International Journal of Engineering & Technology, 7 (4.10) (2018) 190-196

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Comparative Study for Condition Monitoring on Wind Turbine Blade using Vibration Signals through Statistical Features: a Lazy Learning Approach

A. Joshuva[1], V. Sugumaran[2*]

[1] *Centre for Automation and Robotics, School of Mechanical Sciences, Hindustan Institute of Technology & Science (Hindustan University),*
Chennai 603103, India
[2] *School of Mechanical and Building Sciences (SMBS), VIT University, Chennai Campus, Vandalur-Kelambakkam Road, Chennai 600127, India*
**Corresponding author E-mail: v_sugu@yahoo.com*

Abstract

This study is to identify whether the wind turbine blades are in good or faulty conditions. If faulty, then the objective to find which fault condition are the blades subjected to. The problem identification is carried out by machine learning approach using vibration signals through statistical features. In this study, a three bladed wind turbine was chosen and faults like blade cracks, hub-blade loose connection, blade bend, pitch angle twist and blade erosion were considered. Here, the study is carried out in three phases namely, feature extraction, feature selection and feature classification. In phase 1, the required statistical features are extracted from the vibration signals which obtained from the wind turbine through accelerometer. In phase 2, the most dominating or the relevant feature is selected from the extracted features using J48 decision tree algorithm. In phase 3, the selected features are classified using machine learning classifiers namely, K-star (KS), locally weighted learning (LWL), nearest neighbour (NN), k-nearest neighbours (kNN), instance based K-nearest using log and Gaussian weight kernels (IBKLG) and lazy Bayesian rules classifier (LBRC). The results were compared with respect to the classification accuracy and the computational time of the classifier.

Keywords: *Condition monitoring; wind turbine blade; statistical features; machine learning; vibration signals.*

1. Introduction

Greenhouse gasses free power generation, rapid establishment and charging ability, low process and preservation cost and exploiting utilizing free and renewable energies are all focal points of utilizing wind turbines as a power generators. Alongside with these favorable circumstances, the fundamental hindrance of this production is the momentary wind stream. In this manner, utilizing dependable and productive equipment is essential to get as much as energy from the wind in the course of constrained timeframe that it streams emphatically. "The blade is the most significant segment in a wind turbine which currently composed by an advanced aero-dynamic science with a specific end goal to seize the maximum energy from the wind stream. Blades of horizontal axis are presently made-up of composite materials. Composite materials fulfill complex strategy requirements like, light weight and suitable stiffness, while giving great imperviousness to the static and cyclic loading [1].

Due to different natural circumstances, the wind turbine gets faulty particularly in blades. These blades are the key segment for the energy mining from the wind. To find the faults on wind turbine blades, the turbine has to shut-down and physical inspection has to be carried out. This will create a huge loss in energy production and need to spend high labour charges. If the blade damage is large, then it will create the catastrophic damage to the environment and may also damage the entire turbine structure. So to reduce the loss of productivity and labor expenditure, condition monitoring is preferred to find the damage while the turbine is in operating condition [2]."

Various studies were carried out on wind turbine blade condition to name a few, Kusiak and Verma [3] carried out work on a data-driven approach for monitoring blade pitch faults in wind turbines using SCADA data. "This study considered two blade pitch faults namely; blade angle asymmetry and blade angle implausibility and it determine the associations between them. This study was carried out using bagging, artificial neural network (ANN), pruning rule-based classification tree (PART), K-nearest neighbor (K-NN) and genetic programming (GP) algorithms. The classification accuracy of the algorithms for identifying the pitch fault on wind turbine blade was found to be GP-74.7%, Bagging-72.5%, PART-75.5%, ANN-76.2%, K-NN-73.5%.

Abouhnik and Albarbar [4] simulated crack in wind turbine blades and carried out the crack location prediction study using vibration measurements and the level of an empirical decomposed feature intensity level (EDFIL). The main drawback in empirical decomposed feature intensity level is that it very poor in performance and fault detection and moreover crack fault detection was considered and other fault parameters were neglected. Godwin and Mathews [5] carried out a study on classification and detection of wind turbine pitch faults through SCADA data and classified using RIPPER algorithm. The classification accuracy for the pitch fault was found to be 87.05%."

A study on wavelet transform based stress and time history editing of horizontal axis wind turbine blades was carried out by Pratumnopharat et al., [6]. "With wavelet transform, this method extracts fatigue damage parts from the stress-time history and generates the edited stress-time history with the shorter time length. This study uses time correlated fatigue damage (TCFD), mexican hat wavelet (Mexh), meyer wavelet (Meyr), daubechies 30th order (DB30), morlet wavelet (Morl), discrete meyer wavelet (Dmey) for the classification of crack on the blade. Here blade crack analysis was carried out. Johnson et al., [7] carried out a structural design of spars for 100m biplane wind turbine blades by beam finite elements with a cross sectional analysis. This paper mainly focuses on the wind turbine blade design.

Damir et al., [8] carried out a study on numerical models for robust shape optimization of wind turbine blades using 3D geometric modeller. In this study, a computational framework for the shape optimization of wind turbine blade was developed for variable operating conditions specified by local wind speed distributions. This study focused on the blade design using simulation process and fault parameters which affects the performance of the wind turbine was not considered. A classification of operating conditions of wind turbines for a class-wise condition monitoring strategy study was done by Jong et al., [9]. This paper presents a general method that can be used to classify the operating conditions of wind turbine in terms of rotor speed and power. This study used empirical probability density functions based method and Gaussian mixture model (GMM) based method. This paper presents performance evaluation of the proposed class-wise condition monitoring strategy using vibration signals."

Numerous works were carried out using simulation analysis; however only few experimental analyses were performed for wind turbine blade condition monitoring. "Machine learning technique was considered for wind turbine blade fault diagnosis; however, the usage was limited in literature [10]. A very limited set of defects were considered for analysis. This is especially true in case of fault diagnosis of wind turbine blade. This study makes an attempt to find 5 (five) different blade fault conditions by applying machine learning approach and statistical analysis. Fig. 1 shows the methodology of the work done. The contribution of the present study,

- This study considers five faults (blade crack, erosion, hub-blade loose connection, pitch angle twist and blade bend) for wind turbine blade fault diagnosis.
- Statistical feature extraction tool was used to extract the required features from the vibration signals.
- J48 decision tree algorithm was used for feature selection.
- This problem is modeled as a multiclass classification problem and attempts to classify using machine learning classifiers like K-star (KS), locally weighted learning (LWL), nearest neighbour (NN), k-nearest neighbours (kNN), instance based K-nearest using log and Gaussian weight kernels (IBKLG) and lazy Bayesian rules classifier (LBRC).

The rest of the paper is organized as follows. In section 2, the experimental setup and experimental procedure are explained. Section 3 presents the feature extraction process using statistical analysis. The feature selection using J48 decision tree algorithm is presented in section 4." In section 5, the classifiers used in the study are explained in detail. The results obtained from the classifiers and the discussions about their performance are presented in section 6. Conclusions are presented in the final section (section 7).

2. Experimental Studies

The main aim of this study is to identify whether the blades are in good condition or in defective condition. "If it is defective, then the objective is to deduce the condition of fault. The data used in the present study are same as one used in Joshuva and Sugumaran (2017) [11]. The experimental setup, fault simulation and experimental procedure are explained in detail in [11]. The sampling frequency used in the study was 12 KHz and each signal (sample) has a length of 10000 data points. Accelerometer was used along with data acquisition system for acquiring data. For each condition of the wind turbine blade, 100 samples were taken.

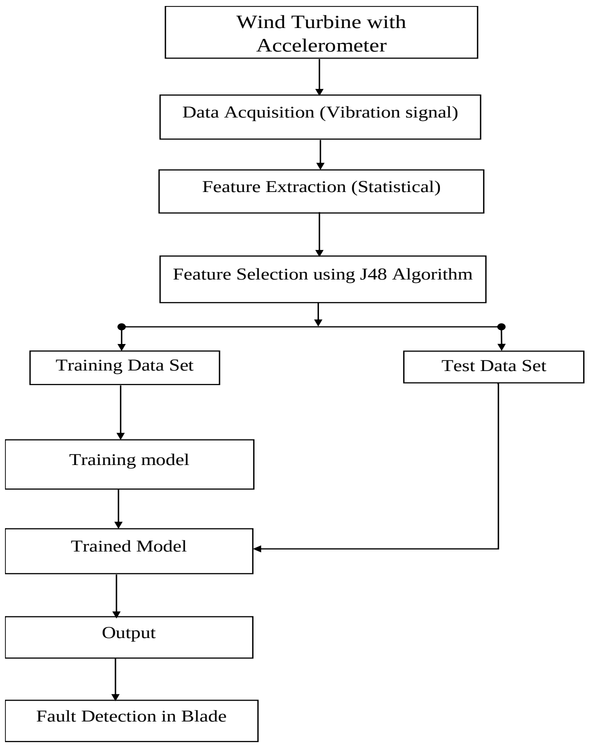
Wind Turbine with Accelerometer
Data Acquisition (Vibration signal)
Feature Extraction (Statistical)
Feature Selection using J48 Algorithm
Training Data Set
Test Data Set
Training model
Trained Model
Output
Fault Detection in Blade

Fig. 1: Methodology

3. Feature Extraction (Statistical feature extraction)

In this study, vibration data for various blade fault conditions were collected from data acquisition system (DAQ). Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Typically, it converts analog waveforms into digital values for processing. The main components of DAQ are sensors (to convert physical parameters to electrical signals), signal conditioning circuitry (to convert sensor signals into a form that can be converted to digital values) and analog-to-digital converters (to convert conditioned sensor signals to digital values). Directly vibration signals cannot be used as input to the classifier [12]. Hence, features need to be extracted using statistical methods. The process of computing some measures which will represent the signal is called feature extraction. Statistical methods for vibration signals yields different parameters such as sum, mean, median, mode, minimum, maximum, range, skewness, kurtosis, standard error, standard deviation and sample variance [13]. When the statistical feature extraction was completed, the features were taken and the feature selection method was implemented. The most contributing features are selected from the obtained statistical features using the J48 decision tree algorithm."

4. Feature Selection (J48 Decision Tree Algorithm)

Data mining techniques are being increasingly used in many modern organizations to retrieve valuable knowledge structures from databases, including vibration data [14]. "An important knowledge structure that can result from data mining activities is the decision tree (DT) that is used for the classification of future events. Decision trees are typically built recursively, following a top-down approach. A standard tree induced with C5.0 (or possibly ID3 or C4.5) consists of a number of branches, one root, a number of nodes and a number of leaves [15-17]. One branch is a chain of nodes from root to a leaf; and each node involves one attribute. The occurrence of an attribute in a tree provides the information about the importance of the associated attribute. J48 algorithm (a WEKA implementation of C4.5 algorithm) is a widely used one to construct decision trees and the detailed description of J48 decision tree algorithm and how it chooses the most dominating features were explained in Joshuva and Sugumaran (2017) [18]. The features that dominate generally represent the wind turbine blade condition descriptors. Referring to Fig. 2, one can identify four such most dominant features, (a) sum, (b) range, (c) standard deviation, and (d) kurtosis.

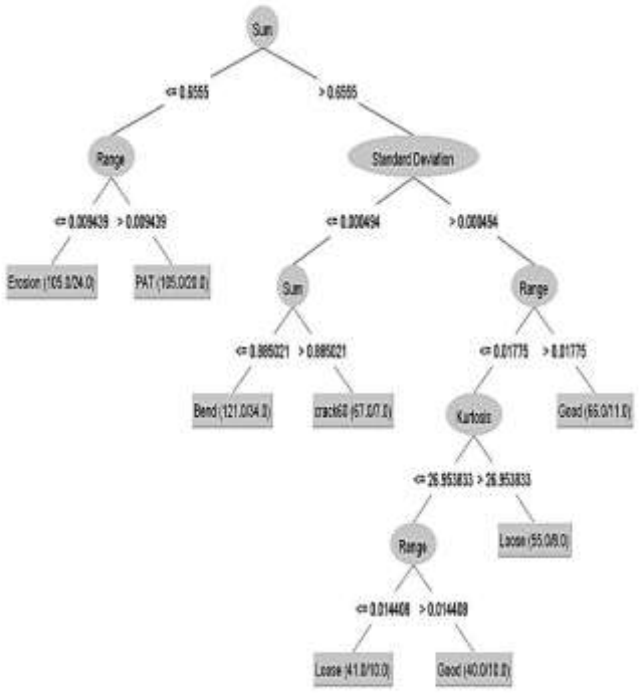

Fig. 2: J48 Tree classification for feature selection

5. Feature Classification (Lazy Classifiers)

The selected features are served as input to the classifiers. The wind turbine blade fault diagnosis was carried out using K-star (KS), locally weighted learning (LWL), nearest neighbour (NN), k-nearest neighbours (kNN), instance based K-nearest using log and Gaussian weight kernels (IBKLG) and lazy Bayesian rules classifier (LBRC).

5.1. K-Star Classifier (KS)

Lazy classifiers are valuable for data-sets with few features. They depend on 'lazy learning technique'. In lazy learning strategy, the classifier does not sum up the information set until a question is made, rather than the concerned learning technique where the training set is initially summed up. K* is an instance-based classifier, that is the class of a test example which is based upon the class of those training instances where they are, as firm by some distance function. It varies from other instance based learners wherein it utilizes entropy-based separation functions [19]. The calculation relies upon two variables, global blend and missing mode." Missing mode decides how missing property valuations are dealt with the classifier that utilizes four modes to treat missing attributes. The modes are as per the following

- Ignore the cases with missing attributes (M1)
- Normalize over the attributes (M2)
- Treat missing qualities as maximally distinctive (M3)
- Average column entropy curves (M4)

5.2. Locally Weighted Learning (LWL)

Locally weighted learning (LWL) strategies are non-parametric and the existing expectation is finished by nearby functions which utilizes just a subset of the information. "The essential thought behind LWL is that rather than building a global model for the entire function space for every point of interest or a nearby model is made taking into account of neighbouring information of the inquiry point [20]. Generally, information focuses which are in the nearby neighbourhood to the present query point accepting a higher weight than the information that has its focus further away. LWL is likewise called lazy learning since the handling of the training data is moved until a query point needs to be replied response. This approach makes LWL an extremely precise function estimation strategy where it is anything but difficult to include new training points.

5.3. Nearest-Neighbour (NN)

The Nearest-Neighbour Algorithm (NN) is an approximating algorithm for finding a sub-optimal solution for a particular problem. It can store all the training examples and it can classify the new samples by comparing them [21]. The algorithm step is as follows

- Choose a hub
- From the chosen hub, pick the circular path of least weight going along with it to another hub
- From the hub, it is linked to (not the one it started off with), discover the circular segment of slightest weight which won't make a cycle (circle) and add it to any way
- Continue in this way until the other part of the hubs are linked
- Once all part of the hubs are linked, join the first and last hub with the base circular section interfacing them, and this finishes the cycle around the system

5.4. K-Nearest Neighbours (Knn)

In pattern recognition, the k-Nearest Neighbours (kNN) algorithm is a non-parametric approach utilized for classification and regression. As a part of both cases, the information comprises the k nearest preparing samples in the feature space [22]. The output relies on whether kNN is utilized for classification or regression. In kNN classification, the output is a class membership. An item is characterized by a popular vote of its neighbours, with the item being allocated to the class most regular among its k closest neighbours (k is a positive whole number, normally small). If k = 1, then the item is essentially allocated to the class of that solitary closest neighbour."

5.5. Instance Based K-Nearest Using Log and Gaussian Weight Kernels (IBKLG)

In K-NN distance can be weighted distance such as Inverse of the distance or based on similarity. "A weight is associated to the training point based on the value of the point on a Gaussian distribution with a mean of zero. By varying standard deviation we can modify the discrepancy between weights assigned to closer and more distant training points, but generally, distances closer to zero (the mean used for the curve) result in higher weights as they are closer to the peak of the curve [23]. Distances farther from the mean fall on the tails of the distribution and therefore are assigned lower weights. The standard deviation (σ) as input for experimenting with different values for the σ hyper-parameter. For calculating logarithmic weight we used natural logarithm($\boldsymbol{log_e(x)}$) and for Gaussian we used the normal distribution formula $\boldsymbol{f(x, \mu, \sigma)} =$

$\frac{1}{\sqrt{2\sigma^2}}e^{\frac{-(x-\mu)^2}{2\sigma^2}}$,with μ (mean) set to zero and experimenting with different σ (standard deviation) values as hyper-parameters.

5.6. Lazy Bayesian Rules Classifier (LBRC)

Lazy Bayesian rules classifier (LBRC) is similar to lazy decision tree with respect to performing lazy learning of decision rules. Lazy Bayesian rules classifier also builds an individual rule that is most appropriate to the test example, using a different technique. However, whereas the consequent of a rule in lazy decision tree is a single class that is used for classification, lazy Bayesian rules classifier uses a local naive Bayesian classifier. Lazy Bayesian rules classifier can be considered as a combination of the two techniques Naive Bayes tree and lazy decision tree [24]. Like other lazy learning algorithms, lazy Bayesian rules classifier needs to keep all the training examples for use at classification time. In contrast, non-lazy learning algorithms, such as C4.5 [25] and the naive Bayesian classifier, learn a theory at training time. After the theory is learned, the training examples can be discarded. Therefore, lazy Bayesian rules classifier may have higher memory requirements than non-lazy learning algorithms at the classification stage, especially when training sets are very large."

6. Results and Discussion

From vibration signals, twelve descriptive statistical features were extracted. Out of theses twelve features, four best contributing features were selected using J48 decision tree algorithm. They are the sum, range, standard deviation, and kurtosis. "From Figure 5, the feature 'sum' is the most contributing features when compared to other features. The other contributing features are range, standard deviation, and kurtosis. The minimum number of instances per leaf and the number of data used for reduced-error pruning was kept at 50 for selecting 4 dominating features in J48 decision tree algorithm. The rest of the features like mean, median, mode, minimum, maximum, skewness, sample variance and standard error were eliminated as they contribute very less in fault classification.

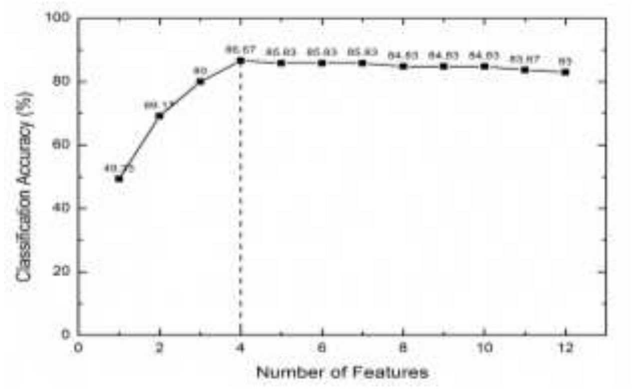

Fig. 3: Classification accuracy for number of features

In Fig. 3, the number of features vs classification accuracy is presented. The classification accuracy during the feature selection process using J48 decision tree algorithm is 86.67%. Other feature combinations did not perform well (Fig. 3). Hence, sum, range, standard deviation, and kurtosis were chosen. Then, these selected features were given as input to the classifier to determine the classification accuracy. From Figure 6, the selected features were given as the input to lazy classifiers like K-star (KS), locally weighted learning (LWL), nearest neighbour (NN), *k*-nearest neighbours (*k*NN), instance based K-nearest using log and Gaussian weight kernels (IBKLG) and lazy Bayesian rules classifier (LBRC) to determine the classification accuracy with respect to faults. For all the classifiers, the parameters like batch size (100) and the debug 'true' option was set as default.

6.1. Classification Result Using K-Star (KS)

In KS classifier, the entropy auto-blend was set as 'true'. Then the missing values of the attributes were set as 'average column entropy curves'. The global bend parameter was varied from 1 to 100 with respect to the missing modes of the attribute. Table 1, shows the results of K-star. Once the model was built, the classification accuracy for the problem using K-star (KS) classifier was found to be 88.33% for global bend parameter of 40 using average column entropy curves. From 600 samples, 530 samples were correctly classified (88.33%) and remaining 70 were misclassified (11.17%). The time taken to build the model was about 0.05s."

Table 1: Classification result of K-star

Global Bend parameter	Classification Accuracy (%)
1	81.50
10	84.50
20	86.67
30	88.00
40	**88.33**
50	87.33
60	86.17
70	85.17
80	83.67
90	81.17
100	25.83

6.2. Classification Result Using Locally Weighted Learning (LWL)

In LWL classifier, the number of neighbours used to set the kernel bandwidth (*k*NN) was fixed to be -1. The base classifier used by the locally weighted learning (LWL) was selected as default (rotation forest). "The nearest neighbour search algorithm has several approaches in LWL; they are ball tree, cover tree, filtered neighbour search, KD tree, and linear NN search. Table 2, shows the results of LWL. After the parameters were set, a model was built by the classifier for the problem. The classification accuracy for the problem identification using locally weighted learning (LWL) classifier was found to be 90.50% for rotation forest as base classifier using linear NN search. Out of 600 samples, 543 samples were correctly classified (90.50%) and remaining 57 were misclassified (9.50%). The time taken to build the model was about 0.03s.

Table 2: Classification result of LWL

Number of neighbours to use in *k*NN (*k*NN=-1)	Classification Accuracy (%)				
	Ball Tree	Cover Tree	Filter Neighbour Search	KD Tree	Linear NN Search
Rotation forest as base classifier	89.63	89.17	89.50	89.67	**90.50**

6.3. Classification Result Using Nearest-Neighbour (NN)

In NN classifier, the classifier capability check parameter was set as 'true' (to reduce run time) for the classifier. After the parameters were set, the classifier builds a model for the problem. The classification accuracy for the problem identification using Nearest-neighbour (NN) classifier was found to be 82.33%. Out of 600 samples, 494 samples were correctly classified (82.33%) and remaining 106 were misclassified (17.67%). The time taken to build the model was about 0.02s.

6.4. Classification Result Using K- Nearest-Neighbour (Knn)

In *k*NN classifier, the number of neighbours to be used in kNN was fixed (kNN=1) as default for the classifier. The distance weighting and NN search algorithm were varied for different conditions. The distance weighting method has three different types they are no distance weighting, weight by 1/distance and weight by 1-distance. The nearest neighbour search algorithm has several approaches in *k*NN; they are ball tree, cover tree, filtered neighbour search, KD tree, and linear NN search. These parameters were varied eventually by keeping one parameter constant. Table 3 shows the results of the kNN. After the parameters were set, a model for the problem was built. Once the model was built with respect to the conditions, the classification accuracy for the problem identification using Nearest-neighbour (NN) classifier was found to be 87.00% for weight by 1/distance using linear NN search. From 600 samples, 522 samples are correctly classified (87.00%) and remaining 78 are misclassified (13.00%). The time taken to build the model was about 0.03s.”

Table 3: Classification result of *k*NN

Number of neighbours to use in *k*NN (*k*NN=1)	Classification Accuracy (%)				
	Ball Tree	Cover Tree	Filter Neighbour Search	KD Tree	Linear NN Search
No distance weighting	86.67	86.17	87.17	86.67	86.17
Weight by 1/distance	85.83	87.33	86.83	86.33	**87.00**
Weight by 1-distance	86.33	86.00	86.33	86.50	86.67

6.4. Classification Result Using Instance Based K-Nearest Using Log and Gaussian Weight Kernels (IBKLG)

In IBKLG classifier, the number of neighbours to be used in kNN was fixed (kNN=1) as default for the classifier. The standard deviation to be used by the Gaussian with zero mean was fixed to be 1 as a default to the classifier. The distance weighting and NN search algorithm were varied for different conditions. The nearest neighbour search algorithm has several approaches in *k*NN; they are ball tree, cover tree, filtered neighbour search, KD tree, and linear NN search. The distance weighting method has two different types they are weight by log (distance) and weight by Gaussian (distance).These parameters were varied eventually by keeping one parameter constant. “Table 4 shows the results of the IBKLG. After the parameters were set, a model for the problem was built. The classification accuracy for the problem identification using instance based K-nearest using log and Gaussian weight kernels (IBKLG) classifier was found to be 82.33% for weight by Gaussian using linear NN search. From 600 samples, 494 samples are correctly classified (82.33%) and remaining 106 are misclassified (17.67%). The time taken to build the model was about 0.02s.

Table 4: Classification result of IBKLG

Number of neighbours to use in *k*NN (*k*NN=1)	Classification Accuracy (%)				
	Ball Tree	Cover Tree	Filter Neighbour Search	KD Tree	Linear NN Search
weight by log (distance)	81.17	81.50	81.63	81.00	81.33
weight by Gaussian (distance)	80.50	81.33	82.50	82.17	**82.33**

6.6. Classification Result Using Lazy Bayesian Rules Classifier (LBRC)

In LBRC classifier, the classifier capability check parameter was set as ‘true’ (to reduce run time) for the classifier. After the parameters were set, the classifier builds a model for the problem. The classification accuracy for the problem identification using lazy Bayesian rules classifier (LBRC) was found to be 82.33%. Out of 600 samples, 494 samples were correctly classified (82.33%) and remaining 106 were misclassified (17.67%). The time taken to build the model was about 0.02s.”

6.7. Comparative Study

From section 6.1 to section 6.6, one can find that locally weighted learning (LWL) classifier has a high classification accuracy (90.50%) compared to other classifiers. Hence, for real-time condition monitoring of wind turbine blade, locally weighted learning can be used effectively. Fig. 4 shows the comparison chart of all the classifiers used in this study. The confusion matrix of LWL is shown in Table 5. In confusion matrix, the diagonal elements represent the correctly classified instances and the others are misclassified ones. As explained earlier, for LWL, Out of 600 samples, 543 samples were correctly classified (90.50%) and remaining 57 were misclassified (9.50%).Also one can observe more misclassifications between good and loose conditions. For the loose condition, the bolts between the hub and the blade were made loose (please note that the blade was in good condition). However, at high wind speed, the blade can stick to the hub and behave like a good condition during operation. Because of this, the signature of the loose condition sometimes resembles good condition and the classifier finds difficult to distinguish between them; hence, more misclassifications.

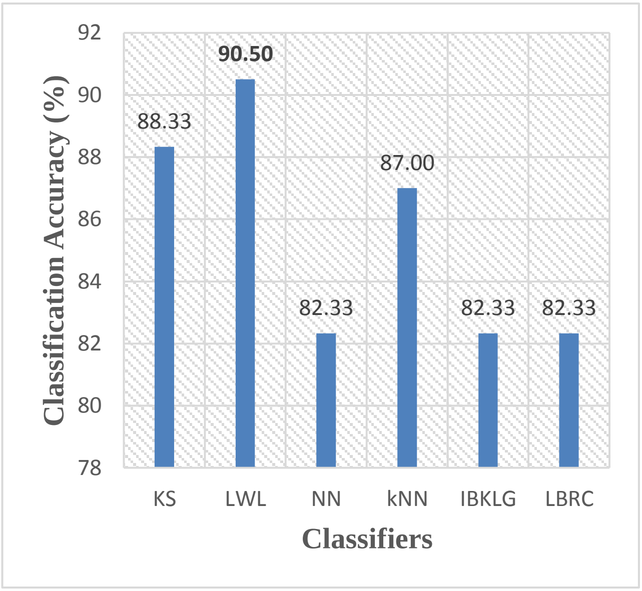

Fig. 4: Classification accuracy of the classifiers

From locally weighted learning (LWL), the kappa statistics were found to be 0.886. It is used to measure the arrangement of likelihood with the true class. “The mean absolute error was found to be 0.055. It is a measure used to measure how close forecasts or prediction are with the ultimate result. The root mean square error was found to be 0.157. It is a quadratic scoring rule which processes the average size of the error. The time taken to build the model is about 0.03s; hence, this can be used in real time for the fault detection on the wind turbine blade. The detailed class-wise accuracy is shown in Table 6. The class-wise accuracy is expressed in terms of the true positive rate (TP), false positive rate (FP), precision, recall and F-Measure.

Table 5: Confusion matrix for locally weighted learning (LWL)

Blade conditions	Good	Bend	Crack	Loose	PAT	Erosion
Good	**83**	0	0	17	0	0
Bend	0	**91**	6	0	0	3
Crack	0	2	**96**	2	0	0
Loose	13	0	6	**81**	0	0
PAT	0	0	0	0	**99**	1
Erosion	0	3	0	0	4	**93**

Table 6: Class-wise accuracy of locally weighted learning (LWL)

Class	TP Rate	FP Rate	Precision	Recall	F-Measure
Good	0.830	0.026	0.865	0.830	0.847
Bend	0.910	0.010	0.948	0.910	0.929
Crack	0.960	0.024	0.889	0.960	0.923
Erosion	0.930	0.008	0.959	0.930	0.944
Loose	0.810	0.038	0.810	0.810	0.810
PAT	0.990	0.008	0.961	0.990	0.975

TP is used to predict the ratio of positives which are correctly classified as faults. FP is commonly described as a false alarm in which the result that shows a given fault condition has been achieved when it really has not been achieved [26]. The true positive (TP) rate should be close to 1 and the false positive (FP) rate should be close to 0 to propose the classifier is a better classifier for the problem classification [27]. In LWL, it shows that the TP near to 1 and FP close to 0, hence one can conclude that the classifier built for the specific problem is effective for the fault diagnosis problem [28]. The classifier error chart is shown in Fig. 5. Here the squared dots represent the misclassification and the 'x' denotes the correct classification."

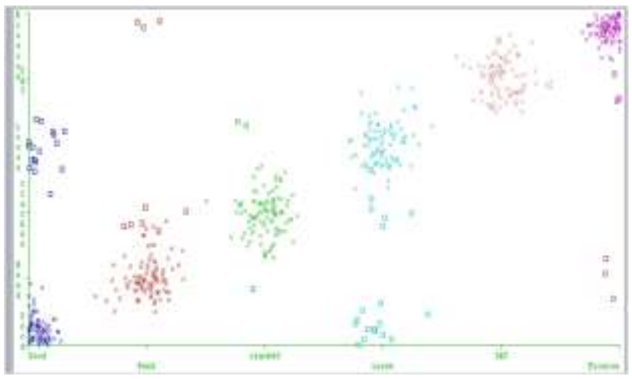
Fig. 5: Classifier Errors (Classification vs Misclassification)

7. Conclusion

The wind turbine is an important structure in extracting wind energy from the accessible wind. This paper displays an algorithm based classification of vibration signals for the evaluation of the wind turbine blade conditions. From the acquired vibration data, four models were developed using data modelling techniques. The models were tested with 10-fold cross validation. All the classifiers were compared with respect to their types and maximum correctly classified instances. The maximum classification accuracy was found to be 90.50% for locally weighted learning (LWL). The error rate is relatively less and LWL may be considered for the blade fault diagnosis. Hence, locally weighted learning (LWL) can be practically used for the condition monitoring of wind turbine blade to reduce the downtime and to maximize the usage of wind energy. The methodology and algorithm suggested in this paper can be potentially used for any kind of wind turbine blade to diagnose the blade fault with minimal modification.

References

[1] Shokrieh MM, Rafiee R, "Simulation of fatigue failure in a full composite wind turbine blade", *Composite Structures*, Vol. 74, No. 3, (2006), pp. 332-342.

[2] Chehouri A, Younes R, Ilinca A, Perron J, "Review of performance optimization techniques applied to wind turbines", *Applied Energy*, Vol. 142, (2015), pp. 361-388.

[3] Kusiak A, Verma A, "A data-driven approach for monitoring blade pitch faults in wind turbines", *IEEE Transactions on Sustainable Energy*, Vol. 2, No. 1, (2011), pp. 87-96.

[4] Abouhnik A, Albarbar A, "Wind turbine blades condition assessment based on vibration measurements and the level of an empirically decomposed feature", *Energy Conversion and Management*, Vol. 64, (2012), pp. 606-613.

[5] Godwin JL, Matthews P, "Classification and detection of wind turbine pitch faults through SCADA data analysis", *IJPHM Special Issue on Wind Turbine PHM*, (2013).

[6] Pratumnopharat P, Leung PS, Court RS, "Wavelet transform-based stress-time history editing of horizontal axis wind turbine blades", *Renewable Energy*, Vol. 63, (2014), pp. 558-575.

[7] Roth-Johnson P, Wirz RE, Lin E, "Structural design of spars for 100-m biplane wind turbine blades", *Renewable Energy*, Vol. 71, (2014), pp. 133-155.

[8] Vučina D, Marinić-Kragić I, Milas Z, "Numerical models for robust shape optimization of wind turbine blades", *Renewable Energy*, Vol. 87, (2016), pp. 849-862.

[9] Ha JM, Oh H, Park J, Youn BD, "Classification of operating conditions of wind turbines for a class-wise condition monitoring strategy", *Renewable Energy*, Vol. 103, (2017), pp. 594-605.

[10] Joshuva A, Sugumaran V, "Fault diagnostic methods for wind turbine: A review", *Asian Research Publishing Network (ARPN) Journal of Engineering and Applied Sciences*, Vol. 11, No. 7, (2016), pp. 4654-4668.

[11] Joshuva A, Sugumaran V, "Wind Turbine Blade Fault Diagnosis Using Vibration Signals through Decision Tree Algorithm", *Indian Journal of Science and Technology*, Vol. 9, No. 48, (2016).

[12] Amarnath M, Sugumaran V, Kumar H, "Exploiting sound signals for fault diagnosis of bearings using decision tree", *Measurement*, Vol. 46, No. 3, (2013), pp. 1250-1256.

[13] Mitchell TM, "Machine learning", Burr Ridge, IL: McGraw Hill. (1997).

[14] Witten IH, Frank E, "Data Mining: Practical machine learning tools and techniques", (2005).

[15] Villacampa O, "Feature Selection and Classification Methods for Decision Making: A Comparative Analysis", (2015).

[16] Sakthivel NR, Sugumaran V, Babudevasenapati S, "Vibration based fault diagnosis of monoblock centrifugal pump using decision tree", *Expert Systems with Applications*, Vol. 37, No. 6, (2010), pp. 4040-4049.

[17] Sugumaran V, Ramachandran K, "Fault diagnosis of roller bearing using fuzzy classifier and histogram features with focus on automatic rule learning", *Expert Systems with Applications*, Vol. 38, No. 5, (2011), pp. 4901-4907.

[18] Joshuva A, Sugumaran V, "A data driven approach for condition monitoring of wind turbine blade using vibration signals through best-first tree algorithm and functional trees algorithm: A comparative study", *ISA transactions*, Vol. 67, (2017), pp. 160-172.

[19] Cleary JG, Trigg LE, "K*: An instance-based learner using an entropic distance measure", InProceedings of the 12th International Conference on Machine learning, Vol. 5, (1995), pp. 108-114.

[20] Atkeson CG, Moore AW, Schaal S, "Locally weighted learning for control", InLazy learning, (1997), pp. 75-113.

[21] Gutin G, Yeo A, Zverovich A, "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP", *Discrete Applied Mathematics*, Vol. 117, No. 1, (2002), pp. 81-86.

[22] Altman NS, "An introduction to kernel and nearest-neighbor non-parametric regression", *The American Statistician*, Vol. 46, No. 3, (1992), pp. 175-185.

[23] Aha DW, Kibler D, Albert MK, "Instance-based learning algorithms", *Machine learning*, Vol. 6, No. 1, (1991), pp. 37-66.

[24] Zheng Z, Webb GI, "Lazy learning of Bayesian rules", *Machine Learning*, Vol. 41, No. 1, (2000), pp. 53-84.

[25] Quinlan JR, "C4. 5: programs for machine learning", (2014).

[26] Burke DS, Brundage JF, Redfield RR, Damato JJ, Schable CA, Putman P, Visintine R, Kim HI, "Measurement of the false positive rate in a screening program for human immunodeficiency virus infections", *New England Journal of Medicine*, Vol. 319, No. 15, (1988), pp. 961-964.

[27] Peck R, Devore JL, "Statistics: The exploration & analysis of data", (2011).

[28] Powers DM, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", (2011).