NUMERICAL INVESTIGATION ON MULTICLASS PROBABILISTIC CLASSIFICATION OF DAMAGE LOCATION IN A PLATE STRUCTURE

Rims Janeliukstis, Sandris Rucevskis, Andrejs Kovalovs, Andris Chate

Institute of Materials and Structures, Riga Technical University,
Latvia, Riga, Kipsalas street 6
E-mail of the corresponding author: Rims.Janeliukstis_1@rtu.lv

ABSTRACT

The present study is devoted to the problem of damage localization by means of data classification. Commercial finite elements program ANSYS is used to make a model of a cantilevered composite plate equipped with 11 strain sensors. The plate is divided into 18 zones and for data classification purposes each of these zones houses 9 points at which a point mass with a magnitude of 5 % and 10 % fraction of plate mass is applied. At each of these points a numerical modal analysis is performed in which first 4 natural frequencies and 11 strain reading is extracted for each point. Point mass, similar to damage, causes local changes of stiffness. The data of strain for every point is an input for classification procedure involving 2 methods – *k* – nearest neighbors and decision trees. Classification model is trained and optimized by fine-tuning the key parameters for both algorithms. Finally, 2 new query points are simulated (by applying the point mass) and subjected to classification in terms of assigning a label of one of 18 zones of the plate, thus localizing these points in terms of one of 18 zones. Damage localization results are compared for both algorithms and are in good agreement with the actual positions of application of point load.

KEYWORDS: classification, plate, damage, data, decision tree, nearest neighbors.

1. INTRODUCTION

Data classification algorithms, such as decision trees, *k*-nearest neighbors, Naïve Bayes, support vector machines and other hold a potential to be applied in damage detection methodologies based on relevant feature extraction from vibration signals of monitored structures. Decision trees along with the *k*-nearest neighbor methods are one of the most widely used classification techniques. Numerous examples of successful fault detection of rotating members in automotive and electrical engineering industry can be found in literature [1-5]. Decision tree classification was performed to extract the statistical features of vibration signals of hydraulic brake system of a commercial automobile [1], while in [3] the test object was a mono block centrifugal pump. In [2] the vibration signals were collected from machining tools and subjected to statistical feature extraction using principle component analysis and decision trees for service life prediction. *k*-nearest neighbors were applied to vibration signals of bearings in electric traction motors to detect and classify the type of degradation [4] and the problem of fault detection in induction motors employing a pattern recognition of current and voltage signatures by *k*-nearest neighbors was tackled in [5].

Decision trees have been widely used in damage prediction for civil engineering applications, such as reinforced concrete buildings [6] exposed to seismic risk where the statistical damage classification is necessary to discern the buildings in need for retrofitting [7]. Mechbal et al. in [8] proposed to use multiclass support vector machines in conjunction with decision tree technique to obtain posterior probabilities of existence, as well as a location of damage in composite plate. Artificial damage of different severities was simulated and applied in different positions of the plate. The proposed method proved to successfully locate the damage in most cases.

The present study, inspired by the results in [8] strives to employ a damage classification technique to eventually locate the damage in a composite plate. A numerical model of a cantilevered plate is created. Plate is divided into 18 zones which are input as class labels for damage localization based on data classification. In each of these zones 9 points are considered where a point mass of 5 % and 10 % fraction of plate’s mass is applied. Plate is equipped with 11 strain sensors and for every loading event the strain is recorded. Modal analysis is conducted and mechanical strain is collected from all sensors along with the first 4 natural frequencies. The collected strain data serves as a training database for models based on k-nearest neighbors and decision trees. Eventually, damage is simulated in 2 unknown positions and is passed to trained classification models to find its location in terms of zones along with respective localization probabilities for every zone.

2. DAMAGE LOCALIZATION METHODS

2.1. Numerical model

A cantilevered carbon fiber reinforced composite plate 360×100 mm is considered in this study. The laminate lay-up for the plate is [90/90/0/0/45/45/-45/-45/-45/45/0/90]$_s$. The ply thickness t = 0.1 mm, thus thickness of the plate is 2.4 mm. The elastic material properties are taken as follows: $E_x = 110\,GPa$, $E_y = 7\,GPa$, $G_{xy} = G_{yz} = 4.5\,GPa$, $\nu_{xy} = 0.33$, $\rho = 1560\ kg/m^3$. The plate is equipped with 11 strain sensors with the layout as shown in Figure 1. Location and orientation angle along X axis of the sensors are given in Table 1. It is assumed that sensors are embedded between layers 2 and 3.

Table 1 – Location and orientation of strain sensors.

Sensor No.	1	2	3	4	5	6	7	8	9	10	11
X, mm	100	100	100	180	180	240	270	270	310	310	350
Y, mm	5	50	95	5	95	50	5	95	5	95	50
Angle, ^{0}	15	0	-15	0	0	0	-45	45	0	0	90

Numerical modal analysis is carried out by using the commercial FE software ANSYS. The finite element model of the plate consists of 8-node shear-deformable shell elements. The plate is divided into 72×20 elements and the clamped boundary conditions are applied at one of the edges of the plate. Damage is simulated as a pseudo defect – an artificial mass with 5 % and 10 % fractions of plate's mass is placed at selected nodes of the plate, thus giving rise to local stiffness changes. Additional mass is applied by using MASS21 finite element. The modal analysis with block Lanczos mode-extraction method is applied to determine eigenfrequencies and eigenmodes.

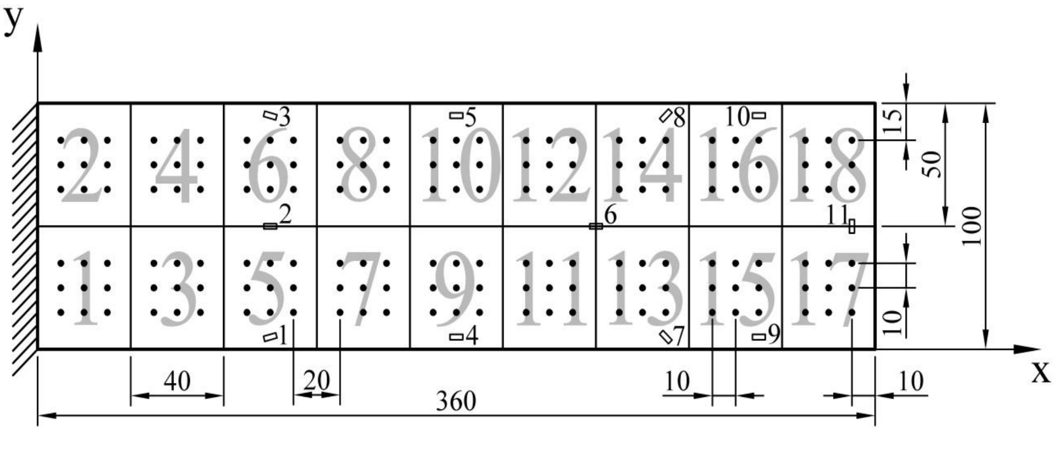


Figure 1 – Scheme of a cantilevered composite plate equipped with sensors (all dimensions in mm).

Damage is simulated as a pseudo defect – an artificial mass with 5 % and 10 % fractions of plate's mass is placed at selected nodes of the plate, thus giving rise to local stiffness changes. A numerical modal analysis is conducted and first 4 natural frequencies along with strain values from all sensors are extracted. A partition of plate into 18 zones, according to Figure 1 is applied – these zones serve as class labels for data classification procedure. 9 points are chosen to represent each zone in order to build a family of data corresponding to each class label for classification purposes. For each zone of the plate the artificial mass is placed at each of these 9 points and modal response is calculated, thus yielding 18 x 9 = 162 data sets, each comprising of 11 strain values and 4 natural frequencies. Thus a matrix of 162 modal response points x 11 strain values is used as predictor values in this study.

2.2. Data mining and training

Generally, the whole dataset can be divided into 3 parts [9]:

- *Training data* which is used to build classifiers;
- *Validation data* to optimize parameters of classifier;
- *Testing data* which was not used in the formation of the classifier; it is used to calculate the error rate of the final, optimized model to predict the performance of the classifier on a new data.

In general, the larger the training sample, the better the classifier, although the returns begin to decrease once a certain amount of data is exceeded. Also, the larger the test sample, the more accurate the error estimate.

2.2.1. k-nearest neighbors algorithm

The nearest-neighbor method was first used by statisticians in early 1950's. In 1960's it was adopted as a classification scheme and since then has been widely used in pattern recognition [9]. *k*-nearest-neighbor algorithm is a form of learning where training instances are stored and each new instance is tested on resemblance to the existing instances through the means of a distance metric. This new instance is labelled as one of the classes of instances based on the closest distance to an instance (nearest neighbor) of the same class. If more than one nearest neighbor is used, then the majority class of the closest *k* neighbors is assigned to the new instance [9].

2.2.2. Decision trees algorithm

Nodes in a *decision tree* involve testing a particular attribute, usually this attribute is compared with a constant. Leaf nodes give a classification that applies to all instances that reach the leaf. To classify an unknown instance, it is routed down the tree according to the values of the attributes tested in successive nodes and when a leaf is reached the instance is classified according to the class of a leaf.

If the attribute that is tested at the node is numeric, this test determines whether its values is greater or less than some constant, giving a 2-way split (*binary trees*).

Each leave contains a numeric value that is the average of all the training set values to which the leaf applies. Decision trees that predict numeric quantities are called *regression trees* [9].

2.2.3. Validation and performance metrics

The parameters of a classifier are optimized through validation of a classification model. Several validation techniques are available and cross-validation is used in this work. In *cross-validation* one decides on a fixed number of folds to partition the data in. For *K* number of folds, the data is split into *K* approximately equal partitions and each in turn is used for testing while the remainder is used for training. The whole procedure is repeated *K* times so that every instance has been used exactly once for testing. It is *K*-fold cross-validation [9]. The standard approach is to apply 10 folds, although it is debatable whether this number fits all cases. In present study the selection of number of folds for K-fold cross-validation is based on error estimates as explained in Section 3.

In this study, various analyses are performed in order to improve the classification accuracy of classifiers. Namely, two accuracy metrics are considered – *resubstitution loss* which is a fraction of misclassifications over all set of instances on the training data from the predictors of classification model and *cross-validation loss* which is an average loss of each cross-validation model when predicting on data that is not used in training [10]. Resubstitution loss is calculated by resubstituting the training instances into a classifier that was constructed from them [9].

There are four different outcomes of classification [9]:

- *true positives* (TP) – data is correctly classified as positive;
- *true negatives* (TN) – data is correctly classified as negative;
- *false positives* (FP) – outcome is incorrectly classified as positive when it is actually negative;
- *false negatives* (FN) - outcome is incorrectly classified as negative when it is actually positive;

TP and FP are correct classifications, while classifications FP and FN are incorrect.

Overall success rate (SR) is computed as

$$SR = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

And error rate is equal to 1- success rate.

In multiclass prediction the result in a test set is displayed as a 2D *confusion matrix* with a row and column for each class. Each matrix element shows the number of test samples for which the actual class is the row and predicted class is the column. Good results correspond to large numbers down the main diagonal and, ideally zero, off-diagonal elements [9].

ROC (receiver operating characteristics) *curves* are used to characterize the trade-off between hit rate and false-alarm rate. They plot the TP rate *tp* on vertical axis vs FP rate *fp* on horizontal axis, all expressed in %. ROC curves depict the performance of a classifier. The area under the ROC curve (AUC) represents the probability that the classifier ranks a randomly chosen positive instance above a randomly chosen negative one. The closer AUC value is to 1, the better the model [9].

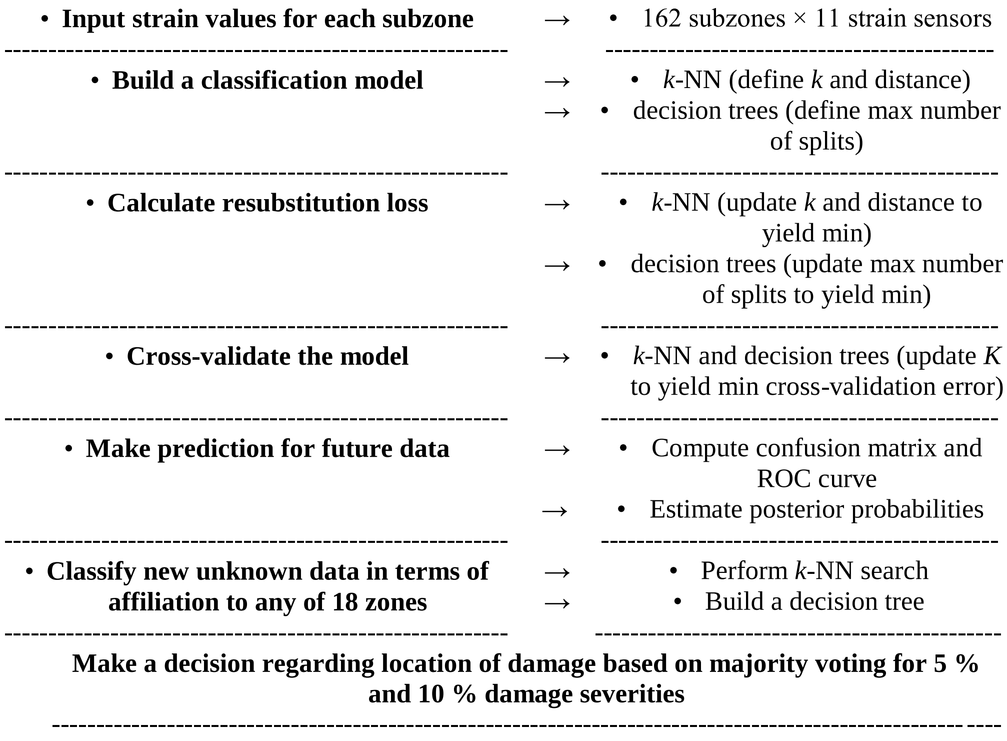
$$tp = \frac{TP \times 100\%}{TP + FN}, \qquad fp = \frac{FP \times 100\%}{FP + TN} \tag{2}$$

Specific parameters for each of methods have to be considered. For example:

- resubstitution loss is influenced by number of nearest neighbors and type of distance metric, which is used to calculate a distance between points in feature space and a point whose class is yet to be determined for k-NN method and by the number of maximum node splits for decision tree method;
- cross-validation loss is affected by number of data partitioning folds for every classification method.

Optimization of these parameters is imperative for successful classification of data.

Damage classification procedure is summarized in block diagram in Figure 2. As a point mass with severities of 5 % and 10 % of plate mass is applied in each of 162 points, the strain data from 11 sensors is collected and passed to the classification scheme. In classification procedure, a matrix consisting of 11 columns of predictor values (strain signals), each corresponding to 162 rows of class labels (zone points on the plate) is considered. 9 points are considered for every zones in order to gather more data for every zone, thus it is easier for a classifier to separate the different zones from one another. A classification model based on *k*-nearest neighbors (*k*-NN) and decision trees is built with some preliminary parameter values. These models are later optimized by minimizing the resubstitution and cross-validation errors through fine-tuning the parameters (number of nearest neighbors *k* and type of distance metric for *k*-NN and tree depth for decision trees) and the number of cross-validation folds for both methods. Once the optimization of models is complete, their performance is evaluated through computation of ROC curves and confusion matrices. Finally, the localization of damage is achieved by simulation of two new data points through the addition of point load of 5 % and 10 % fraction of plate mass and passed to the trained classifiers in order to assign them a class label – one of 18 zones of the plate in order for this damage to be localized.



• Input strain values for each subzone → • 162 subzones × 11 strain sensors

• Build a classification model → • *k*-NN (define *k* and distance) → • decision trees (define max number of splits)

• Calculate resubstitution loss → • *k*-NN (update *k* and distance to yield min) → • decision trees (update max number of splits to yield min)

• Cross-validate the model → • *k*-NN and decision trees (update *K* to yield min cross-validation error)

• Make prediction for future data → • Compute confusion matrix and ROC curve → • Estimate posterior probabilities

• Classify new unknown data in terms of affiliation to any of 18 zones → • Perform *k*-NN search → • Build a decision tree

Make a decision regarding location of damage based on majority voting for 5 % and 10 % damage severities

Figure 2 – Block scheme of the damage localization based on data classification algorithms.

3. RESULTS

During the modal analysis procedure, 4 natural frequencies of the cantilevered composite plate are computed for all 162 mass application points on the plate. A spatial frequency distribution is calculated and for the 1st frequency is shown in Figure 3. The statistical measures for all frequencies are shown in Table 2. As expected, the lowest frequency variation is observed at the clamped end of the plate due to restriction of movement.

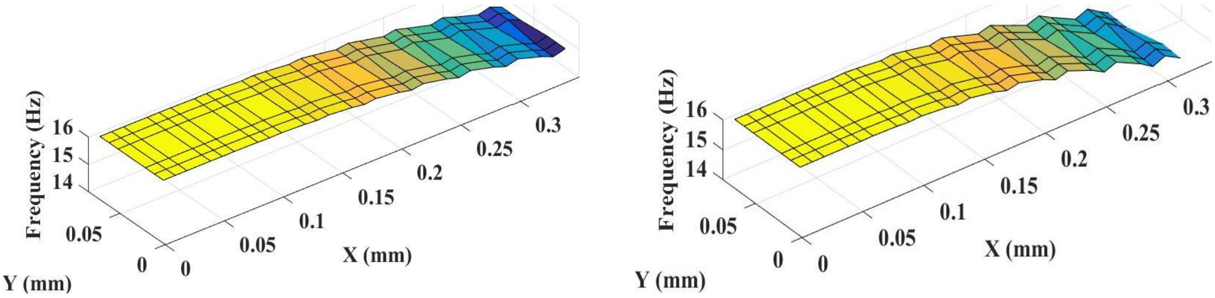


Figure 3 – Spatial distribution of **1st** resonant frequency values for different damage severities. Left: 5 %, right: 10 %.

Table 2 – Statistical measures of spatially distributed resonant frequencies for different damage severities.

	5 % mass				10 % mass			
Frequencies	1st	2nd	3rd	4th	1st	2nd	3rd	4th
Maximum (Hz)	15.84	96.55	100.16	276.79	15.84	96.55	100.16	276.78
Minimum (Hz)	14.56	86.03	96.55	254.90	13.54	79.99	96.55	237.08
Average (Hz)	15.49	94.01	98.95	270.37	15.18	91.27	98.76	264.69
Standard deviation (Hz)	0.40	2.33	1.30	5.40	0.73	4.04	1.36	10.00
Coefficient of variation (%)	2.58	2.48	1.31	2.00	4.80	4.43	1.38	3.78

3.1. Minimization of classification error

The success of data classification heavily relies on selection of optimum parameter values for each classification scheme. Misclassification of data is indicated by resubstitution and cross-validation error, each of which are affected by definite parameters of classifiers.

3.1.1. Resubstitution error

For k-NN algorithm these parameters include the number of nearest neighbors (k) and the type of distance metric. Generally, the standard Euclidean distance is used which, however assumes that the attributes of instances are equally important [9]. In present study, however, *Chebychev distance* metric is adopted as it gives the smallest resubstitution error. Chebychev distance d between vectors x_s and y_t is defined as follows [11]:

$$d_{st} = max_j\{|x_{sj} - y_{tj}|\} \quad (3)$$

It is found that the optimum number k for nearest neighbors in our case should be 3 for both severities of damage as it yields no resubstitution error. The key parameter for data classification with decision trees is tree depth which is characterized by number of node splits. Figure 4 shows the resubstitution error with respect to maximum number of node splits. IT is seen, that this error decreases with the increasing complexity of the tree, reaching minimum at 20 nodes and giving an error of 1.23 % for both severities of damage.

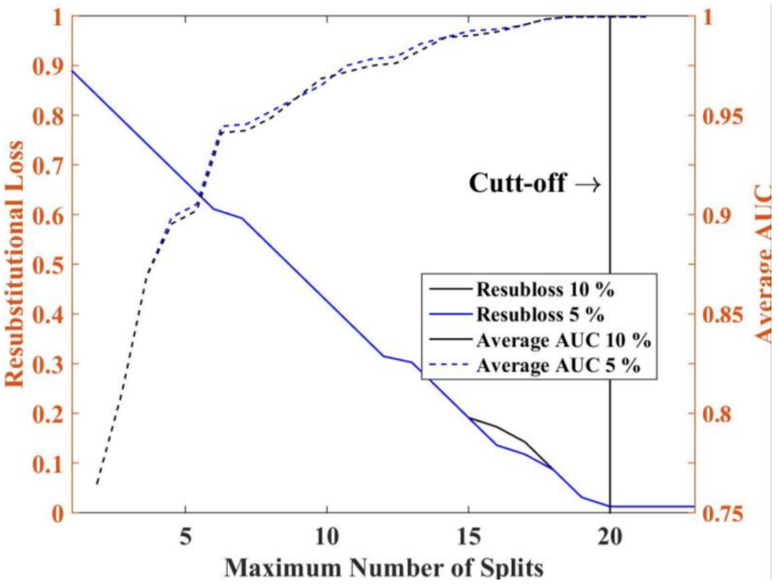


Figure 4 – Resubstitution loss vs maximum number of leaf splits for 5 % and 10 % damage severity.

3.1.2. Cross-validation error

In this study, *K*-fold cross-validation scheme is adopted to validate the classification model. In order to select the appropriate number of folds (*K*), it is decided not to rely on the classical approach for *K* = 10 but instead perform an analysis of how the variation of *K* influences the cross-validation error. These results for both damage severities (5 % and 10 %) are shown in Figure 5. The values for cross-validation error are calculated for discrete values of *K* as integers obtained from division of total number of features (162) by integer values. Therefore, *K* is assigned the values of 2, 3, 6, 9, 18, 27, 54, 81 and 162. The minimum possible number of folds is 2, as at least 1 fold must be reserved for testing the data. As it can be seen, this error drops significantly with increasing number of partitioning folds and the minimum error of cross-validation for *k*-NN algorithm 0.62 % corresponds to number of folds *K* = 9 for both severities of damage. Thus 9 folds are selected for optimization of *k*-NN model, while *K* = 27 folds are selected for decision tree scheme for 5 % and 10 % damage severities, leading to cross-validation error of 16.67 % and 15.43 %, respectively.

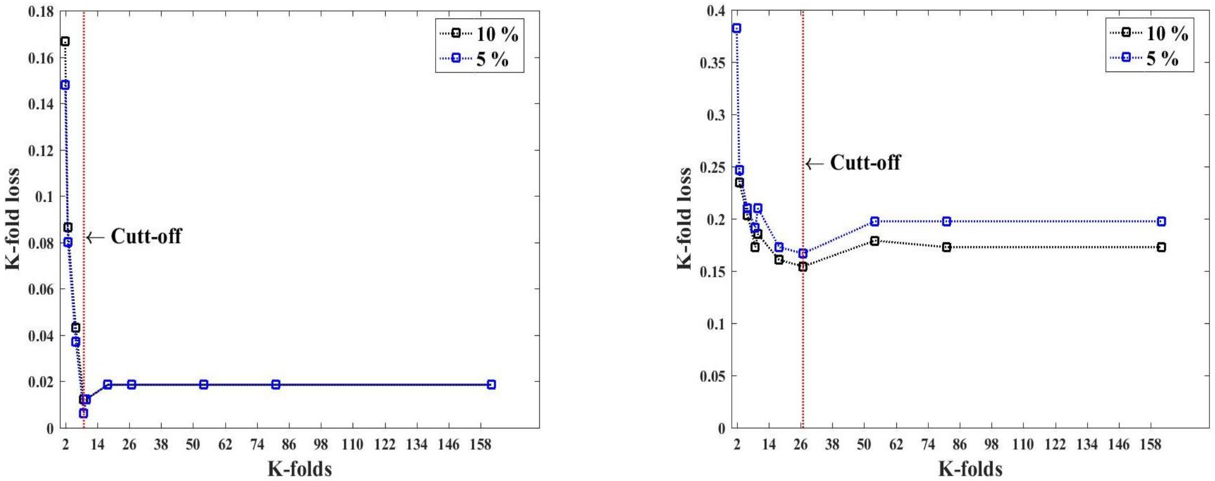


Figure 5 – *K*-fold loss vs number of *K*-folds. Left: *k*-NN, right: decision trees.

All information regarding error assessment for both algorithms is shown in Table 3 and Table 4.

Table 3 – Loss assessment for *k*-NN algorithm.

Damage severity	10 %	5 %
Number of *K*-folds	9	9
K-fold loss (%)	0.62	0.62
k	3	3
Resubstitution loss (%)	0	0

Table 4 – Loss assessment for decision tree algorithm.

Damage severity	10 %	5 %
Number of *K*-folds	27	27
K-fold loss (%)	15.43	16.67
Maximum number of splits	20	20
Resubstitution loss (%)	1.23	1.23

3.2. Evaluation of classification performance

In this study, two tools are used to measure the quality of classification:

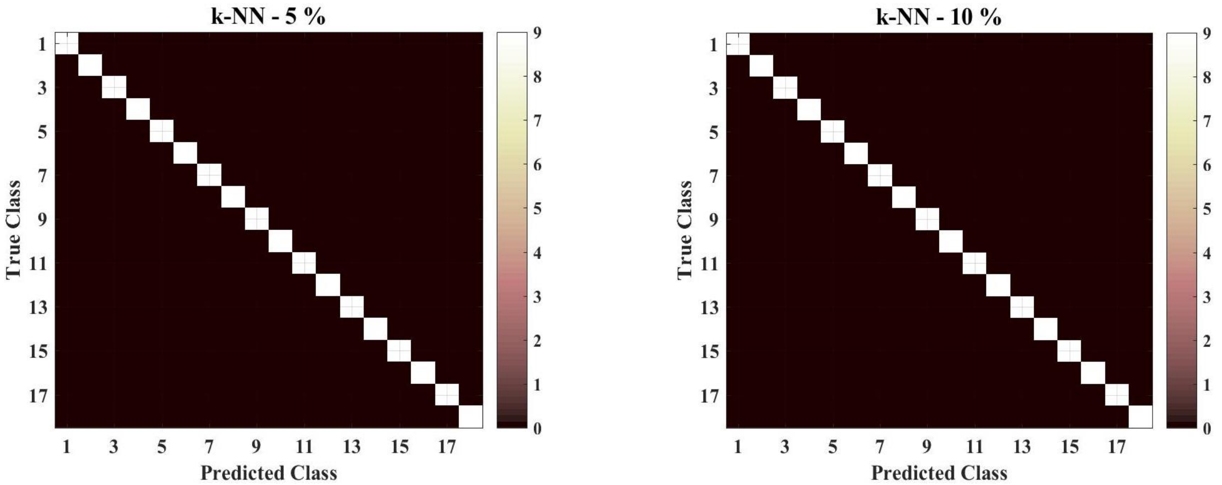
- confusion matrices;
- ROC curves.

3.2.1. Confusion matrices

In our case, confusion matrices consist of 18x18 elements (real 18 zones of the plate x predicted 18 zones of the plate). For a match between each pair of classes a maximum value of 9 can be reached, meaning that all 9 mass application points are predicted correctly for each of 18 zones of the plate. A perfect classification is achieved using *k*-NN algorithm – for both damage severities predicted class is completely matched with a true class for both severities of damage. As for decision trees, a slight misclassification is attributed to classes no. 2 and 4 where 1 of 9 subzones is classified as belonging to class (zone) no. 1, when it is actually no. 2 and 1 of 9 subzones is attributed to class no. 3, when it is actually no. 4. These results are depicted in Figure 6 for both classification algorithms for both damage severities.

3.2.2. ROC curves

ROC curves are computed for each of 18 classes (zones of the plate). Also, the average values for Area Under Curve (AUC) is given, accounting for all 18 classes. AUC is a useful metric of classifier performance as it is independent of the decision criterion selected and prior probabilities and it does not depend on the imbalance of the training set [12]. As one can see in Figure 7, average AUC is equal to 1 for *k*-NN based classification, indicating a perfect classification. Whereas for decision trees this value is about 0.9995 for both severities of damage. This fact is confirmed by analysis of confusion matrices; AUC values for all classes are 1, except for classes no. 1, 2, 3 and 4. For these classes AUC = 0.9978, suggesting only a very slight misclassification.



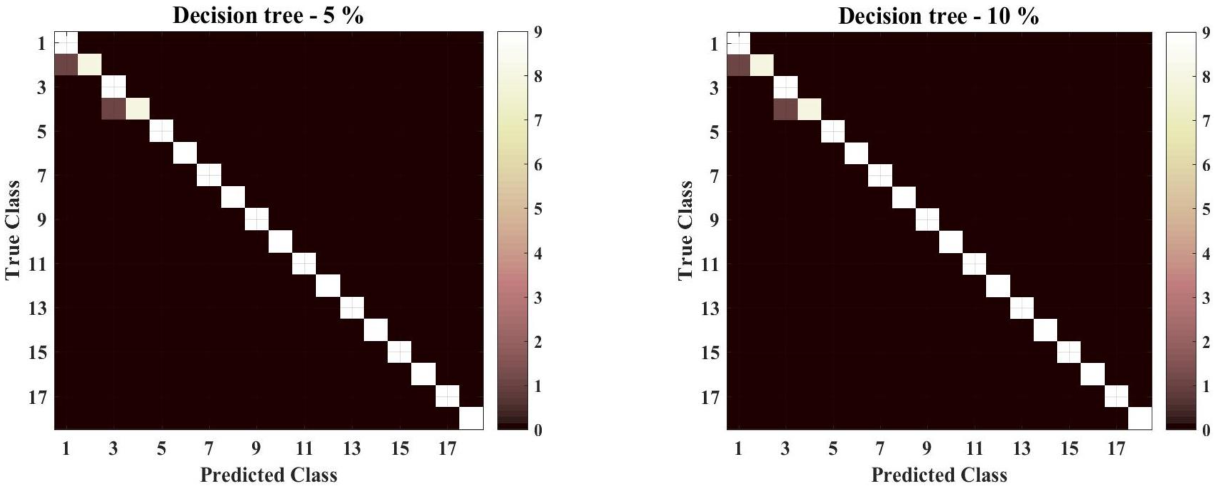


Figure 6 – Confusion matrices for *k*-NN and decision tree algorithms. Left: 5 % damage, right: 10 % damage.

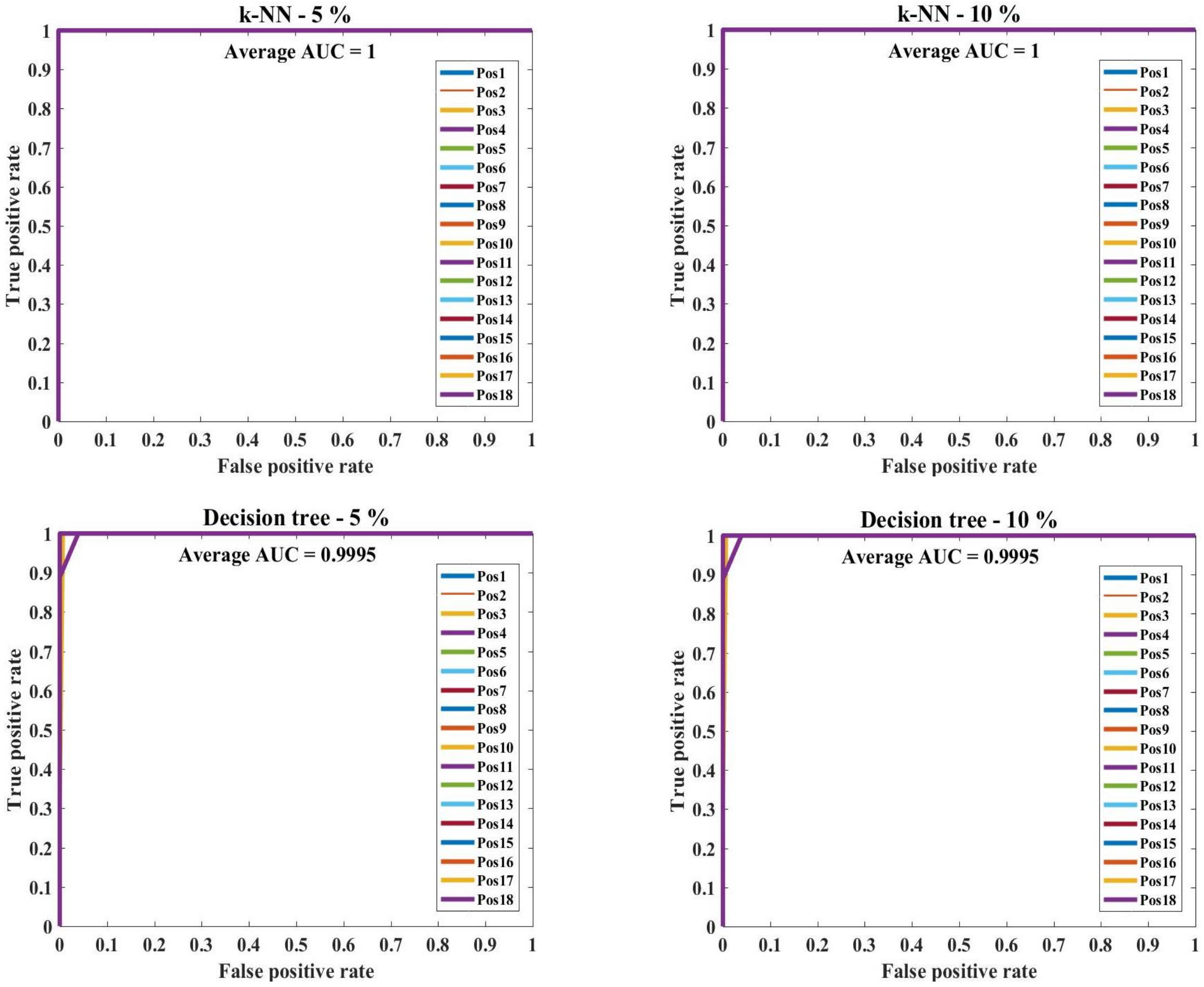


Figure 7 – ROC curves for *k*-NN algorithm. Left: 10 % damage, right: 5 % damage.

3.3. Damage localization

The presents study is focused on developing a methodology of damage localization based on data classification. After the model of classification is trained, validated and optimized, new data is assigned to belong one of the class labels (one of 18 zones of the composite plate). For this purpose, the damage is simulated by the application of artificial point mass at 2 points of unknown coordinates (in the range of plate dimensions) and this information is passed to already trained *k*-NN and decision tree models. The coordinates of these points are shown in Table 5 for both damage cases.

Table 5 – Coordinates of new points subjected to classification for damage localization.

Damage severity 10 %				Damage severity 5 %			
X_1	0.34	Y_1	0.005	X_1	0.13	Y_1	0.035
X_2	0.2	Y_2	0.05	X_2	0.32	Y_2	0.07

3.3.1. Localization with k-NN

k- nearest neighbor search is performed, involving the following steps:

- selection of predictor pairs that yields the best separation of class groups – strain data of sensor 3 with respect to sensor 2 is selected to build scatter plots, so that the 18 classes (zones of the plate) are well separated with 9 points (subzones in each zone) for each class;
- the query points from Table 5 are plotted in this domain of feature space;
- query points are surrounded by a circle denoting a radius of distance metric so that the selected number of nearest neighbors (3) is inside the circle;
- these query points are assigned a class based on majority voting – the class of majority of 3 nearest points (either 3 out of 3 or 2 out of 3);

The scatter plots are shown in Figure 8 along with the classification results for unknown query points for damage severities 5 % and 10 %.

The computations of classification models are performed in MATLAB. In *k*-nearest neighbor search the results for the assignment a class label to a query point are displayed in the manner shown in Table 6. The number of nearest neighbors that fall within the distance denoted by a circle is depicted for each class as long as the total number of nearest neighbors found match the set number of *k* (3 in our case). When all (3) nearest neighbors of a query point are found, the program stops classification and no further classes are considered. This is marked as “Not relevant (NR)” for these classes. Cases when all nearest neighbors correspond to the same class are marked as 3/3 or 100 % and shaded in darker green, meaning 100 % confidence that the query point under consideration belongs to that particular class (zone of the plate). If, on the other hand, the query point is assigned a class based on 2/3 nearest neighbors or 66.67 %, this is marked as light green because of the majority voting. If only 1/3 nearest neighbors are found in the particular class, this class is not assigned to the query point and is dismissed (indicated by orange shading). The results are as follows:

- for 5 % damage severity – the first query point belongs to zone no. 7 and the second query point – most likely to zone no. 16;
- for 10 % damage severity – the first query point belongs to zone no. 17 and the second query point – most likely to zone no. 10.

Table 6 – Damage localization results for *k*-NN algorithm.

Class		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		Count																	
5 %	QP1	0	0	0	0	0	0	3/3	NR										
	QP2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2/3	1/3	NR
10 %	QP1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3/3	NR
	QP2	0	0	0	0	0	0	0	0	0	2/3	1/3	NR						

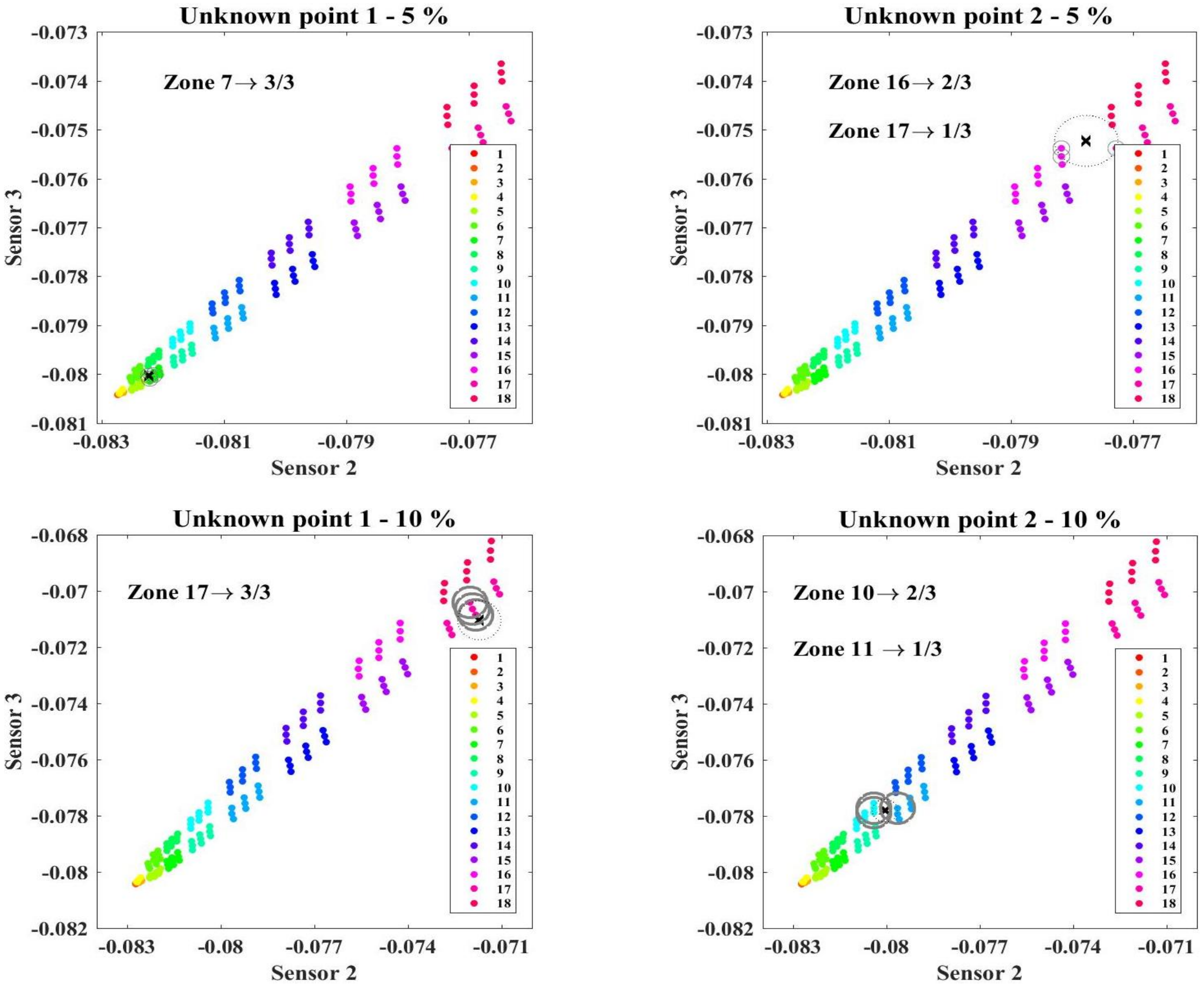


Figure 8 - *k*-NN search for 2 query points.

3.3.2. *Localization with decision trees*

First, a decision tree classifier is trained and these decision tree structures are shown in Figure 9. Decision trees are rather complex with a total of 41 nodes, consisting of 21 *leaf nodes* which contain information about the label class label and 20 *branching nodes* where the decision about the root down the tree from top to bottom is made based on the result of inequality involving predictor values (in our case xi) is made by examining the inequality relations for both severities of damage. The letters “xi” denote the numeration of strain sensor data. The numeration of nodes starts at the top node, also called the *root node* and proceeds in direction from left to right.

The result of query point (Table 5) classification is shown in Table 7. Decision trees algorithm yields only 1 value of the class label that is assigned to the query point, meaning that the probabilities for all other classes will be 0. The results of classification can be summarized as follows:

- for 5 % damage severity – the first query point belongs to zone no. 7 and the second query point – to zone no. 18;
- for 10 % damage severity – the first query point belongs to zone no. 17 and the second query point – to zone no. 9.

Table 7 – Damage localization results for decision tree algorithm.

Class		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		Probability of detection																	
5 %	QP1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	QP2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10 %	QP1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	QP2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

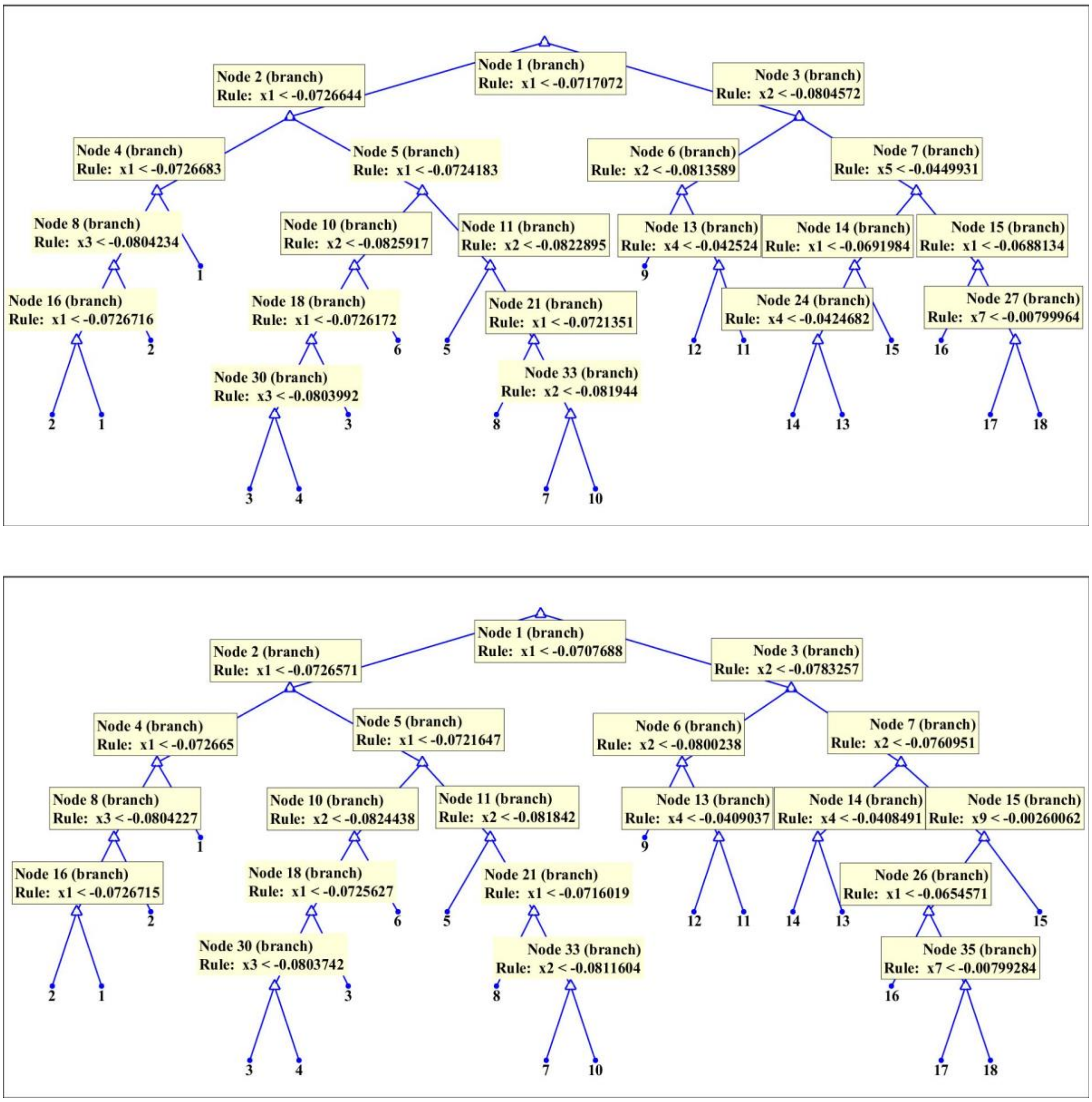


Figure 9 – Decision tree classification. Top: damage severity 5 %, bottom: damage severity 10 %.

3.3.3. Comparison of localization performance

For 5 % damage severity the agreement between performance of both algorithms is tracked in classification of the first new query point to belong to class label no. 7, while the second query point is classified somewhat differently (*k*-NN – label no. 16 and decision trees – label no. 18) with the classification difference of 2 labels. By examining a scheme of the plate (Figure 1), it is seen that zones no. 16 and 18 are neighboring ones. The width of a zone is 40 mm, meaning that the uncertainty of coordinate *x* for this result can lie anywhere from 80 mm to 0 mm (at the boundary between both zones). As it can be seen from Table 5, the coordinate *x* for a second query point is indeed 0.32 m = 320 mm which corresponds to the boundary of zones no. 16 and 18. That is why a classification algorithm can yield either label no. 16 or label no. 18 and both of these will be correct.

For 10 % damage severity both classification algorithms have classified the first query point as to belong to zone no. 17. which is correct according to Figure 1. The second query point actually lies in the intersection of 4 zones (9, 10, 11 and 12), meaning that the results of classification are equally likely for all these zones. The decision trees algorithm has classified the point as to belong to zone no. 9, whereas *k*-NN – most likely to zone no. 10, although there is some probability that is in zone no. 11. These results suggest that both classifiers performed in good agreement with one another.

4. CONCLUSIONS

In present study, the damage localization methodology for plate structures based on data classification is proposed. A numerical model of composite cantilevered plate is partitioned into 18 zones that serve as class labels in classification process. A point mass of 2 different severities (5 % and 10 % of plate's mass) is applied at 9 points for each of 18 zones to collect more data for each class. Next, the modal analysis is performed and for each event of loading a mechanical strain is recorded from 11 sensors, embedded into the plate. All strain data is collected and passed to the *k*-nearest neighbors and decision trees classifier algorithms. Classifier models are built and their parameters are optimized to minimize the resubstitution and cross-validation errors. The performance of classifiers is assessed through ROC curves with accompanying area under curve metric and confusion matrices. These metrics suggest a high quality of classification for both, *k*-nearest neighbors and decision trees. Finally, two artificial damage events through application of point mass are simulated and this information is passed to classifier algorithms to assign a class label to these query points based on trained data. It is found that there is a good agreement between the localization results of both classifiers and these results are in accordance with the actual coordinates of query points for both severities of damage (5 % and 10 %).

ACKNOWLEDGEMENTS

The research leading to these results has received the funding from Latvia State Research Program under grant agreement "Innovative Materials and Smart Technologies for Environmental Safety, IMATEH".

5. REFERENCES

[1] Jegadeeshwaran R., Sugumaran V., "Comparative study of decision tree classifier and best first tree classifier for fault diagnosis of automobile hydraulic brake system using statistical features", Measurement 46, 2013, pp. 3247-3260.

[2] Elangovan M., Babu Devasenapati S., Sakthivel N.R., Ramachandran K.I., "Evaluation of expert system for condition monitoring of a single point cutting tool using principle component analysis and decision tree algorithm", Expert Systems with Applications 38, 2011, pp. 4450-4459.

[3] Muralidharan V., Sugumaran V., "Feature extraction using wavelets and classification through decision tree algorithm for fault diagnosis of mono-block centrifugal pump", Measurement 46, 2013, pp. 353-359.

[4] Baraldi P. Cannarile F., Di Maio, F., Zio E., "Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable conditions", Engineering Applications of Artificial Intelligence 56, 2016, pp. 1-13.

[5] Casimir R., Boutleux E., Clerc G., Yahoui A., "The use of features selection and nearest neighbors rule for faults diagnostic in induction motors", Engineering Applications of Artificial Intelligence 19(2), 2006, pp. 169-177.

[6] Karbassi A., Mohebi B., Rezaee S., Lestuzzi P., "Damage prediction for regular reinforced concrete buildings using the decision tree algorithm", Computers and Structures 130, 2014, pp. 46-56.

[7] Tesfamariam S., Liu Z., "Earthquake induced damage classification for reinforced concrete buildings", Structural Safety 32, 2010, pp. 154-164.

[8] Mechbal N., Uribe J.S., Rébillat M., "A probabilistic multi-class classifier for structural health monitoring", Mechanical Systems and Signal Processing, 60-61, 2015, pp. 106-123.

[9] Witten I.H., Frank E., Hall M.A., "Data Mining: Practical Machine Learning Tools and Techniques", Third edition, Morgan Kaufman Publishers, 2011.

[10] https://uk.mathworks.com/help/stats/classificationensemble.resubloss.html

[11] https://uk.mathworks.com/help/stats/classification-using-nearest-neighbors.html#bsehylk

[12] Rokach L., Maimon O., "Data mining with decision trees: Theory and Applications", Second edition, World Scientific, 2015.