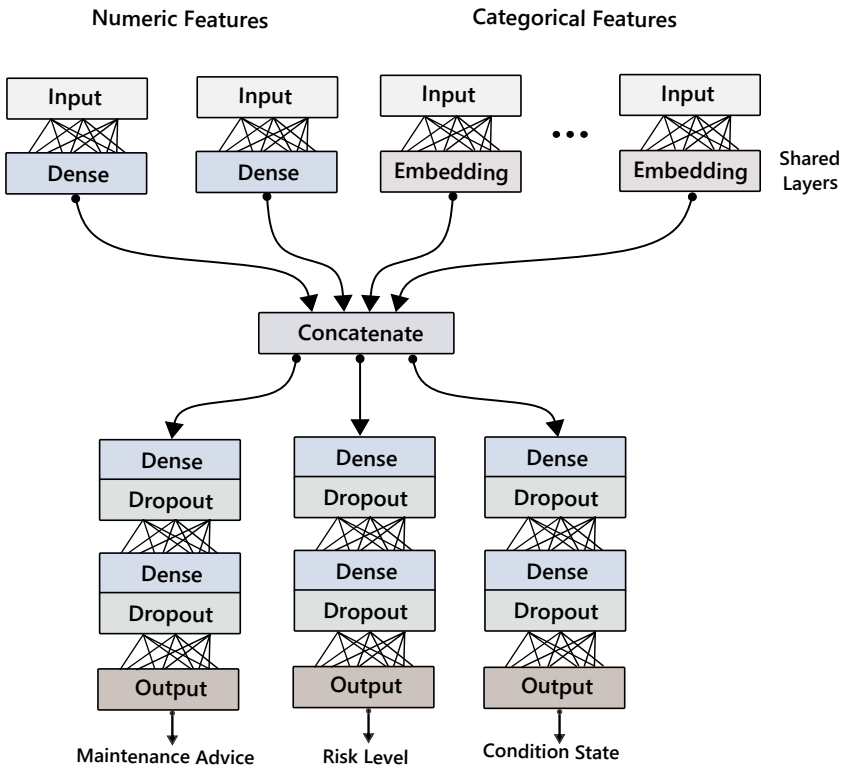


Fig. 9. Multi-task neural networks with entity embedding and two shared layers.

layers that are trained in parallel on all the related tasks. Therefore, the **multi-task learning neural networks (MTL-NN)** consists of common layers across multiple tasks as well as task-specific layers. Fig. 9 presents the architecture of MTL-NN, which seeks to develop a single unified model for the prediction of all three tasks. MTL-NN is performed through hard or soft parameter sharing [66]. We applied hard-parameter sharing in which initial hidden layers are shared across all the tasks, whereas the final layers are problem-specific. The entity embedding layers, as well as dense layers for the numeric features, are shared among the tasks. Likewise, the task-specific layers have the same configuration, as noted in single-task architecture. We applied L2-regularization on shared and on task-specific layers to avoid over-fitting. Finally, the optimization of the loss function is done simultaneously by alternating between different tasks randomly. The categorical (weighted) cross-entropy is optimized as an objective function by 'Adam' optimizer. Similar to a single task NN-EE, we applied class weights to MTL-NN to tackle the class imbalance problem. The architecture of NN-EE and MTL-NN is shown in Figs. 8 and 9 and their parametric details are implemented with python neural network API keras [67].

The objective of MTL-NN is to benefit from the shared representations, where the features learned for one task may improve the learning of other tasks. The shared representation in MTL-NN is shown to improve the generalization performance of multiple tasks when they are related [68]. Due to joint representation among related tasks, MTL-NN is likely to perform better compared to single-task specific NN-EE(cw). Additionally, these pre-trained shared layers from our multi-task network can be used as an initialization (transfer learning) for rapidly solving other related tasks, where labeled data is scarce.

4.3. Evaluation approaches

An optimal predictive model must be able to generalize well for new (unseen) data. We applied stratified random sampling and cross-validation to evaluate the performance of the models based on various metrics. In the following sections, a description of the evaluation approach and performance metrics is provided.

4.3.1. Stratified random sampling

In the Stratified Random Sampling (SRS) approach, the entire dataset is randomly split into training and test sets. In contrast to the standard sampling, the SRS ensures that each data split has equal target class representation. The training set is used to train the model, and the test set is used to evaluate the model's performance. Typically, 70% of the data instances are selected for training, and the rest 30% is used for testing. We also further split the training set to get the validation set, which is used to tune the hyper-parameters of different models by applying a random search strategy [53]. In the final evaluation phase, the validation set was then combined, and the model is trained on the whole training set and evaluated on the held-out test set.

4.3.2. Stratified cross validation

In Stratified Cross-Validation (SCV), the whole dataset is randomly split into a number of equally sized units referred as 'folds'. By having N number of folds, the $N-1$ are used for the training, while the N_{th} fold is used for the model testing. This process is repeated N times until each fold had the opportunity of being used as N_{th} test and training fold. Finally, the output is averaged across all folds to estimate the performance of the model. This method ensures that every data point is used at least once as a training example and once as a test example. The SCV is performed for the completeness of validations and the evaluation of the model's robustness.

4.3.3. Performance metrics

Several metrics can be used to evaluate the performance of the predictive model. For the classification tasks, the confusion matrix analysis is used, which represents the models' predicted classes on test data for which the true values are already known. It is essential to introduce the confusion matrix first in order to explain relevant performance measures. Table 3 shows the confusion matrix for the binary

Table 3
Confusion matrix for binary classification problem.

	Predicted negative	Predicted positive
Actual negative	True negatives (TN)	False positives (FP)
Actual positive	False negatives (FN)	True positives (TP)

classification problem having positive and negative as target classes.

The values at principal diagonal confusion matrix (i.e., TN and TP) represent the correct classification by a model. However, the secondary diagonal values (i.e., FP and FN) show the misclassifications. To elaborate further, the False Positives (FP) are *positive* samples that are incorrectly classified as a *negative* class. Similarly, False Negative (FN) are *negative* class samples incorrectly classified as *positive*. A normalized confusion matrix with perfect classification has TN and TP of value *one* and FP and FN of value *zero*. With the confusion matrix, a number of performance measures can be calculated [69]. The metrics used in this study are explained as follows:

Accuracy is a measure of correct predictions compared to the available data instances. It shows how often the model classifies the instances correctly. The accuracy is a good measure when the data is balanced for each class. However, in the case of an imbalanced dataset, this metric without other performance measures can be misleading. The accuracy is computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

F-score is a combination of precision (or positive predictive value) and recall (sensitivity) measures [70]. The precision determines the exactness of the model. It is a ratio of correctly predicted positive instances (TP) to the total positively predicted instances (TP + FP). In contrast, the recall provides a measure of the model's completeness. It is a ratio of correctly predicted positive instance (TP) to the total instance of the positive class (TP + FN) in test data. In other words, the precision represents the model's performance with respects to false positives, whereas the recall shows the performance with regards to false negatives. The F-score conveys the balance between precision and recall by taking their weighted harmonic mean. F-score is calculated as follows:

$$F - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Similar to the accuracy, F-score performs well with the fairly balanced dataset. In the case of an imbalanced dataset, the adjusted F-measure is utilized.

Cohen's Kappa presents an inter-rater agreement between qualitative items, which measures the relative observed agreement (p_o) with the hypothetical probability of chance agreement (p_e) [71]. The kappa measure does not only calculate the percentage accuracy but also considers the possibility of an agreement between raters (qualitative items) by chance. The value of kappa is calculated as follows:

$$\text{Kappa} = \frac{p_o - p_e}{1 - p_e}$$

In the case of imbalanced datasets, the kappa is a robust measure compared to F-score and accuracy. It can be said that kappa determines how well a model performed (p_o) as compared to how well it could have performed by chance (p_e) while considering the marginal distribution of a target class. The value of cohen's kappa ranges between -1 and 1.

5. Results

For three classification tasks, several predictive models are developed by applying learning algorithms. The evaluations of predictive models for each task and the results of multi-task learning neural networks are reported in this section.

5.1. Condition state prediction

The condition state prediction is a multi-class classification problem where an instance can belong to any of the five possible condition states (see Fig. 4). The objective of the classifier is to accurately predict the condition state of an unseen instance given the data of selected features (see Table 2). We developed five distinct models for this purpose, where the logistic regression and decision trees are applied as baseline techniques.

Table 4 shows the evaluation results with a Stratified Random Sampling (SRS) approach. In SRS, the dataset is randomly divided into train and test sets to ensure the equal class representation. Additionally, we also evaluated our model by under-sampling the majority class up to 40% in order to tackle the class imbalance problem (discussed in detail in Section 3.4). The logistic regression obtains inferior *kappa* value, which depicts a random approximation by the model. All the tree-based classification models have negligible performance differences for the *test set*. The under-sampling approach has improved the accuracy of tree-based models. For instance, the *kappa* value of gradient boosting trees is improved from 0.56 on the *test set* to 0.64 on the *under-sampled set*. The neural network with entity embedding (NN-EE) performed significantly good among all the models. The performance of NN-EE is further improved by assigning class weights, which tackle the data imbalance problem at the algorithmic level, as shown with bold text in Table 4. By the addition of class weights to NN-EE, the *accuracy* and *F-score* is approximately 80% with the *kappa* value of 0.70 on the *test set*. In other words, the NN-EE (cw) model can classify the correct condition state of an element with 80% accuracy given the inspection details.

The same set of models is further evaluated with a 10-fold Stratified Cross-Validation (SCV) approach introduced in Section 4.3. Table 5 shows the averaged scores across the 10-folds along with standard deviations on the test set and the under-sampled set. The NN-EE (cw) has performed best among all the models with 81% *accuracy*, 82% *F-score*, and 0.73 *kappa* values on the complete *test set*. Approximately, all the models show slightly improved performance scores compared to SRS. This is due to the difference of validation approach, where the SRS approach evaluates the model on an unseen test set, and the SCV approach trains and test the model iteratively on a randomly chosen subset of data.

Fig. 10 presents the confusion matrix analysis of logistic regression

Table 4
Results of condition state prediction with SRS on complete and under-sampled test set.

Classifiers	Test set			Under-sampled majority class		
	Accuracy	F-Score	Kappa	Accuracy	F-Score	Kappa
Logistic Regression (LR)	0.5082	0.3724	0.0362	0.3511	0.3126	0.1046
Decision Tree (DT)	0.6980	0.7037	0.5488	0.7022	0.7032	0.6034
Random Forest (RF)	0.7196	0.7103	0.5579	0.7323	0.7317	0.6422
Gradient Boosting Trees (GBT)	0.7271	0.7180	0.5674	0.7326	0.7327	0.6446
NN with Entity Embeddings (NN-EE)	0.7877	0.7914	0.6811	0.7906	0.7929	0.7224
NN-EE with class weights (NN-EE (cw))	**0.7967**	**0.8081**	**0.7089**	**0.791**	**0.7946**	**0.7244**

Table 5
Results of condition state prediction with SCV on complete and under-sampled test set.

Classifiers	Test set			Under-sampled majority class		
	Accuracy	F-score	Kappa	Accuracy	F-score	Kappa
LR	0.5083 ± 0.0020	0.3768 ± 0.0024	0.0439 ± 0.0034	0.3553 ± 0.008	0.3243 ± 0.0090	0.1113 ± 0.0109
DT	0.7005 ± 0.0044	0.7062 ± 0.0046	0.5499 ± 0.0067	0.7131 ± 0.0068	0.714 ± 0.0067	0.6178 ± 0.0090
RF	0.7185 ± 0.0055	0.7143 ± 0.0048	0.5616 ± 0.0078	0.748 ± 0.0038	0.7473 ± 0.0041	0.6632 ± 0.0052
GBT	0.7219 ± 0.0098	0.7112 ± 0.0117	0.5563 ± 0.0182	0.7345 ± 0.0146	0.7343 ± 0.0147	0.6471 ± 0.0194
NN-EE	0.8158 ± 0.0062	0.8069 ± 0.0062	0.7165 ± 0.0074	0.8222 ± 0.0036	0.8238 ± 0.0037	0.7639 ± 0.0047
NN-EE(cw)	0.8128 ± 0.0050	0.8243 ± 0.0048	0.7328 ± 0.0067	0.8253 ± 0.0029	0.8283 ± 0.0029	0.7695 ± 0.0038

(LR), gradient boosting trees (GBT) and neural networks with entity embedding and class weights (NN-EE(cw)). The analysis provides a summary of correctly and incorrectly classified instances for each class. Fig. 10a presents the confusion matrix of LR as a baseline, where the LR model shows the poor classification performance. For instance, an instance having condition state *very good* (first row of Fig. 10a) is only 2% of times correctly classified, whereas it is 50% times classified as *good*, 35% as *reasonable*, and 13% times as *bad*. The results of GBT presented in Fig. 10b are significantly better; however, the first three classes are still relatively poorly classified with below 80% accuracy. The result of NN-EE (cw) in Fig. 10c shows further improvements, where the model can correctly classify the last three classes (i.e., reasonable, bad and very bad) with more than 80% of times. The confusion matrix also reveals that the model often (24% of times) confuses an instance of class *good* as *very good*. This can be attributed to similar damage details that cause the misclassification of these classes.

5.2. Risk level prediction

The prediction models can classify the risk level of a bridge (element) to five classes namely *negligible*, *limited*, *increased*, *high*, and *unacceptable* (see Fig. 5). Table 6 provides the results of models' evaluation with the SRS approach on the complete and under-sampled data. Comparative to condition state prediction, all the prediction models attain relatively better performance scores. This means it is easier for a classifier to relate damage features to risk levels directly. The NN-EE (cw) shows the best performance among all the models with an *accuracy* of 87% and a *kappa* value of 0.76 on the *test set*, as depicted by bold text in Table 6. Additionally, the NN-EE (cw) obtained significantly improved *kappa* score (0.76) compared to the NN-EE without class weights (0.68). By applying the under-sampling approach, all the models show improved performance except for the NN-EE (cw). It can be due to a relatively balanced dataset resulting from under-sampling, which might have eliminated the benefits of cost-sensitivity learning applied to NN-EE.

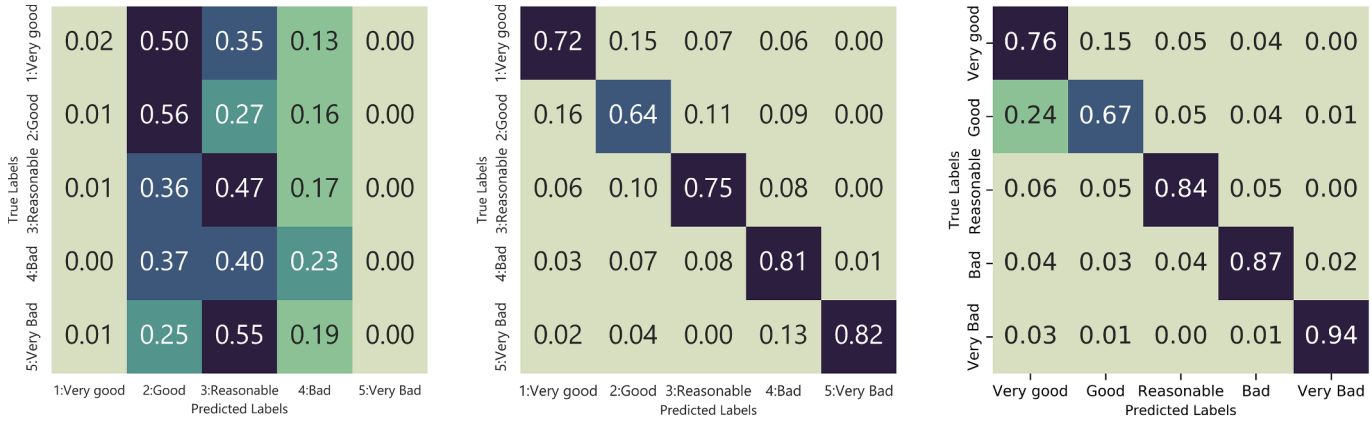
The developed models are also evaluated with SCV approach on a complete and under-sampled set. The averaged values of SCV across 10-folds and their standard deviation are provided in Table 7. The NN-EE (cw) performed best among all the models with 0.78 *kappa*, whereas the GBT has the best predictive performance among tree-based models with *0.60 kappa* value on the *test set*. The obtained results further validate the robustness of the models, which are trained and tested on the various folds of IMA dataset.

In addition to the numerical performance measures, we have performed confusion matrices analysis to explore the classes that are difficult to classify for predictive models. The confusion matrices for LR as a baseline, GBT as a best tree-based model, and NN-EE(cw) model as best classifiers are provided in Fig. 11. The LR model poorly classifies all risk levels as *limited* and *increased*. This can be attributed to high-class imbalance, which introduces a bias in favor of the majority classes. The confusion matrix of GBT shows better classification compared to LR.

However, the risk level class *unacceptable* is 84% of times predicted as *increased* class. The prediction of *high* risk level is also misclassified as *increased* risk level for at least 28% of the times. The confusion matrix of NN-EE(cw) shows better classification of instances to its respective risk levels. This is because NN-EE (cw) become invariant to class imbalance when aided with class weights.

5.3. Maintenance advice prediction

Typically, the decision-makers suggest the appropriate maintenance advice after analyzing the details of the damages noted during the inspection. We trained the models on the historical maintenance advice data and damages details. In the test phase, the model is presented only with the inspection data having damage details. A good classifier must



(a) LR as a baseline.

(b) GBT having best performance among tree-based algorithms.

(c) NN-EE(cw) having best performance among all.

Fig. 10. Confusion matrices of condition state prediction with SRS on under-sampled test set.

Table 6
Results of risk level prediction with SRS on the complete and under-sampled test sets.

Classifiers	Test set			Under-sampled majority class		
	Accuracy	F-Score	Kappa	Accuracy	F-Score	Kappa
Logistic Regression (LR)	0.6272	0.4999	0.0236	0.5221	0.4654	0.0884
Decision Tree (DT)	0.7452	0.7433	0.4949	0.6997	0.6991	0.4908
Random Forest (RF)	0.7497	0.7417	0.4839	0.7516	0.7481	0.567
Gradient Boosting Trees (GBT)	0.8076	0.8056	0.6157	0.8037	0.8026	0.664
NN with Entity Embeddings (NN-EE)	0.8420	0.8407	0.6888	0.8516	0.8487	0.7446
NN-EE with class weights (NN-EE (cw))	**0.8705**	**0.8738**	**0.7626**	**0.8309**	**0.8317**	**0.7187**

Table 7
Results of risk level prediction with SCV on complete and under-sampled test sets.

Classifiers	Test set			Under-sampled majority class		
	Accuracy	F-Score	Kappa	Accuracy	F-Score	Kappa
LR	0.6266 ± 0.0028	0.5019 ± 0.0044	0.0251 ± 0.0076	0.5238 ± 0.0107	0.4656 ± 0.0105	0.0912 ± 0.0195
DT	0.7529 ± 0.0082	0.7512 ± 0.008	0.5119 ± 0.0154	0.7238 ± 0.0117	0.7236 ± 0.012	0.5311 ± 0.0183
RF	0.7464 ± 0.0108	0.74 ± 0.011	0.4828 ± 0.0219	0.7701 ± 0.006	0.7677 ± 0.0058	0.6014 ± 0.0099
GBT	0.8041 ± 0.0105	0.8023 ± 0.0119	0.6089 ± 0.0244	0.8235 ± 0.0088	0.8232 ± 0.0091	0.6974 ± 0.0153
NN-EE	0.8597 ± 0.0067	0.8594 ± 0.0069	0.7279 ± 0.013	0.8593 ± 0.0087	0.8585 ± 0.0084	0.7273 ± 0.0169
NN-EE(cw)	0.8806 ± 0.0066	0.8841 ± 0.0065	0.7835 ± 0.0117	0.8807 ± 0.0058	0.8844 ± 0.0056	0.7840 ± 0.01

predict the correct maintenance advice for an inspection instance. These classifiers can assign an instance to one of the seven categories which are *no action*, *technical inspection*, *fixed maintenance plan*, *monitor*, *further investigation*, *maintenance*, and *replace* (see Fig. 6). In following, several classifiers are evaluated for the prediction performance using SRS and SCV evaluation approaches.

Table 8 shows the results of various models that are trained and evaluated on a complete and under-sampled dataset for the prediction of maintenance advice. The tree-based models performed significantly better with *accuracy* above 80% and *kappa* value above 0.70 on the *under-sampled set*. The NN-EE (cw) model is the best performing model with *accuracy* and F-*score* of 88% and *kappa* of 0.84 *on under-sampled set*, as shown by bold text in Table 8.

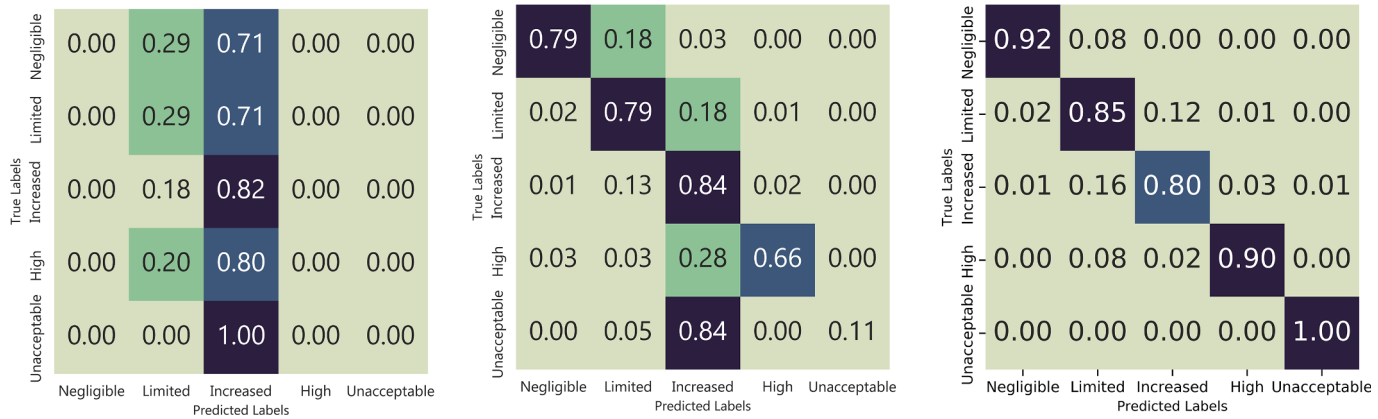
The maintenance advice classifiers are further evaluated for their robustness with SCV approach on the complete and under-sampled set. The evaluation results using SCV are presented in Table 9. As noted for all the above cases, the NN-EE (cw) performed best among all the evaluated models with *accuracy* and F-*score* of 87% and *kappa* of 0.80 for both *complete* and *under-sampled set*.

For further investigation of the classification capability of the models, we performed confusion matrix analysis. Fig. 12 shows the confusion matrix of LR as a baseline, GBT as a best tree-based model, and NN-EE (cw) as the best performing model. Even after under-sampling of the majority class (i.e. *maintenance*), the LR confusion matrix, presented in Fig. 12a, shows very poor performance. This is due to class imbalance as the under-sampling approach does not completely balance all the classes, and a model tends to favor the majority classes over the minority classes. On the other hand, the GBT model with the same dataset shows very good classification except for *fixed maintenance, no action* and *monitor* classes (see Fig. 12b). The NN-EE (cw) model shows significant performance improvement as shown in Fig. 12c.

5.4. Multi-task learning

The models, discussed thus far, treat each prediction problem independent of each other. In the multi-task learning framework, we developed a unified neural network that learns shared representation as well as problem-specific features to further improve the model performance (see Fig. 9 for architectural details). This section reports the results of multi-task learning (MTL-NN) applied for the prediction of condition state, risk level, and maintenance advice prediction.

Table 10 presents the MTL-NN evaluation results on SRS on a



(a) LR as a baseline.

(b) GBT having best performance among tree-based algorithms.

(c) NN-EE(cw) having best performance among all.

Fig. 11. Confusion matrices of risk level prediction with SRS on under-sampled test set.

Table 8
Results of maintenance advice prediction with SRS on complete and under-sampled test set.

Classifiers	Test set			Under-sampled majority class		
	Accuracy	F-score	Kappa	Accuracy	F-score	Kappa
Logistic Regression (LR)	0.5804	0.4505	0.0222	0.4969	0.4389	0.2518
Decision Tree (DT)	0.797	0.8012	0.6707	0.807	0.8074	0.7404
Random Forest (RF)	0.8002	0.7997	0.6664	0.8303	0.8288	0.7704
Gradient Boosting Trees (GBT)	0.8089	0.8102	0.6842	0.8341	0.8335	0.7756
NN with Entity Embeddings (NN-EE)	0.85	0.8507	0.7522	0.871	0.872	0.8266
NN-EE with class weighs (NN-EE(cw))	**0.8634**	**0.8695**	**0.7909**	**0.883**	**0.8842**	**0.8442**

Table 9
Results of maintenance advice prediction with SCV on complete and under-sampled test set.

Classifiers	Test set			Under-sampled majority class		
	Accuracy	F-score	Kappa	Accuracy	F-score	Kappa
LR	0.5903 ± 0.004	0.48 ± 0.0122	0.0716 ± 0.0207	0.4869 ± 0.0162	0.421 ± 0.0167	0.2322 ± 0.0257
DT	0.7912 ± 0.0101	0.793 ± 0.0093	0.6559 ± 0.0153	0.8153 ± 0.0121	0.8161 ± 0.0118	0.7507 ± 0.0161
RF	0.7927 ± 0.0079	0.791 ± 0.0073	0.6537 ± 0.0121	0.8333 ± 0.01	0.8324 ± 0.01	0.7746 ± 0.0131
GBT	0.8048 ± 0.0134	0.8052 ± 0.0116	0.674 ± 0.0207	0.8321 ± 0.0075	0.8311 ± 0.0076	0.7731 ± 0.0093
NN-EE	0.8650 ± 0.0073	0.8663 ± 0.0069	0.7802 ± 0.0119	0.8685 ± 0.0085	0.8684 ± 0.0071	0.7828 ± 0.0115
NN-EE(cw)	0.8690 ± 0.0083	0.8755 ± 0.0075	0.8009 ± 0.0121	0.8701 ± 0.008	0.8765 ± 0.0071	0.8026 ± 0.0114

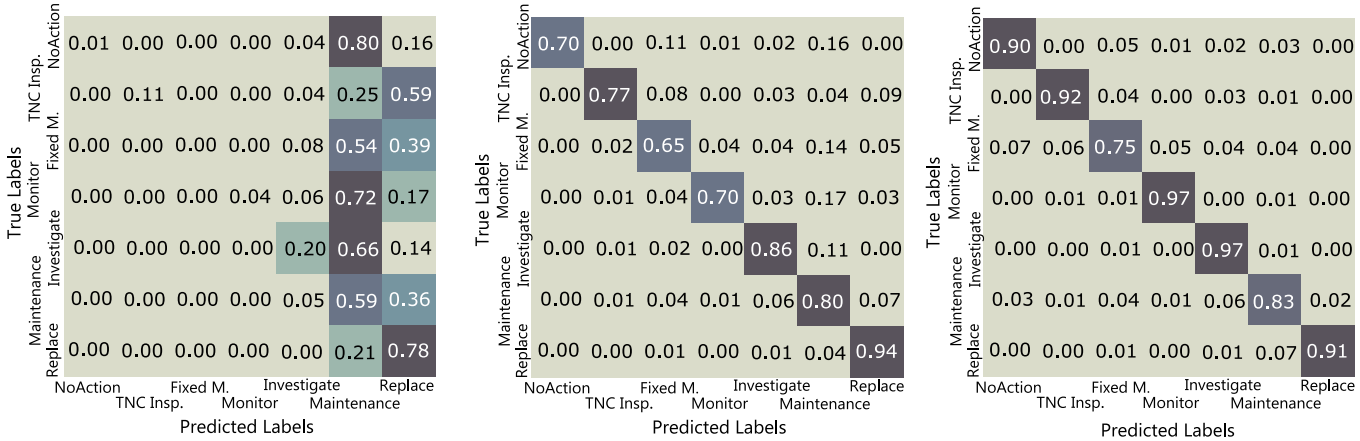
complete dataset. By comparing the results of NN-EE (cw) for condition state prediction on complete test set (see Table 4), we found that MTL-NN(cw) performed slightly better with the improvement of *kappa* value 0.1. The MTL-NN is shown to have improved performance compared to NN-EE without class weights. The same trends are noted for risk level prediction task. On the contrary, the MTL-NN (cw) for maintenance advice shows a slight decline in performance accuracy when compared with NN-EE(cw) (see Table 8).

In addition to numerical performance measures, the confusion matrices analysis of MTL-NN(cw) is performed for each prediction task. The resulting confusion matrices are shown in Fig. 13. The confusion matrix of condition prediction task in Fig. 13a shows significantly improved performance results specifically for the *good*, *reasonable* and *very bad* classes. For the risk level task, the MTL-NN(cw) model shows significant improvement in classification for two risk classes namely *increased* and *high* risk (see Fig. 11c for comparison). The confusion matrix of maintenance advice task in Fig. 13c presents slightly decline in performance compared to the NN-EE(cw) model of Fig. 12c. It can be noted that some classes of maintenance advice task shows decline in performance (such as *investigation*, and *monitor*), where others (such as *fixed maintenance* and *technical inspection*) shows improvements in classification accuracy.

In summary, the multi-task learning aims to improve the learning efficiency and prediction accuracy by optimizing multiple objectives from shared features representation. The goal to develop MTL-NN was not only to develop a unified model but also to improve the performance on individual tasks by learning shared representations. The MTL-NN (cw) model shows the improvement in classification results for two prediction tasks (condition state and risk level) compared to single-task learning.

6. Interpretability of models results

For the safety-critical domains such as health, manufacturing, and transportation, the decision-aid systems must be transparent and interpretable. The models developed using ML techniques are known for being black boxes, which provide little to no explanation of their prediction logic. The interpretable and explainable ML models are active research areas [72–74]. Miller [75] defines the interpretability as the degree to which a human can understand the cause of the decision. In



(a) LR as a baseline.

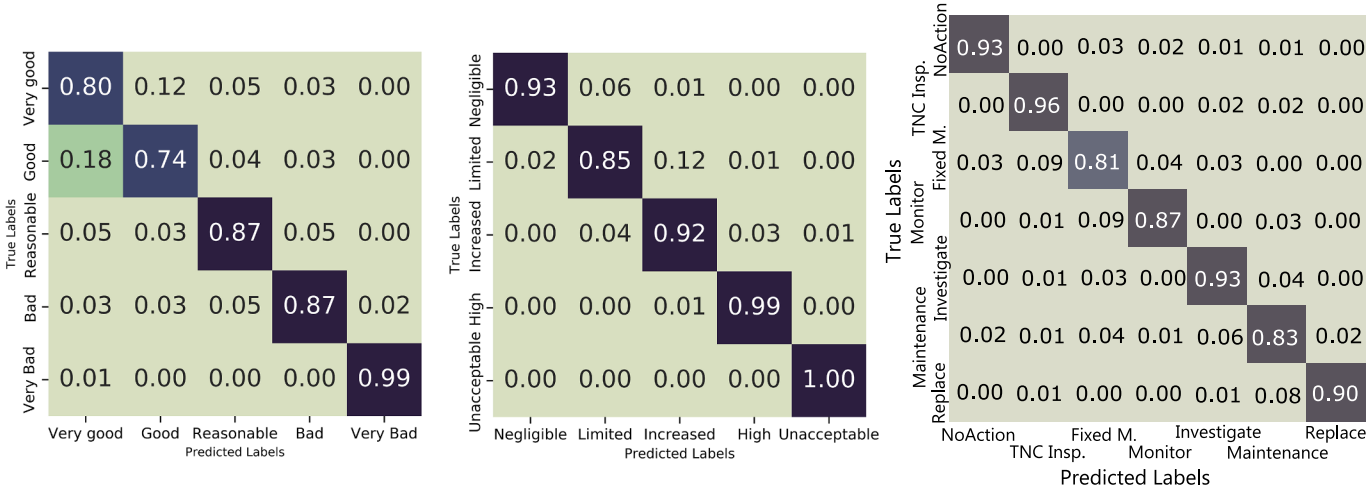
(b) GBT having best performance among tree-based algorithms.

(c) NN-EE(cw) having best performance among all.

Fig. 12. Confusion matrices of maintenance advice prediction with SRS on under-sampled test set.

Table 10
Results of multi-task learning (MTL-NN) with SRS on complete test set.

Classifiers	Condition state			Risk level			Maintenance Advice		
	Accuracy	F-score	Kappa	Accuracy	F-score	Kappa	Accuracy	F-score	Kappa
MTL-NN	0.8127	0.789	0.7034	0.8587	0.8545	0.7171	0.8611	0.8615	0.7727
MTL-NN(cw)	0.798	0.8092	0.7132	0.8765	0.88	0.7753	0.8572	0.8638	0.7809



(a) Condition state prediction. (b) Risk level prediction. (c) Maintenance advice prediction.

Fig. 13. Confusion matrices of all prediction tasks with SRS on test set by MTL-NN(cw) model.

other words, a model is said to be easily interpretable if a human decision-maker can comprehend and reason about the model's predictions drawn from his domain knowledge.

There are several model-agnostic techniques to interpret the performance of any ML model (see chapter 2 of Molnar [74] for a detailed overview). However, most of the techniques are mainly suitable for regression tasks. In our study, we provide the single instance level explanation of the NN-EE(cw) model by employing the Local Interpretable Model-Agnostic Explanations (LIME) framework [76]. By giving a new test instance to the trained NN-EE(cw), the LIME framework can explain the positively and negatively contributing features weights to classify an instance to a respective predictive class. The instance-level explanation enables the domain experts and decision-makers to understand, interpret, and possibly further improve the model's performance. Additionally, the LIME explanation also decides if the model is trustworthy since the model may sometimes pick the spurious correlation.

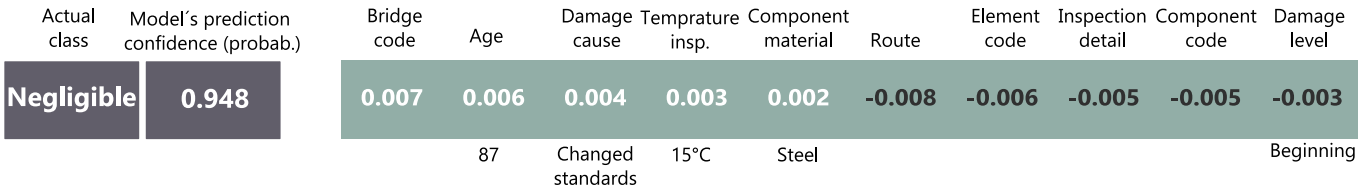
The instance-level explanation of NN-EE(cw) model for condition state, risk level and maintenance advice prediction are presented in Figs. 14, 15, and 16 respectively. We provide the explanation of model results for two randomly chosen instances from the test set for each prediction problem. The LIME explanations show the actual class of a test instance and the model's prediction confidence in terms of probability. The higher predicted probability shows that the model is confident in its predictions and vice versa. The LIME framework assigns weights to each feature, which quantifies features' importance in the overall prediction of the model. The positive value represents that the feature positively contributes to the model predictions, whereas the negative value shows that the features do not contribute towards the model's prediction.

In addition to the features' weight, the instance level explanation also provides the actual data values that are used for the model prediction. The value of feature *code* is omitted for the sake of brevity. It is

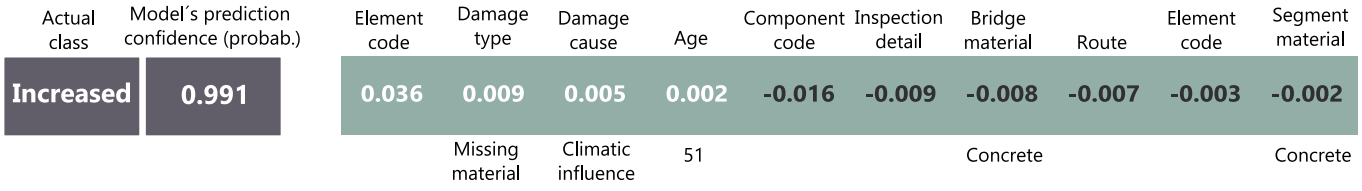
(a) Explanation of NN-EE(cw) model's prediction for *good* condition state class

(b) Explanation of NN-EE(cw) model's prediction for *very bad* condition state class

Fig. 14. Instance-level explanation of NN-EE(cw) model for condition state prediction using the LIME framework.



(a) Explanation of NN-EE(cw) model's prediction for *negligible* risk level class.



(b) Explanation of NN-EE(cw) model's prediction for *increased* risk level class.

Fig. 15. Instance-level explanation of NN-EE(cw) model for risk level prediction using the LIME framework.

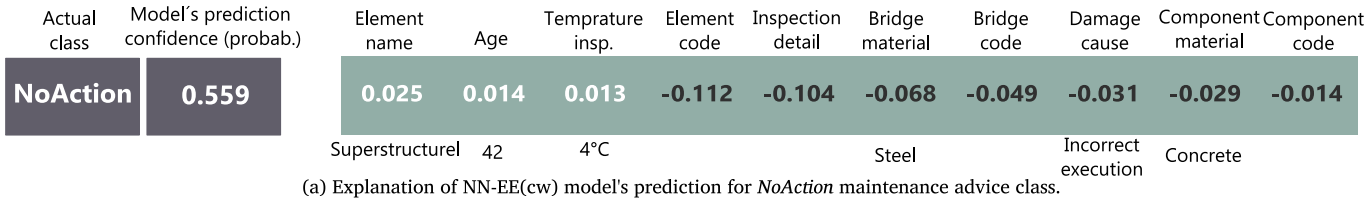
important to note that the LIME explanations illustrate only five most and least contributing features that facilitate model in prediction, where the NN-EE (cw) was trained on a set of twenty features (see Table 2).

Fig. 14a shows the explanation of NN-EE(cw) model trained for the condition state prediction. The explanation of a test instance shows that the element code, bridge material, and segment material are positively contributing features. In contrast, inspection details and other features are negatively influencing the models' results. The model classified the instance as good with 99% confidence. When given the other test instances, the same model may find a different set of features to be important, as shown in Fig. 14b. Therefore, the features that are negatively influencing one instance classification can positively contribute to the classification of another instance.

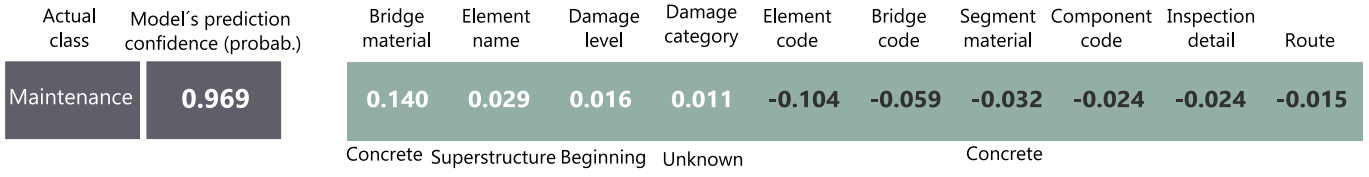
The explanation of risk level prediction model, given in Fig. 15a and b, finds the damage cause and age features quite important whereas the component code, route, and inspection details are negatively influencing the classification. The model correctly classified an instance to *negligible* risk class with 94% probability, and the other instance is classified as *increased* risk class with 99% probability. The classification logic of the maintenance advice model is also explained and presented in Fig. 16. For the *NoAction* class, the model shows only 55% confidence, which means that the model is likely to confuse this instance with other classes.

For each instance, a different set of features is most and least important; therefore, it is difficult to establish an overall feature importance score. For several instances, the bridge, element, and component codes are shown to have high importance weights. There are possibly two reasons for such behavior of the model. First, in practice, the decision-makers assess the condition state, risk level, and maintenance advice in the IMA process based on their inherent understanding of specific bridge and their components. Second, the model may find the data of similar codes from the dataset, establishes inherent correlations, and learns their characteristics during training. It is also interesting to note, for most of the cases, the element code is found to be important compared to component code. This is due to the data collection process, where the details of the damages are noted at the component level, whereas condition state and risk level is stored at the element level. Additionally, the instance-level analysis may also reveal the set of features that the predictive model finds useful, which may also differ from the real decision-making practices.

With the ability to better interpret the results of predictive models, domain experts and decision-makers can better interact with the process and trust the models' prediction. This gives a reliable decision-aid in the subjective assessment of bridges maintenance planning. Furthermore, the interpretability of the models can also reveal the hidden discrepancies and can be used for feature selections and models improvements activities.



(a) Explanation of NN-EE(cw) model's prediction for *NoAction* maintenance advice class.



(b) Explanation of NN-EE(cw) model's prediction for *maintain* maintenance advice class.

Fig. 16. Instance-level explanation of NN-EE(cw) model for the maintenance advice prediction using the LIME framework.

7. Discussion

This paper has investigated several tree-based algorithms and neural networks with entity embedding for the development of predictive models for bridge maintenance planning. Multiple learning algorithms such as gradient boosting trees and neural networks were trained and evaluated on the principal inspection data of bridges. The objective of the predictive models is to support asset owners in subjective assessment tasks of *Inspection to Maintenance Advice (IMA) process* for bridge maintenance decisions. Typically, the decision-maker retrieves the principal inspection data on the case-by-case basis to assess the condition state, risk level, and to decide about the maintenance actions. In contrast, the predictive models introduced in this paper can aid decision-makers in these subjective procedures by predicting the condition state, risk level, and maintenance advice efficiently with more than 80% accuracy.

For the development of predictive models, we particularly focused on utilizing only those datasets that are generated from the in-use business process of BMS. The use of available data within the agencies yields practical models that are well-aligned with the business practices and can readily be implemented in practice. With the intention to deploy the predictive models as support to subjective assessment procedures of BMS, our work especially takes into account the interpretability of the models' outcomes on single case-basis through employing the Local Interpretable Model-Agnostic Explanations (LIME) framework.

As result of model development process, few key points and details of additional conducted experiments are provided below:

- The overall IMA process (see Fig. 1) substantially relies on experts' opinions and their technical knowledge to decide the condition state, risk level, and maintenance advises. This makes the management of infrastructure vulnerable in case of leaving personnel or having new inspection managers.
- The IMA process records a large amount of data as a result of inspection, damage analysis, risk assessment, and risk control activities, which collectively consist of 73 features. A large number of features may give rise to missing and sometimes inconsistent data. Additionally, the manual feature analysis is time-consuming and difficult for decision-makers. We performed careful feature engineering (see Section 3.3) on the IMA dataset and chose only 20 features for the development of machine learning models (see Table 2).
- The bridge as a structure is decomposed to elements and components (see Table 1). Assuming that each element has different characteristics, we initially developed predictive models for each element individually. Contrary to the expectations, the element-specific classifiers performed relatively poorly due to a smaller subset of data. Conversely, the single model, for all elements combined, showed much better performance as presented in Section 5.
- The IMA dataset is collected via several subjective assessment procedures. Though the inspection officers are professionally trained, the data is still likely to have an inconsistent assessment. We performed rigorous exploratory data analysis and cleaning to identify and eliminate potentially inconsistent data instances. Several meetings with domain experts were also conducted to ensure the adequate quality of the dataset for the machine learning model.
- The IMA dataset is found to be imbalanced in terms of class representation (see Section 3.4). In addition to applying under-sampling and cost-sensitive learning to tackle data imbalance problems, we also applied the Synthetic Minority Over-Sampling Technique (SMOTE) to generate the data instance of minority classes. However, compared to under-sampling and class weights in objective functions, the SMOTE does not improve the predictive performance of the models.

To summarize, this paper develops several predictive models to provide support in the subjective assessment procedure of bridge maintenance planning. With each new principal inspection activity that reports the inspection results, our models can reliably classify the condition state, the risk level, and the optimal maintenance advise with certain accuracy. These support models will streamline the subjective assessment process and will also improve the data quality. Since the models use the data from the standard IMA process, the proposed model development approach is generic and can also be applied to related types of structures such as culverts, tunnels, sluices, which are often part of asset management system [3]. From the implementation perspective of these predictive models, it is worth noting that the real data evolve over time; for instance, new condition classes may be introduced, or new features can be recorded. The proposed multi-task learning network can be used to accommodate the evolving data as it can improve the performance on the (related) future tasks instead of training models from scratch.

8. Conclusion

This paper shows how the historical and operational data available at transport agencies can be used to support the asset owners from condition assessment procedures to maintenance planning. We used a large *Inspection to Maintenance Advice (IMA)* dataset of concrete highway bridges from the road agency. Based on the IMA decision procedure, we have developed several machine learning models that can predict condition states, possible risk levels, and finally, recommend the most suitable maintenance actions.

We explored various supervised algorithms from traditional machine learning to deep learning paradigm in order to find the optimal model for the prediction tasks. Among the tree-based methods, the gradient boosting tree models performed best with 0.56, 0.61 and 0.68 *kappa* values on the complete dataset for condition state, risk level, and maintenance advice prediction tasks, respectively. To develop predictive models with further improved predictive ability, we explored the neural network with entity embeddings for learning from structured data. Similarly, the class weights are implemented on NN-EE to tackle the class imbalance problem. The NN-EE with class weights improved the *kappa* score to 0.70, 0.76 and 0.79 with accuracy close to 80% for condition state, risk level, and maintenance advice, respectively. For all the given algorithms, the discrete predictive models, i.e., task-specific classifiers, are developed. Considering that the bridge maintenance planning may have numerous related tasks with shared features (input), we have explored the prospect of implementing a Multi-Task Learning framework (MTL). The MTL resulted in a unified model for all the prediction tasks where the *kappa* is further improved to 0.71 and 0.77 for condition state and risk level tasks and reduced to 0.78 *kappa* value for maintenance advice task. Also, the MTL model learns the shared representations of related tasks which are useful for learning future tasks in a low-data regime.

The developed predictive models can assist the asset owners in the process of condition assessment from visual inspection by analyzing similar cases from the past instantly. Instead of relying only on the subjective assessment, the machine learning models successfully extract useful insights from the raw inspection data and can predict the most relevant class with a sufficient probability. The developed models can readily be applied as a part of the IMA process to support the decision-makers in maintenance planning tasks. Additionally, the predictions of the models are evaluated by instance-level explanation to understand which features are essential and why models predict certain classes. The future work of this study aims to employ the monitoring data and images captured during the inspections in order to further automate the condition assessment procedure and to reduce the subjectivity.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This study has been performed under funding from the European Union's Horizon 2020 - Research and Innovation Framework Programme with grant agreement No 636285 DESTination Rail.

References

[1] J. Miles, K. Chen, W. Association, ITS Handbook: Recommendations From the World Road Association (PIARC), 2 ed., Artech House, Boston, Massachusetts, 2004.
[2] European Commission, Transport in the European Union: Current Trends and Issues, Tech. Rep. European Union, Brussels, Belgium, 2018 URL https://ec.europa.eu/transport/themes/infrastructure/news/2018-04-25-transport-european-union-current-trends-and-issues_en , Accessed date: 27 February 2020.
[3] Z. Mirzaei, B.T. Adey, L. Klatter, J.S. Kong, Overview of existing bridge management systems, 6th International Conference on Bridge Maintenance, Safety and Management, July 8–12, Stresa, Italy, International Association for Bridge Maintenance And Safety (IABMAS), 2012URL https://www.research-collection.ethz.ch/handle/20.500.11850/62546 , Accessed date: 27 February 2020.
[4] M.J. Markow, W.A. Hyman, Bridge Management Systems for Transportation Agency Decision Making, 397 The National Academies Press, Washington, DC, 2009, https://doi.org/10.17226/14270 (ISBN: 9780309098359).
[5] G.P. Bu, J. Lee, H. Guan, Y. Loo, M. Blumenstein, Prediction of long-term bridge performance: integrated deterioration approach with case studies, J. Perform. Constr. Facil. 29 (2014) (doi:(ASCE)CF.1943-5509.0000591).
[6] S.B. Chase, Y. Adu-Gyamfi, A. Aktan, E. Minaie, et al., Synthesis of National and International Methodologies Used for Bridge Health Indices, Technical Report, United States, Federal Highway Administration, Turner-Fairbank Highway Research Center, McLean, Virginia, 2016 URL: https://www.fhwa.dot.gov/publications/research/infrastructure/structures/bridge/15081/15081.pdf , Accessed date: 27 February 2020.
[7] V. Gattulli, L. Chiaramonte, Condition assessment by visual inspection for a bridge management system, J. Comput. Aided Civ. Infrastruct. Eng. 20 (2005) 95–107, https://doi.org/10.1111/j.1467-8667.2005.00379.x.
[8] S. Alaswad, Y. Xiang, A review on condition-based maintenance optimization models for stochastically deteriorating system, Reliab. Eng. Syst. Saf. 157 (2017) 54–63, https://doi.org/10.1016/j.ress.2016.08.009.
[9] J.S. Kong, D.M. Frangopol, Life-cycle reliability-based maintenance cost optimization of deteriorating structures with emphasis on bridges, J. Struct. Eng. 129 (2003) 818–828, https://doi.org/10.1061/(ASCE)0733-9445(2003)129:6(818).
[10] M. Liu, D.M. Frangopol, Optimal bridge maintenance planning based on probabilistic performance prediction, Eng. Struct. 26 (2004) 991–1002, https://doi.org/10.1016/j.engstruct.2004.03.003.
[11] X. Hu, S. Madanat, Determination of optimal MR&R policies for retaining life-cycle connectivity of bridge networks, J. Infrastruct. Syst. 21 (2015), https://doi.org/10.1061/(ASCE)IS.1943-555X.0000226.
[12] F. Ghodoosi, S. Abu-Samra, M. Zeynalian, T. Zayed, Maintenance cost optimization for bridge structures using system reliability analysis and genetic algorithms, J. Constr. Eng. Manag. 144 (2017), https://doi.org/10.1061/(ASCE)CO.1943-7862.0001435.
[13] G. Morcous, Z. Lounis, Maintenance optimization of infrastructure networks using genetic algorithms, Autom. Constr. 14 (2005) 129–142, https://doi.org/10.1016/j.autcon.2004.08.014.
[14] R. Ahmad, S. Kamaruddin, A review of condition-based maintenance decision-making, European Journal of Industrial Engineering 6 (2012) 519–541, https://doi.org/10.1504/EJIE.2012.048854.
[15] A. Sharma, G. Yadava, S. Deshmukh, A literature review and future perspectives on maintenance optimization, J. Qual. Maint. Eng. 17 (2011) 5–25, https://doi.org/10.1108/13552511111116222.
[16] S. Bush, T. Henning, J.M. Ingham, A. Raith, An agent-based framework for improved strategic bridge asset management, Austroads Bridge Conference, Sydney, New South Wales, Australia, 4.4 2014 URL http://hdl.handle.net/2292/25208.
[17] Y. Wijnia, P. Herder, The state of asset management in the netherlands, in: Proceedings of the 4th World Congress on Engineering Asset Management, October, 25–27, Queensland, Australia, Springer, London, 2010, pp. 164–172, https://doi.org/10.1007/978-0-85729-320-6_19.
[18] B.F. Spencer Jr., V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering (2019) 199–222, https://doi.org/10.1016/j.eng.2018.11.030.
[19] R. Abduljabbar, H. Dia, S. Liyanage, S. Bagloee, Applications of artificial intelligence in transport: an overview, Sustainability 11 (2019) 189, https://doi.org/10.3390/su11010189.
[20] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte, A. Fouilloy, Machine learning methods for solar radiation forecasting: a review, Renew. Energy 105 (2017) 569–582, https://doi.org/10.1016/j.renene.2016.12.095.
[21] I. Abdallah, V. Dertimanis, H. Mylonas, K. Tatsis, E. Chatzi, N. Dervilis, K. Worden, E. Maguire, Fault diagnosis of wind turbine structures using decision tree learning algorithms with big data, Proceedings of European Safety and Reliability Association, June 17–21, Trondheim, Norway, Taylor & Francis, London, UK, 2018, pp. 3053–3061, , https://doi.org/10.1201/9781351174664-382.
[22] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, Y. Si, A data-driven design for fault detection of wind turbines using random forests and XGboost, IEEE Access 6 (2018) 21020–21031, https://doi.org/10.1109/ACCESS.2018.2818678.
[23] F. Granata, S. Papirio, G. Esposito, R. Gargano, G. De Marinis, Machine learning algorithms for the forecasting of wastewater quality indicators, Water 9 (2017) 105, https://doi.org/10.3390/w9020105.
[24] Y.H. Kim, J. Im, H.K. Ha, J.-K. Choi, S. Ha, Machine learning approaches to coastal water quality monitoring using goci satellite data, GIScience & Remote Sensing 51 (2014) 158–174, https://doi.org/10.1080/15481603.2014.900983.
[25] J. Chin, V. Callaghan, I. Lam, Understanding and personalising smart city services using machine learning, the internet-of-things and big data, 26th International Symposium on Industrial Electronics (ISIE), June 18–21, IEEE, Edinburgh, UK, 2017, pp. 2050–2055, , https://doi.org/10.1109/ISIE.2017.8001570.
[26] M. Mohammadi, A. Al-Fuqaha, Enabling cognitive smart cities using big data and machine learning: approaches and challenges, IEEE Commun. Mag. 56 (2018) 94–101, https://doi.org/10.1109/MCOM.2018.1700298.
[27] J. Masino, J. Thumm, M. Frey, F. Gauterin, Learning from the crowd: road infrastructure monitoring system, Journal of Traffic and Transportation Engineering (English Edition) 4 (2017) 451–463, https://doi.org/10.1016/j.jtte.2017.06.003.
[28] V.M. Souza, R. Giusti, A.J. Batista, Asfault: a low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning, Pervasive and Mobile Computing 51 (2018) 121–137, https://doi.org/10.1016/j.pmcj.2018.10.008.
[29] F.J. Morales, A. Reyes, N. Caceres, L. Romero, F.G. Benitez, Automatic prediction of maintenance intervention types in roads using machine learning and historical records, Transp. Res. Rec. 2672 (2018), https://doi.org/10.1177/0361198118790624.
[30] H. Li, D. Parikh, Q. He, B. Qian, Z. Li, D. Fang, A. Hampapur, Improving rail network velocity: a machine learning approach to predictive maintenance, Transportation Research Part C: Emerging Technologies 45 (2014) 17–26, https://doi.org/10.1016/j.trc.2014.04.013.
[31] G. Manco, E. Ritacco, P. Rullo, L. Gallucci, W. Astill, D. Kimber, M. Antonelli, Fault detection and explanation through big data analysis on sensor streams, Expert Syst. Appl. 87 (2017) 141–156, https://doi.org/10.1016/j.eswa.2017.05.079.
[32] T. de Bruin, K. Verbert, R. Babuška, Railway track circuit fault diagnosis using recurrent neural networks, IEEE Transactions on Neural Networks and Learning Systems 28 (2016) 523–533, https://doi.org/10.1109/TNNLS.2016.2551940.
[33] E.K. Chalouhi, I. Gonzalez, C. Gentile, R. Karoumi, Damage detection in railway bridges using machine learning: application to a historic structure, Procedia Engineering 199 (2017) 1931–1936, https://doi.org/10.1016/j.proeng.2017.09.287.
[34] T. Böhm, Remaining useful life prediction for railway switch engines using classification techniques, International Journal of Prognostics and Health Management 2153-2648, 59 (2017).
[35] Z. Allah Bukhsh, A. Saeed, I. Stipanovic, A.G. Doree, Predictive maintenance using tree-based classification techniques: a case of railway switches, Transportation Research Part C: Emerging Technologies 101 (2019) 35–54, https://doi.org/10.1016/j.trc.2019.02.001.
[36] J. van der Velde, L. Klatter, J. Bakker, A holistic approach to asset management in the Netherlands, Struct. Infrastruct. Eng. 9 (2013) 340–348, https://doi.org/10.1080/15732479.2012.657650.
[37] Z. Allah Bukhsh, I. Stipanovic, G. Klanker, A. O'Connor, A.G. Doree, Network level bridges maintenance planning using multi-attribute utility theory, Struct. Infrastruct. Eng. 15 (2018) 1–14, https://doi.org/10.1080/15732479.2017.1414858.
[38] B.D. Barkdoll, Effects of Climate Change on Bridge Scour, World Environmental and Water Resources Congress, American Society of Civil Engineers, 2012, pp. 2532–2537, https://doi.org/10.1061/9780784412312.253.
[39] K. Gavin, L.J. Prendergast, I. Stipanovič, S. Škarič, et al., Recent development and remaining challenges in determining unique bridge scour performance indicators, The Baltic Journal of Road and Bridge Engineering 13 (2018) 291–300, https://doi.org/10.2113/gseegeosci.13.1.1.
[40] J. Bakker, L. Klatter, Risk based inspection (RBI) at Rijkswaterstaat, in: F. Biondini, D.M. Frangopol (Eds.), Proceedings of the 6th International Conference on Bridge Maintenance, Safety and Management, July 8–12, Stresa, Itlay, Taylor & Francis, London, UK, 2012, pp. 510–517 ISBN.
[41] L. Klatter, T. Vrouwenvelder, J.M. Van Noortwijk, Societal and reliability aspects of bridge management in the Netherlands, Structure & Infrastructure Engineering 5 (2009) 11–24, https://doi.org/10.1080/15732470701322743.
[42] S. Kaufman, S. Rosset, C. Perlich, Leakage in data mining: formulation, detection, and avoidance, Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining, August 21–24, San Diego, California, Association for Computing Machinery, New York, NY, USA, 2011, pp. 556–563, , https://doi.org/10.1145/2020408.2020496.
[43] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (5) (2001) 1189–1232 URL www.jstor.org/stable/2699986 ISSN: 00905364.
[44] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, Proceedings of the First Instructional Conference on Machine Learning, December 3–8, Piscataway, New Jersey, 242 2003, pp. 133–142 (doi:10.1.1.121.1424).
[45] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng.

21 (2009) 1263–1284, https://doi.org/10.1109/TKDE.2008.239.
[46] D.H. Wolpert, W.G. Macready, et al., No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82, https://doi.org/10.1109/4235.585893.
[47] H. Trevor, T. Robert, F. JH, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, NY, 2009 ISBN.
[48] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems? The Journal of Machine Learning Research 15 (2014) 3133–3181, https://doi.org/10.5555/2627435.2697065.
[49] S.B. Kotsiantis, Supervised machine learning: a review of classification techniques, in: Proceedings of Emerging Artificial Intelligence Applications in Computer Engineering, June, IOS Press, Amsterdam, The Netherlands, 2007, pp. 3–24, https://doi.org/10.5555/1566770.1566773.
[50] B. Leo, J. Friedman, C. Stone, R.A. Olshen, Classification and Regression Trees, Chapman and Hall/CRC, 2017.
[51] Xristica, What is the difference between bagging and boosting? URL: https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/.
[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830 URL https://scikit-learn.org/stable/ https://doi.org/10.5555/1953048.2078195.
[53] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305, https://doi.org/10.5555/2503308.2188395.
[54] Y. Bengio, et al., Learning deep architectures for AI, Foundations and Trends in Machine Learning 2 (2009) 1–127, https://doi.org/10.1561/2200000006.
[55] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, Proceedings of International Conference on Unsupervised and Transfer Learning Workshop, July, Journal of Machine Learning Research, Washington, USA, 2012, pp. 17–36, , https://doi.org/10.5555/3045796.3045800.
[56] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, https://doi.org/10.1038/nature14539.
[57] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the Association for Computing Machinery 60 (2017) 84–90, https://doi.org/10.1145/3065386.
[58] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ArXiv preprint, URL, 2014. https://arxiv.org/abs/1409.1556.
[59] G. Hinton, L. Deng, D. Yu, G. Dahl, R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, et al., Deep neural networks for acoustic modeling in speech recognition, IEEE Signal Process. Mag. 29 (2012) 82–97, https://doi.org/10.1109/MSP.2012.2205597.
[60] T.N. Sainath, A.-r. Mohamed, B. Kingsbury, B. Ramabhadran, Deep convolutional neural networks for LVCSR, Proceedings of International Conference on Acoustics, Speech and Signal Processing, May 26–31, IEEE, Vancouver, BC, Canada, 2013, pp. 8614–8618, , https://doi.org/10.1109/ICASSP.2013.6639347.
[61] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155, https://doi.org/10.5555/944919.944966.
[62] Y. Kim, Convolutional neural networks for sentence classification , ArXiv preprint, URL, 2014. https://arxiv.org/abs/1408.5882.
[63] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828, https://doi.org/10.1109/TPAMI.2013.50.
[64] M.A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2018 eBook. URL http://neuralnetworksanddeeplearning.com/.
[65] C. Guo, F. Berkhahn, Entity embeddings of categorical variables, ArXiv preprint, URL, 2016. https://arxiv.org/abs/1604.06737.
[66] R. Caruana, Multitask Learning, 1 Springer, 1997, pp. 41–75, https://doi.org/10.1023/A:1007379606734.
[67] F. Chollet, Keras, URL https://github.com/fchollet/keras, (2015) , Accessed date: 27 February 2020.
[68] Y. Zhang, Q. Yang, A survey on multi-task learning , ArXiv preprint, URL https://arxiv.org/abs/1707.08114, (2017) , Accessed date: 27 February 2020.
[69] P. Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, 1 ed., Cambridge University Press, 2012 ISBN.
[70] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (2009) 427–437, https://doi.org/10.1016/j.ipm.2009.03.002.
[71] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. 20 (1960) 37–46, https://doi.org/10.1177/001316446002000104.
[72] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, URL ArXiv, https://arxiv.org/abs/1702.08608, (2017).
[73] Z.C. Lipton, The mythos of model interpretability, Queue 16 (2018) 31–57, https://doi.org/10.1145/3236386.3241340.
[74] C. Molnar, Interpretable machine learning, a guide for making black box models explainable, URL https://christophm.github.io/interpretable-ml-book/, (2019) , Accessed date: 27 February 2020.
[75] T. Miller, Explanation in artificial intelligence: insights from the social sciences, Artif. Intell. 267 (2018) 1–38, https://doi.org/10.1016/j.artint.2018.07.007.
[76] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining, August 22–27, San Francisco, California, Association for Computing Machinery, New York, NY, USA, 2016, pp. 1135–1144, , https://doi.org/10.1145/2939672.2939778.