

Crack Detection and Localization on Wind Turbine Blade Using Machine Learning Algorithms: A Data Mining Approach

A. Joshuva¹ and V. Sugumaran²

Abstract: Wind turbine blades are generally manufactured using fiber type material because of their cost effectiveness and light weight property however, blade get damaged due to wind gusts, bad weather conditions, unpredictable aerodynamic forces, lightning strikes and gravitational loads which causes crack on the surface of wind turbine blade. It is very much essential to identify the damage on blade before it crashes catastrophically which might possibly destroy the complete wind turbine. In this paper, a fifteen tree classification based machine learning algorithms were modelled for identifying and detecting the crack on wind turbine blades. The models are built based on computing the vibration response of the blade when it is excited using piezoelectric accelerometer. The statistical, histogram and ARMA methods for each algorithm were compared essentially to suggest a better model for the identification and localization of crack on wind turbine blade.

Keywords: Structural health monitoring, fault diagnosis, blade crack, statistical features, histogram features, ARMA features, tree algorithms, vibration signals.

1 Introduction

As in the request on energy creation from renewable sources continually rises, modern advancements in wind turbine design are imposed by improved requirements imitated in wind turbine size increment for yielding more energy from available wind source, advanced in blade design, equipped control system advancement, and enhanced structural health and condition monitoring (SCHM). Apart from previously mentioned necessities, the expenditures of energy production must be at least similar with the expenditures of energy generation from regular sources to create wind turbine structure a mechanically satisfactory. The wind turbine machinery has benefits among other uses of renewable energy machineries due to its technological development, superior structure and comparative cost affordability [Manwell, McGowan and Rogers (2010)]. The main achievement of a wind energy development is depending on the dependability of a wind turbine structure. Inadequate dependability will right away result in the raise of operation and maintenance cost and fall in the lifetime of the wind turbine structure.

¹ Centre for Automation and Robotics (ANRO), Department of Mechanical Engineering, Hindustan Institute of Technology and Science, Padur, Chennai, 603103, India.

² School of Mechanical and Building Sciences (SMBS), VIT University, Chennai Campus, Vandalur-Kelambakkam Road, Chennai, 600127, India.

*Corresponding Author: A. Joshuva. Email: joshuva1991@gmail.com.

To increase the wind turbine system consistency, it is essential to isolate critical components and illustrate the failure modes effectively so that the attention on monitoring techniques can be initiated essentially. Failure can take place at any part of the wind turbine like, generator bearing, a bolt shears, wind turbines blades, gear box bearings, and a load-bearing brace buckles etc. As the blades are the significant components of a wind turbine structure and the cost of the blades is about 15-20% of the entire cost of the wind turbine hence, wide attention has been provided for the condition monitoring of blades. It is normally problematic to forecast the life time of a blade, however, it is feasible to predict the condition of the blade. The usage of condition monitoring has developed significantly in the past decade because of its capability to permit real-time monitoring of abilities as a means to succeed the objective of early failure detection.

There are two types of approaches which are carried out for condition monitoring: traditional approach and machine learning approach. The traditional approach is mainly used where frequency component does not change with respect to time. Rotating machines produce non-stationary signals. Since the frequency components change due to wear and tear, fault discrimination is very difficult using an automated system in the traditional approach. Hence, it is not preferred. In machine learning approach, algorithms have the capability to learn continuously and adapt themselves to the varying situations. Hence, researchers often resort to machine learning approach for fault diagnosis of mechanical systems [Joshuva and Sugumaran (2016)].

Many studies were carried out on blade crack analysis on wind turbine blade, to name a few, Barnard et al. [Barnard and Wendell (1997)] carried out a work on a simple method of estimating wind turbine blade fatigue damage at potential wind turbine sites. The keystone of this method was an easy model for the blade's root flap bending moment. The model needs as input a simple set of wind measurements which may be attained as portion of a scheduled site characterization study. By utilizing the model to simulate a time series of the root flap bending moment, fatigue damage rates have been estimated. The technique was evaluated by comparing these estimates with damage estimates derived from actual bending moment data; the agreement between the two was quite good. The simple connection between wind measurements and fatigue provided by the model allows one to readily discriminate between damaging and more benign wind environments.

A work on an integrated approach to wind turbine fatigue analysis was carried out by Laino et al. [Laino and Hansen (1997)]. In this study, a steel blade root suggests changes affecting the normal operation of the turbine alter fatigue life more than occasional, high load events. The material fatigue characteristics will affect the lifetime estimates and it was discussed in terms of the S-N curve utilized in this study. Ghoshal et al. [Ghoshal, Sundaresan, Schulz et al. (2000)] carried out a study on various structural health monitoring techniques for wind turbine blades. In this study, four different methods are tested for the damage detection on wind turbine blade they are, transmittance function, operational detection shape, wave propagation and resonant comparison.

A study of fatigue damage in wind turbine blades was carried out by Marin et al. [Marin, Barroso, Paris et al. (2009)]. In this study, superficial cracks, geometric concentrator, abrupt change of thickness have been considered. The crack was simulated in the

transition zone between the root of the blade and the zone of aerofoil profile. This study deals with the propagation of the crack on wind turbine blade using ANSYS. Abouhnik et al. [Abouhnik and Albarbar (2014)] 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.

A study on wavelet transform based stress and time history editing of horizontal axis wind turbine blades was carried out by Pratumnopharat et al. [Pratumnopharat, Leung and Court (2014)]. 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. In this study, Time correlated fatigue damage (89.82%), Mexican hat wavelet (79.23%), Meyer wavelet (79.76%), Daubechies 30th order (80.81%), Morlet wavelet (80.34%) and Discrete Meyer wavelet (80.30%) was used for the classification of crack on the blade. Bouzid et al. [Bouzid, Tian, Cumanan et al. (2015)] done a work on structural health monitoring of wind turbine blades using acoustic source localization and wireless sensor networks and obtained an error rate of 7.98% in their work.

Liu et al. [Liu, Jiang and Chu (2015)] carried out a study on the influence of alternating loads on nonlinear vibration characteristics of cracked blade in a rotor system using FEM analysis. In this study, the experiments for different alternating loads for the identification of the crack fault were performed. A work on crack diagnosis of wind turbine blades based on EMD method was carried out by Cui et al. [Cui, Ding and Hong (2016)]. This study was based on aerodynamics and fluid-structure coupling theory, an aero-elastic analysis on wind turbine blades model was first made in ANSYS Workbench. Secondly, based on the aero-elastic analysis and EMD method, the blade cracks were diagnosed and identified in the time and frequency domains, respectively. Finally, the blade model, strain gauge, dynamic signal acquisition and other equipment were used in an experimental study for the aero-elastic analysis and crack damage diagnosis of wind turbine blades was to verify the crack diagnosis method.

Numerous works were carried out using simulation analysis; however only few experimental analyses were performed for crack identification on wind turbine blade. Machine learning technique was considered for condition monitoring of wind turbine blade; however, the usage was limited in literature. This study makes an attempt for crack detection and localization on wind turbine blade by applying machine learning approach and comparing with statistical, histogram and ARMA analysis. Fig. 1 shows the methodology of the work done. The contribution of the present study:

- Crack detection and localization on wind turbine blade was carried out.
- Statistical, histogram and ARMA feature extraction tools was used to extract the required features from the vibration signals.
- J48 decision tree algorithm was used for feature selection.
- The objective was classified using machine learning classifiers.

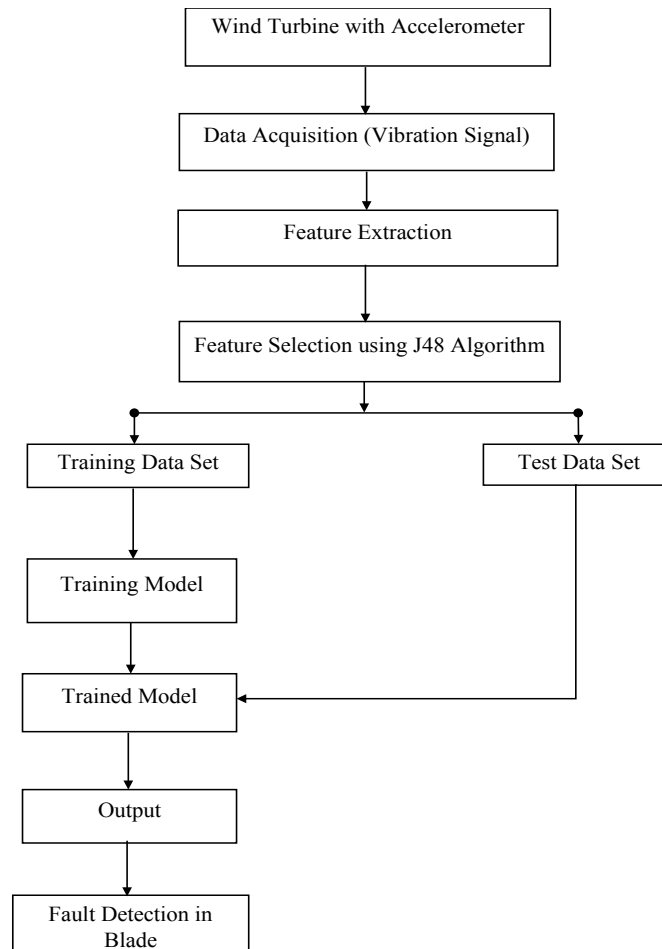


Figure 1: Methodology

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, histogram and ARMA features. The feature selection using J48 decision tree algorithm is presented in section 4. In Section 5, the machine learning classifiers 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 classify whether the blades are in good condition or in a defective state. If it is defective, then the objective is to identify the type of fault. The experimental setup and experimental procedure are described in the following subsections [Joshuva and Sugumaran (2017)].

2.1 Experimental setup

The experiment was carried out on a 50 W, 12 V variable wind turbine (MX- POWER, model: FP-50W-12V). The technical parameters of a wind turbine are given in Tab. 1.

Table 1: Technical parameters of wind turbine

Model	FP-50W-12V
Rated Power	50 W
Rated Voltage	12 V
Rated Rotating Rate	850 rpm
Start-up Wind Speed	2.5 m/s
Cut-in Wind Speed	3.5 m/s
Cut-out Wind Speed	15 m/s
Security Wind Speed	40 m/s
Rated Wind Speed	12.5 m/s
Engine	Three-phase permanent magnet generator
Rotor Diameter	1050 mm
Blade Material	Carbon fiber reinforced plastics

The wind turbine was mounted on a fixed steel stand in-front of the open circuit wind tunnel outlet. The wind tunnel speed ranges from 5 m/s to 15 m/s and act as a wind source to start the wind turbine. The wind speed was varied continuously in order to simulate the environmental wind condition. Experimental setup is shown in Fig. 2. Piezoelectric type accelerometer was used as transducer for acquiring vibration signals. It has high-frequency sensitivity for detecting faults. Hence accelerometers are widely used in condition monitoring. In this case, a uniaxial accelerometer of 500 g range, 100 mV/g sensitivity, and resonant frequency around 40 Hz was used. The piezoelectric accelerometer (DYTRAN 3055B1) was mounted on the nacelle near to the wind turbine hub to record the vibration signals using an adhesive mounting technique. It was connected to the DAQ system through a cable.

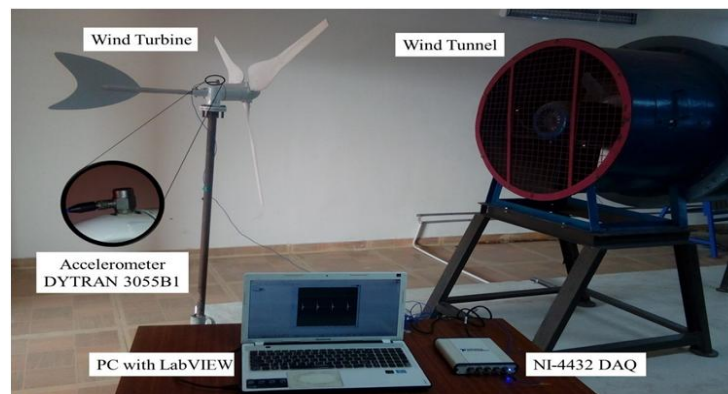


Figure 2: Experimental setup

The data acquisition system (DAQ) used was NI USB 4432 model. The card has five analog input channels with a sampling rate of 102.4 kilo samples per second with 24-bit resolution. The accelerometer is coupled to a signal conditioning unit which consists of an inbuilt charge amplifier and an analogue-to digital converter (ADC). From the ADC, the vibration signal was taken. These vibration signals were used to extract features through feature extraction technique. One end of the cable is plugged to the accelerometer and the other end to the AIO port of DAQ system. NI-LabVIEW was used to interface the transducer signal and the system (PC).

2.2 Experimental procedure

In the present study, three-blade variable horizontal axis wind turbine (HAWT) was used. Initially, the wind turbine considered was in good condition (free from defects, new setup) and the signals were recorded using the accelerometer. These signals were recorded with the following specification:

- 1. Sample length:** The sample length was chosen long enough to ensure data consistency; and also the following points were considered. Feature measures are more meaningful, when the number of samples is sufficiently large. On the other hand, as the number of samples increases the computation time increases. To strike a balance, sample length of 10000 was chosen.
- 2. Sampling frequency:** The sampling frequency should be at least twice the highest frequency contained in the signal as per Nyquist sampling theorem. By using this theorem sampling frequency was calculated as 12 kHz (12000 Hz).
- 3. A number of signal samples:** Minimum of 100 (hundred) signal samples were taken for each condition of the wind turbine blade and the vibration signals were recorded by using NI LabVIEW.

The vibration signals are acquired using DAQ. Data acquisition (DAQ) is the process of converting analog sampling signals to digital numeric values that can be manipulated by a computer. DAQ hardware is used hereto interface between the sensor signal and a PC. The following faults were simulated one at a time on a blade while other blades remain in good condition and the corresponding vibration signals were acquired. Fig. 3(a) to Fig. 3(d) show the different blade fault conditions which are simulated on the blade along and Fig. 4 shows their respective vibration signatures.



Figure 3a: Good condition blade



Figure 3b: Blade root crack fault

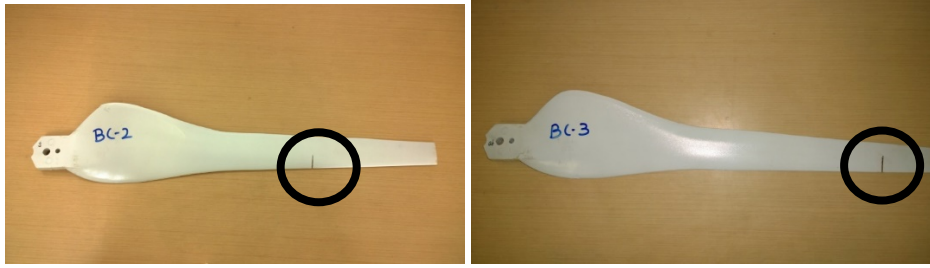


Figure 3c: Blade mid-span crack fault **Figure 3d:** Blade tip crack fault

- a) Blade good (BG): This blade is free from defects.
- b) Blade crack (BC-1): This occurs due to foreign object damage on blade while it is in operating condition. On blade, 15 mm crack was made at root of the blade.
- c) Blade crack (BC-2): On blade, 15 mm crack was made at mid-span of the blade.
- d) Blade crack (BC-3): On blade, 15 mm crack was made at tip of the blade.

Fig. 4 shows the vibration signals of the blade which were taken from different crack conditions of wind turbine blade at 850 RPM. They show the vibration signal plot (sample number vs amplitude) for good condition blade, blade root crack fault, blade mid-span crack fault, blade tip crack fault respectively. This gives some basic idea about how the magnitude of the acquired vibration signal varies over time with respect to the faults that were simulated.

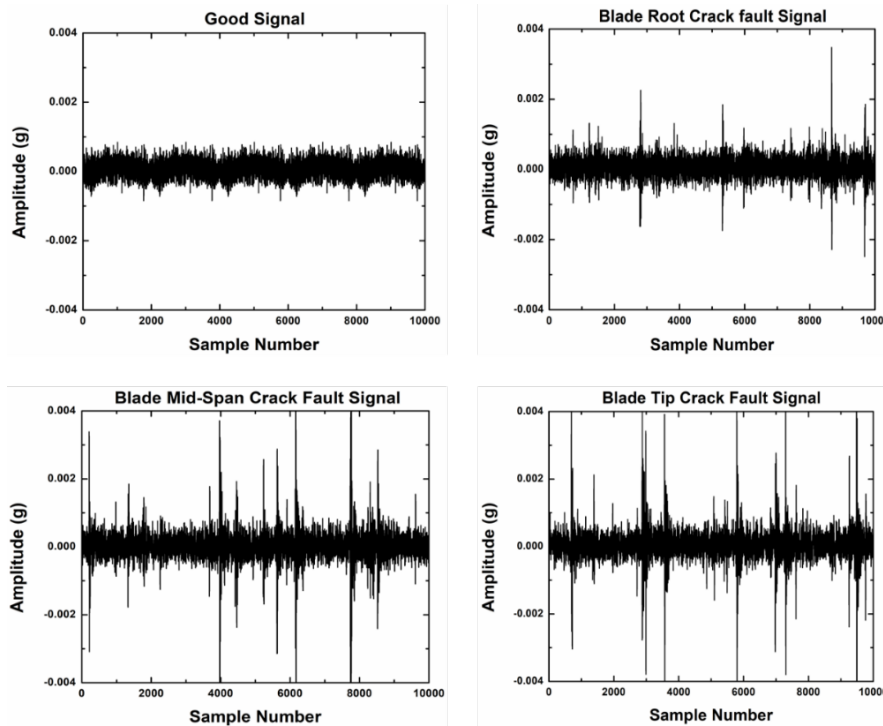


Figure 4: Vibration signatures of the blade

3 Feature extraction process

3.1 Statistical analysis for feature extraction

The vibration signals were obtained for good and other faulty conditions of the blades. “If the time domain sampled signals are given directly as inputs to a classifier, then the number of samples should be constant. The number of signal samples obtained is a function of rotatory motion of the blade speed. Hence, it cannot be used directly as the input to the classifier. However, a few features must be extracted before the classification process. Descriptive statistical parameters [Herp, Ramezani, Bach-Andersen et al. (2018)] such as sum, mean, median, mode, minimum, maximum, range, skewness, kurtosis, standard error, standard deviation and sample variance were computed to serve as features in the feature extraction process.

- Sum: It is the sum of all feature values for each sample.
- Mean: The arithmetic average of a set of values or distribution.
- Median: Middle value sorting out the greater and lesser splits of a data set.
- Mode: Most frequent value available in the data set.
- Minimum value: It refers to the least signal point value in a given signal.
- Maximum value: It refers to the extreme signal point value in a given signal.
- Range: Difference in extreme and least signal point values for a given signal.
- Skewness: Skewness illustrates the degree of irregularity of a distribution around its mean. The following formula was used for calculation of skewness.

$$\text{Skewness} = \frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - \bar{x}}{S_d} \right)^3 \quad (1)$$

- Kurtosis: Kurtosis point toward the flatness or the spikiness of the signal. Its value is very low for normal condition of the blade and high for the faulty condition of the blade due to the spiky nature of the signal and ‘s’ is the sample standard deviation

$$\text{Kurtosis} = \left\{ \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum \left(\frac{x_i - \bar{x}}{S_d} \right)^4 \right\} - \frac{3(n-1)^2}{(n-2)(n-3)} \quad (2)$$

- Standard error: Standard error is a measure of the amount of error in the prediction of y for an individual x in the regression, where x and y are the sample means and ‘n’ is the sample size.

$$\text{Standard Error (y)} = \sqrt{\frac{1}{n-2} \left[\sum (y - \bar{y})^2 - \frac{\sum [(x - \bar{x})(y - \bar{y})]^2}{\sum (x - \bar{x})^2} \right]} \quad (3)$$

- Standard deviation: This is a measure of the actual energy or power content of the vibration signal. The following formula was used for calculation of standard deviation.

$$\text{Standard Deviation } (\sigma) = \sqrt{\frac{n \sum x^2 - (\sum x)^2}{n(n-1)}} \quad (4)$$

- Sample variance: It is the variance of the signal points and the following formula was used for calculation of sample variance.

$$\text{Sample Variance} = \frac{n \sum x^2 - (\sum x)^2}{n(n-1)} \quad (5)$$

When the statistical feature extraction was completed, the features were chosen and feature selection method was carried out. The statistical features form the input to the feature selection method. With the selected feature, further classification was carried out. The selected statistical features are explained in Section 4.

3.2 Histogram analysis for feature extraction

The histogram was used as a feature extracting tool in this study. The reason behind choosing the histogram method for feature extraction is because it allows the viewers to easily compare the data and also they work well with large ranges of information or samples. They also provide a more actual form of consistency, as the intervals are always equal, a factor that allows easy data transfer from frequency tables to histograms. Hence, the histogram is preferred for feature extraction. Feature extraction involves reducing a number of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power; also it may cause a classification algorithm to over-fit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. From the noted vibration signals, the needed feature is taken and that features are denoted as histogram features. There are two main factors to be considered in the selection of bins they are, bin range and bin width [Joshuva and Sugumaran (2018)].

Bin is the sub-range used for grouping the data. Suppose, we are interested in the distribution of the marks of the students in a class then we have sub ranged like 0-10, 11-20, 21-30...91-100. Each sub-range can be called a bin. To construct a histogram, the first step is to bin the range of values, that is, divide the entire range of values into a series of intervals and then count how many values fall into each interval. The bins are usually specified as consecutive, non-overlapping intervals of a variable. The bins (intervals) must be adjacent and are often (but are not required to be) of equal size. The bin range must be from lowest of minimum amplitude (-0.01698) to the extreme of maximum amplitude (0.020592) of all the four classes (good, crack30, crack60 and crack90). The number of bins for the fault diagnosis of wind turbine blade has been attained by carrying out a sequence of trials using a J48 algorithm with a different number of bins. Initially, the range of bin is separated into two equivalent portions. That is to say, the number of bins utilized is two.

The two histogram features, to be specific, $X1$ and $X2$ are extracted and the relating classification accuracy is additionally acquired by using the J48 algorithm. The approach and methodology of performing the same using J48 algorithm are clarified in Section 4. A set of related trails is done with various numbers of bins from 2, 3, 4,5,..., 100 and the corresponding results are shown in Fig. 5. From Fig. 5, bin size 62 has been chosen since the classification accuracy of bin 62 was found to be 82.5%. A set of 62 starting from $X1$, $X2$... $X64$ were extracted from the vibration signals and these are denoted as histogram features. The amplitude ranges from -0.01698 to 0.020592. For further study, rather than utilizing vibration signals directly, the histogram features extracted from vibration signals

are utilized. The procedure of calculating applicable parameters of the signals that represent the data contained in the signal is called feature extraction. Histogram analysis of vibration signals yields distinctive parameters. All the extracted histogram features, X_1 to X_{64} extracted from the vibration signals may not contain the needed information for classification. The applicable ones are selected using the J48 algorithm.

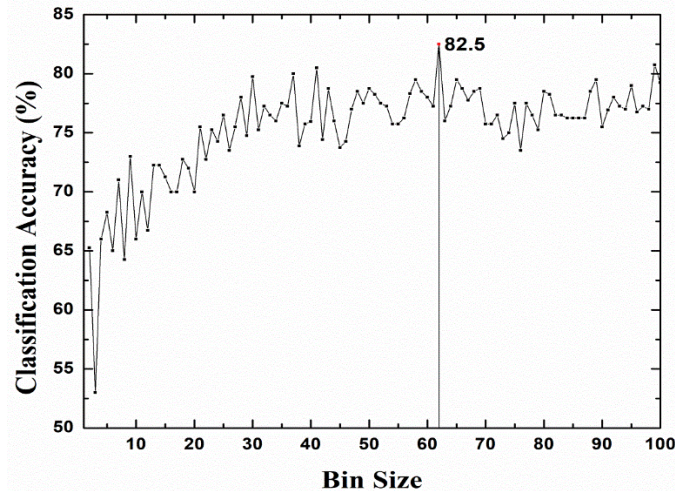


Figure 5: Bin size vs. classification accuracy

3.3 Autoregressive-moving average (ARMA) analysis for feature extraction

Autoregressive-Moving-Average (ARMA) features have been extracted to associate the attributes from input space to resultant space. Each input data set contains 10000 data points facilitating the blade signal. These signals were supplied as the source to the classifier. In general, algorithms find it is complex to deal with the large number of input features. In order to minimize the number of input variables, many researches provide a small number of measures of the data points rather than the data themselves. Thus, feature extraction process receives a particular attention to extract the meaningful information from the signals [Said and Dickey (1984)]. Autoregressive-Moving-Average (ARMA) models are numerical models of the auto correlation in a time series. ARMA models can be utilized to foresee the behaviour of a time series of past values alone. Such an expectation can be utilized as a standard to assess the conceivable significance of different variables to the system. ARMA models are generally utilized for forecast of monetary and mechanical time arrangement. ARMA models can be delineated by a progression of conditions. For effortlessness, the time arrangement was decreased to zero-mean first by subtraction of the specimen mean.

$$y_t = Y_t - \bar{Y}, t = 1, 2, \dots, N \quad (6)$$

where y_t the mean is adjusted series, Y_t is the original time series and \bar{Y} is the sample mean. Autoregressive (AR) models are a subset of ARMA models. In AR model, a time series is represented as a linear function of its defined values. The number of lagged defined values included is shown by the order of the AR model. The first-order autoregressive model can be understood easily. The equation for this model is

$$e_t = y_t + a_1 y_{t-1} \quad (7)$$

where, e_t is the noise, y_t is the mean-adjusted series, a_1 is the lag-1 autoregressive coefficient, y_{t-1} is the previous series. The occurred errors are defined as: the random-shock, and the residual. The residuals e_t is assumed to be random in time (not auto-correlated), and normally distributed. The equation for the first-order autoregressive model (1st AR) can be rewritten as

$$y_t = -a_1 y_{t-1} + e_t \quad (8)$$

The AR () model takes a form of regression model that has y_t regressed on its past value, and that e_t is analogous to the regression residuals. It is termed as autoregressive due to the regression on self (auto). An autoregressive model with higher-order contains more lagged y_t terms as predictors. For instance, the second order autoregressive model (2nd AR) is given as

$$e_t = y_t + a_1 y_{t-1} + a_2 y_{t-2} \quad (9)$$

where, a_1 and a_2 are the coefficients of autoregressive on lags 1 and 2. The pth order autoregressive model. It (pth AR) incorporates lagged terms on time $t-1$ to $t-p$. The moving average (MA) model is a structure of ARMA model in which the time series is regarded as a MA (unevenly weighted) of a random shock series e_t . The first-order moving average (1st MA) model is given by

$$y_t = e_t + c_1 e_{t-1} \quad (10)$$

where, e_t , e_{t-1} are the residuals at times t and $t-1$ and c_1 is the first-order MA coefficient. As with the AR models, MA models with higher-order include higher lagged terms. For instance, the second order moving average model, (2nd MA) is

$$y_t = e_t + c_1 e_{t-1} + c_2 e_{t-2} \quad (11)$$

The order of the MA model is denoted with the letter q . A second-order MA model is denoted by MA (q) with $q=2$. It has been seen that the autoregressive model incorporates lagged terms of the time series itself, while that the MA model incorporates lagged terms on the error or residuals. By including both sorts of lagged terms, it can be distinguished, what are called autoregressive-moving-average or ARMA models. The order of the ARMA model is incorporated in brackets as ARMA (p,q), where p is the autoregressive order and q the moving-average order. The simplest, and most frequently utilized ARMA model is ARMA (1,1) model as given below

$$y_t + a_1 y_{t-1} = e_t + c_1 e_{t-1} \quad (12)$$

The extraction of feature is performed through three strategies, in particular ARBURG, ARYULE and PYULEAR.

1. ARBURG function is used to calculate an estimate of autoregressive model parameters using Burgs Method. Here, $a=arburg(x,p)$ returns the normalized autoregressive (AR) parameters corresponding to a model of order p for the input array, x . If x is a vector, then the output array, a , is a row vector. If x is a matrix, then the parameters along the n^{th} row of a model the n^{th} column of x . In such case, a has $p+1$ columns and p must be less than the number of elements (or rows) of x .

$[a,e]=arburg(x,p)$ returns the estimated variance, e , of the white noise input.

$[a,e,k]=arburg(x,p)$ returns the reflection coefficients in k .

2. ARYULE is given by the expression $a=aryule(x,p)$ employs Yule-Walker method to fit an order- p AR model to input signal x by minimizing least square errors.

$[a,e]=aryule(x,p)$ returns the estimated variance, e , of the white noise input.

$[a,e,k]=aryule(x,p)$ returns the reflection coefficients in k .

3. PYULEAR is the autoregressive power spectral density (PSD) estimate using Yule-Walker method. $pxx=pyulear(x,order)$ returns the power spectral density estimate, pxx , of a discrete-time signal, x , found using the Yule-Walker method. When x is a vector, it is treated as a single channel. When x is a matrix, the PSD is computed independently for each column and stored in the corresponding column of pxx . pxx is the distribution of power per unit frequency. The frequency is expressed in units of rad/sample.

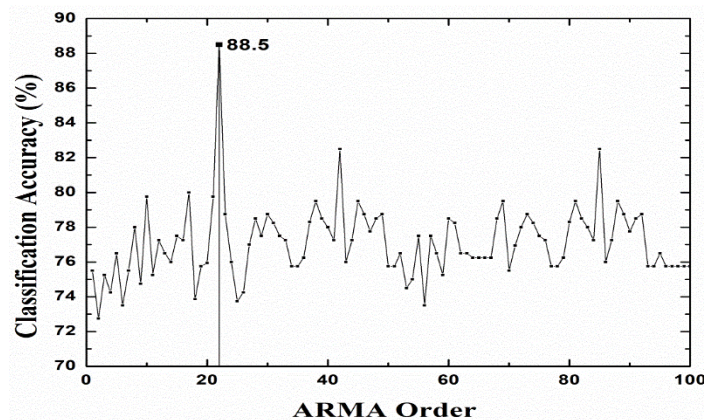


Figure 6: ARMA order vs. classification accuracy

Order is the order of the autoregressive (AR) model used to produce the PSD estimate. $pxx=pyulear(x,order,nfft)$ uses $nfft$ points in the discrete Fourier transform (DFT). For real x , pxx has length $(nfft/2+1)$ if $nfft$ is even and $(nfft+1)/2$ if $nfft$ is odd. For complex-valued x , pxx always has length $nfft$. If you omit $nfft$, or specify it as empty, then $pyulear$ uses a default DFT length of 256. From Fig. 6, order 22 was chosen as it provides the maximum classification accuracy of 88.5%. The MATLAB pseudo code for ARMA feature extraction is given below:

```
function []=ARMA()
for c=1:6
for n=1:100
if(n<=9)
    f=c*10+n;
elseif(n>=10 && n<=99)
    f=c*100+n;
elseif(n==100)
    f=c*1000+n;
```

```

end
    file = sprintf('%d.txt',f);
    a=load(file);
    [a1,e1,k1] = arburg(a,22);
    [a2,e2,k2] = aryule(a,22);
    [a3,e3,k3]= pyulear(a,22);
    outputfile = sprintf('D:\meanord022.xls');
if ~exist(outputfile,'file')
    fp = fopen(outputfile,'w');
%%Attributes Name
    fprintf(fp,'%d\t %d\t %d\t %d\t %d\t %d\t %d\t %d\t %d\t %d \n ', mean(a1), mean (e1),
mean (k1), mean(a2), mean(e2), mean(k2), mean(a3), mean(e3), mean(k3));
else
    fp = fopen(outputfile,'a');
    fprintf(fp,'%d\t %d\t %d\t %d\t %d\t %d\t %d\t %d\t %d\t %d \n ', mean(a1), mean (e1),
mean (k1), mean(a2), mean(e2), mean(k2), mean(a3), mean(e3), mean(k3));
end
end
end”

```

4 Feature selection using 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. “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. The acronym TDIDT, which stands for Top-Down Induction on Decision Trees, refers to this kind of algorithm. 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. 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 (Malik and Mishra (2017)).

The procedure of forming the decision tree and exploiting the same for vibration analysis is characterised by the following:

1. The set of required features extracted from wind turbine blade vibration studies forms the input to the algorithm; the output is the decision tree.
2. The decision tree has leaf nodes, which represent class labels, and other nodes associated with the classes (level of magnitude in this case) being analysed.

3. The branches of the tree represent each possible value of the parameter node from which they originate.
4. The decision tree can be used to express the structural information present in the data by starting at the root of the tree (top most nodes) and moving through a branch until a leaf node.
5. The level of contribution by each individual parameter is given by a feature measure within the parenthesis in the decision tree. The first number in the parenthesis indicates the number of data points that can be classified using that parameter set. The parameters appearing in the nodes of decision tree are in descending order of importance.
6. At each decision node in the decision tree, one can select the most useful parameter for classification using appropriate estimation criteria.

The criterion used to identify the best parameter invokes the concept of entropy and information gain discussed in detail in the following subsections. Decision tree algorithm (C4.5) has two phases: building and pruning. The building phase is also called as the ‘growing phase’.

4.1 Building phase

In the building phase, the training sample set with discrete-valued attributes is recursively partitioned until all the records in a partition have the same class. The tree has a single root node for the entire training set. Then for every partition, a new node is added to the decision tree. For a set of samples in a partition S , a test attribute X is selected for further partitioning the set into S_1, S_2, \dots, S_L . New nodes for S_1, S_2, \dots, S_L are created and these are added to the decision tree as children of the node for S . Also, the node for S is labelled with test X , and partitions S_1, S_2, \dots, S_L are then recursively partitioned. A partition in which all the records have identical class label is not partitioned further, and the leaf corresponding to it is labelled with the corresponding class. The construction of decision tree depends very much on how a test attribute X is selected. C4.5 uses entropy based information gain as the selection criteria. The entropy information gain is calculated in the following way.

Step-1: Calculate $Info(S)$ to identify the class in the training set S

$$Info(S) = - \sum_{i=1}^k \left\{ \left[freq \left(C_i, \frac{S}{|S|} \right) \right] \log_2 \left[freq \left(C_i, \frac{S}{|S|} \right) \right] \right\} \quad (13)$$

where $|S|$ is the number of cases in the training set. C_i is a class, $i=1, 2, \dots, K$. K is the number of classes and $freq(C_i, S)$ is the number of cases included in C_i .

Step-2: Calculate the expected information value, $Info_x(S)$ for test X to partition S

$$Info_x(S) = - \sum_{i=1}^L \left[\frac{|S_i|}{|S|} Info(S_i) \right] \quad (14)$$

where L is the number of outputs for test X , S_i is a subset of S corresponding to the i^{th} output and is the number of cases of subset S_i .

Step-3: Calculate the information gain after partition according to test X :

$$Gain(X) = Info(S) - Info_x(S) \quad (15)$$

Step-4: Calculate the partition information value $SplitInfo(X)$ acquired for S partitioned into L subsets

$$SplitInfo(X) = -\frac{1}{2} \left[\sum_{i=1}^L \frac{|S_{il}|}{|S|} \log_2 \frac{|S_{il}|}{|S|} + \left(1 - \frac{|S_{il}|}{|S|} \right) \log_2 \left(1 - \frac{|S_{il}|}{|S|} \right) \right] \quad (16)$$

Step-5: Calculate the gain ratio of $Gain(X)$ over $SplitInfo(X)$

$$GainRatio(X) = Gain(X) - SplitInfo(X) \quad (17)$$

The $GainRatio(X)$ compensates for the weak point of $Gain(X)$ which represents the quantity of information provided by X in the training set. Therefore, an attribute with the highest $GainRatio(X)$ is taken as the root of the decision tree.

4.2 Pruning phase

A large decision tree constructed from a training set usually does not retain its accuracy over the whole sample space due to over-training or under-fitting. Therefore, a fully grown decision tree needs to be pruned by removing the less reliable branches to obtain better classification performance over the whole instance space even though it may have a higher error over the training set. The C4.5 algorithm uses an error-based post pruning strategy to deal with over-training problem. For each classification node C4.5 calculates a kind of predicted error rate based on the total aggregate of misclassifications at that particular node. The error based pruning technique essentially reduces to the replacement of vast sub-trees in the classification structure by singleton nodes or simple branch collections if these actions contribute to a drop in the overall error rate of the root node.

4.3 Discretisation of continuous-valued attribute

It is important to know about how C4.5 solves the classification problem with continuous attributes because most of the signals in fault diagnosis field have continuous values. In fact, the discretisation process of continuous-valued attributes in C4.5 algorithm is a process to select the optimal threshold. For a continuous-valued attribute X , suppose it has m values in the training set and the values are sorted in ascending order, i.e., $\{a_1, a_2, \dots, a_m\}$ ($a_1 \leq a_2 \leq \dots \leq a_m$). For a special value a_i , it partitions the samples into two groups (a_1, a_2, \dots, a_i) and $(a_{i+1}, a_{i+2}, \dots, a_m)$. One has X values up to a_i , the other has X values greater than a_i and a_i is an optional threshold for discretisation. Therefore, there exist $m-1$ kinds of partitions or there are $m-1$ thresholds available. For each of these partitions, compute the information gain (see Section 4.1) and choose the partition (given the j^{th} partition) that maximises the gain. Accordingly, the boundary value a_j in the optimal partition is selected as the optimal threshold. This dynamic discretisation method is executed for each candidate attribute in every process to select the best test attribute.”

4.4 Application of Decision tree for feature selection

The algorithm has been applied to the problem under discussion for feature selection. Input to the algorithm is the set of features described in Section 4; the output is the decision tree, which is shown in Fig. 7 to Fig. 9. It is clear there from that the top node is the best node for classification. The level of contribution is not same and all eleven features are not equally important. The level of contribution by each individual feature is given by a feature measure within the parenthesis in the decision tree. The first number in the parenthesis indicates the number of data points that can be classified using that feature set. The second number indicates the number of samples against this action. If the

first number is very small compared to the total number of samples, then the corresponding features can be considered as outliers and hence ignored.

The other features appear in the nodes of decision tree in descending order of importance. It is to be stressed here that only features that contribute to the classification appear in the decision tree and others do not. Features that have less discriminating capability can be consciously discarded by deciding on the threshold. This concept is made use of in selecting good features. The algorithm identifies the good features for the purpose of classification from the given training dataset and thus reduces the domain knowledge required to select good features for pattern classification problem. A feature is 'a good feature', when its discriminating ability is high among the classes. It is characterised by the following:

- (a) The feature values do not vary much within a class.
- (b) It varies much among the classes.

The features which satisfy the above conditions will have more information gain while splitting and thus they appear in the order of importance in decision tree.

4.5 Features suggested by J48 decision tree

4.5.1 Suggested statistical features

From Fig. 7, one can identify the most dominating statistical features to represent the crack location in wind turbine blade are *sum* and *standard deviation*.

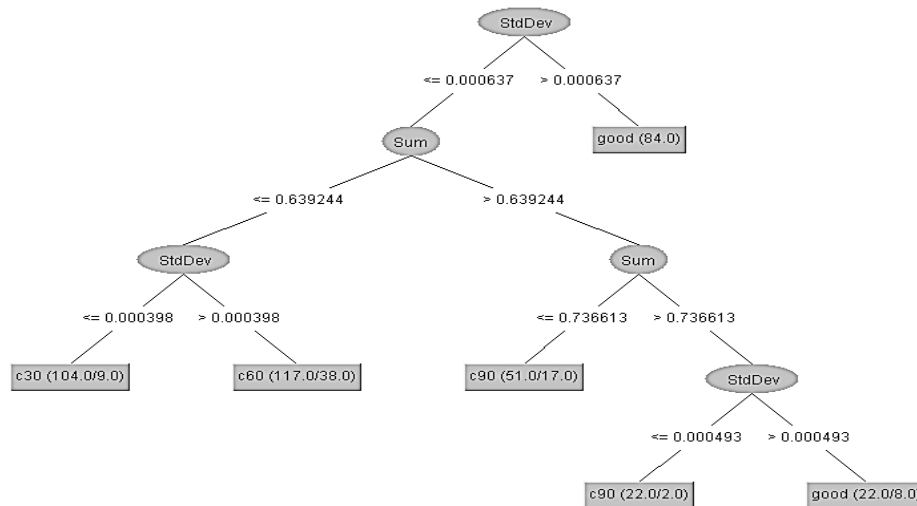


Figure 7: J48 Tree classification for statistical feature selection

4.5.2 Suggested histogram features

From Fig. 8, one can identify the most dominating histogram features to represent the crack location in wind turbine blade are *X28*, *X29*, *X31* and *X36*.

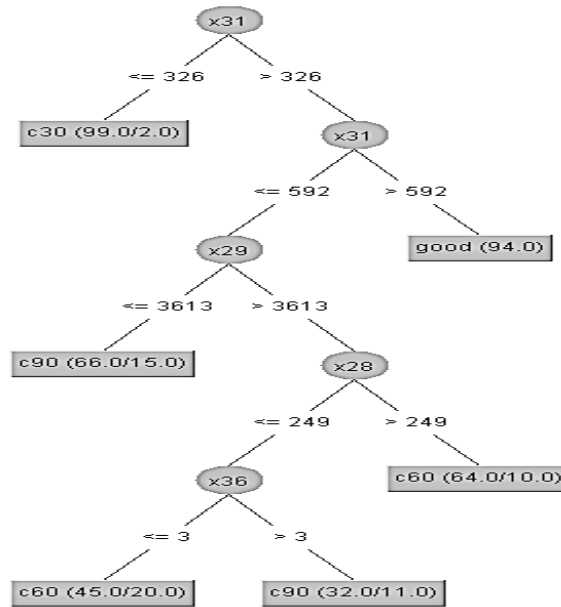


Figure 8: J48 Tree classification for histogram feature selection

4.5.3 Suggested ARMA features

From Fig. 9, one can identify the most dominating ARMA features to represent the crack location in wind turbine blade are a_2 and k_2 .

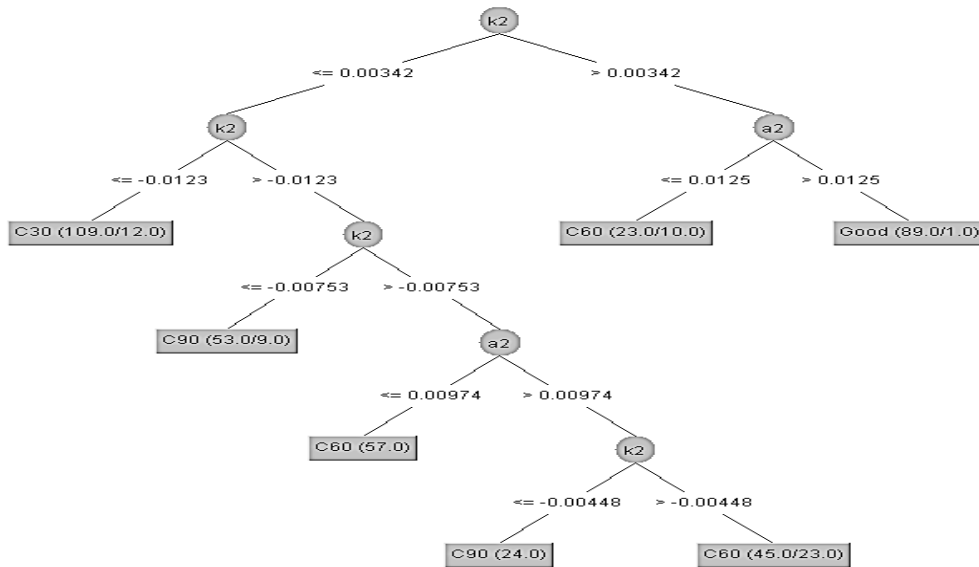


Figure 9: J48 Tree classification for ARMA feature selection

5 Machine learning classifiers

After the feature selection, the fault classification was carried out using machine learning classifiers. “In this study, multilayer perceptron (MLP) and logistic model tree (LMT) was used. A multilayer perceptron (MLP) is a feed-forward artificial neural network model that plots sets of data onto an arrangement of suitable yields. A multilayer perceptron contains different layers of hubs in an engaged outline, with individual layer totally connected to the following one. Aside from the input hubs, the individual hub is a neuron or preparing component with a nonlinear initiation capability. Multilayer perceptron utilizes a directed learning technique called back-propagation for instructing the system. Multilayer perceptron is a change of the standard linear perceptron and can separate information that is not linearly separable. The basic concept of a single perceptron was introduced by Rosenblatt [Rosenblatt (1958)]. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function. Mathematically, this can be written as

$$y = \varphi(\sum_{i=1}^n \omega_i x_i + b) = \varphi(\omega^T x + b) \quad (18)$$

where ω denotes the vector of weights, X is the vector of inputs, b is the bias and φ are the activation function.

A logistic model tree (LMT) essentially comprises a standard decision tree structure with logistic regression tasks at the leaves [Landwehr, Hall and Frank (2005)]. As in normal decision trees, a test on one of the qualities is connected with each internal hub. For an identified property with k values, the hub has k child hubs, and illustrations are sorted down one of the k branches relying upon their estimation of the feature. For numeric features, the hub has two child hubs and the test comprises of contrasting the characteristic significance to the threshold. Generally, a logistic model tree comprises of a tree structure that is comprised of an arrangement of internal or non-terminal hubs N and an arrangement of leaves or terminal hubs T . Let S indicate the entire occurrence in space, spread over by all characteristics that are available in the information. At that point the tree structure gives a separate section of S into areas S_t , and each area is characterized by a leaf in the tree.

$$S = \cup_{t \in T} S_t, \quad S_t \cap S_{t'} = \emptyset \text{ for } t \neq t' \quad (19)$$

Not like all decision trees, the leaves $t \in T$ has a related logistic regression function f_t rather than only a class name. The regression function f_t considers a subset $V_t \subseteq V$ of all characteristics present in the data and models the class relationship possibilities as

$$Pr(G = j | X = x) = \frac{e^{F_j(x)}}{\sum_{k=1}^j e^{F_k(x)}} \quad (20)$$

where

$$F_j(x) = \alpha_0^j + \sum_{v \in V_t} \alpha_v^j \cdot v \quad (21)$$

If $\alpha_{v_k}^j = 0$ for $v_k \neq V_t$. The model denoted by the whole logistic model tree is given by,

$$f(x) = \sum_{t \in T} f_t(x) \cdot I(x \in S_t) \quad (22)$$

where $I(x \in S_t)$ is 1 if $x \in S_t$ and 0 otherwise.

6 Results and discussion

From vibration signals, descriptive statistical features, histogram features and ARMA features were extracted. The best contributing features were selected using J48 decision tree algorithm. For feature selection process, in J48 decision tree algorithm, the minimum number of instances per leaf and the number of data used for reduced-error pruning was kept at 50. The rest of the features were eliminated as they contribute very less in crack detection and localization. Then, these selected features were given as input to the MLP and LMT classifier to determine the classification accuracy. In statistical feature selection, the most dominating features are sum and standard deviation (Fig. 7). For histogram feature selection, X_{28} , X_{29} , X_{31} and X_{36} were selected as the best contributing features (Fig. 8) and for ARMA feature selection, a_2 and k_2 was selected (Fig. 9). The selected features (statistical, histogram and ARMA) were given as the input to MLP and LMT classifiers for the crack detection and localization on wind turbine blade.

The overall classification accuracy for both the classifiers with respect to the selected feature was shown in Fig. 10. Here, one can find that for crack detection and localization, using ARMA features, multilayer perceptron (MLP) provides the maximum classification accuracy of 94.75% with the computational time of 1.51 seconds. In the MLP, the hidden layer was fixed to be 1. The learning rate (updated weight) was assigned to be 0.3 and the momentum applied to the weight while updating was fixed to be 0.2. This result was concluded using 10-fold cross validation. The data is divided randomly into 10 parts in which the class is represented in approximately the same proportions as in the full dataset. Each part is held out in turn and the learning scheme trained on the remaining nine-tenths; then its error rate is calculated on the holdout set. Thus, the learning procedure is executed a total of 10 times on different training sets. Finally, the 10 error estimates are averaged to yield an overall error estimate. In this way, the error rate is estimated efficiently and in an unbiased way. All the classification models built with given data set follows the 10 folds cross validation method.

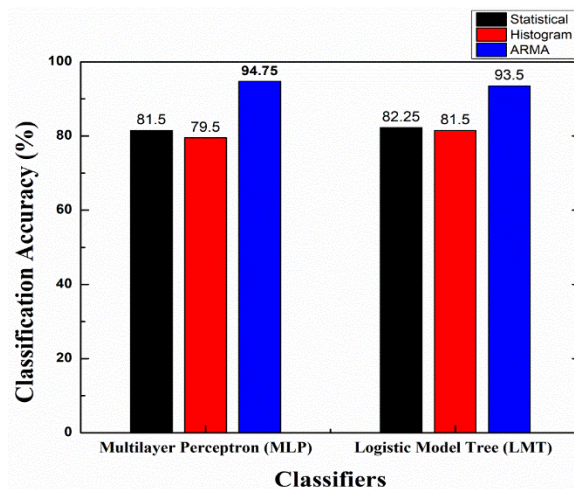


Figure 10: Overall classification accuracy

The confusion matrix of the MLP is shown in Tab. 2. In confusion matrix, the diagonal element represents the correctly classified instance and the others are misclassified. In confusion matrix too (Tab. 2), C30 represents the blade root crack, C60 represents the blade mid-span crack and C90 represents the blade tip crack. From MLP, the kappa statistic was found to be 0.93. The kappa statistic is used to measure the arrangement of likelihood with the true class. The mean absolute error is a measure used to measure how close forecasts or prediction are to the ultimate result. For MLP, the mean absolute error was found to be 0.0444. The root mean square error is a quadratic scoring rule which processes the average size of the error and for MLP; the root mean square error value is about 0.1491. From 400 samples, 379 samples were correctly classified (94.75%) and remaining 21 were misclassified (5.25%) with the computation time of 1.51 s.

Table 2: Confusion matrix for multilayer perceptron (MLP) classifiers

Blade conditions	Good	C30	C60	C90
Good	99	0	1	0
C30	0	93	1	6
C60	0	1	95	4
C90	0	4	4	92

Table 3: Classwise accuracy of multilayer perceptron (MLP) classifier

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC area
Good	0.990	0.000	1.000	0.990	0.995	1.000
C30	0.930	0.017	0.949	0.930	0.939	0.992
C60	0.950	0.020	0.941	0.950	0.945	0.987
C90	0.920	0.018	0.948	0.948	0.948	0.989

The detailed classwise accuracy of MLP is given in Tab. 3. The relative absolute error was found to be 11.8495% and root relative squared error was found to be 34.4336%. The classwise accuracy is expressed in terms of the true positive rate (TP), false positive rate (FP), precision, recall, F-Measure and receiver operating characteristics (ROC) area. True Positives (TP) is defined as the number of instances covered by the rule that are correctly classified, i.e., its class matches the training target class. False Positives (FP) is given as the number of instances covered by the rule that are wrongly classified, i.e., its class differs from the training target class. True Negatives (TN) is the number of instances not covered by the rule, whose class differs from the training target class. False Negatives (FN) is the number of instances not covered by the rule, whose class matches the training target class. The true positive (TP) rate should be close to 1 and the false positive (FP) rate should be close to 0 for a better classifier. One can observe from Tab. 3, the TP rate of most of the classes are close to 1 and FP rate were close to 0. This reassures that the result presented by confusion matrix in Tab. 2. Precision is the probability of retrieved instances that are relevant for the class. That is, it is the ratio of true positive (TP) to the retrieved instances (TP+FP). It is stated as $\frac{TP}{TP+FP}$. Precision is

also called as the positive predictive value and can be defined as a measure of exactness or quality.

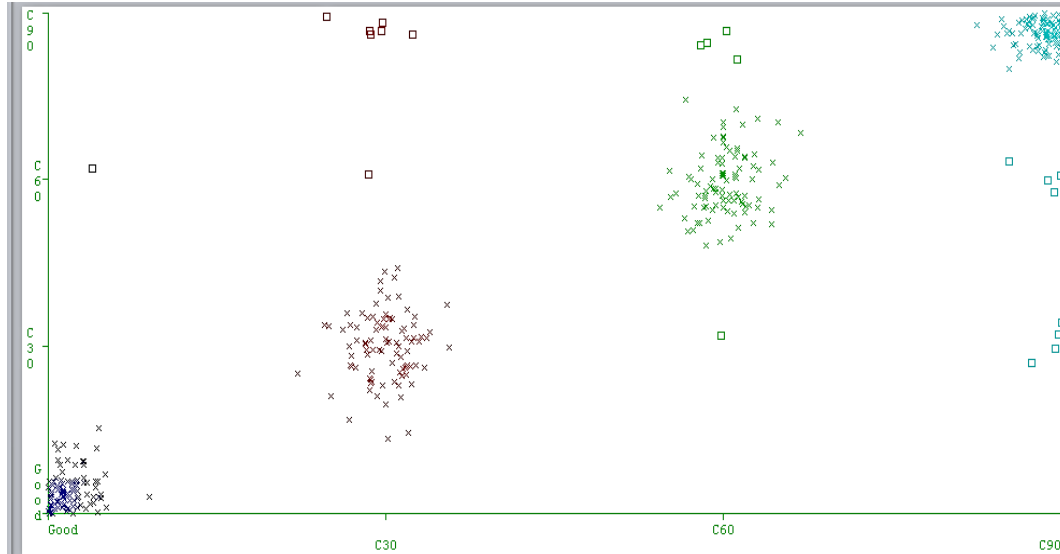


Figure 11: Classifier errors (classification vs. misclassification)

Recall is the information retrieval which shows the probability of the faults that are relevant to the classification that is successfully retrieved. That is the ratio of true positive (TP) to the overall instances (TP+FN). False negative (FN) is considered as type 2 error in which the instances indicate the misclassification but it is actually correctly classified. It is stated as $\frac{TP}{TP+FN}$. Recall is also called as the measure of completeness or quantity. F-measure is defined as the harmonic mean of both recall and precision. That is, this measure is approximately the average of the two (recall and precision) when they are close, and is more generally the square of the geometric mean divided by the arithmetic mean. The f-measure is expressed as $2 * \frac{Recall * Precision}{Recall + Precision}$. In MLP, from 400 samples, 379 samples were correctly classified (94.75%) and remaining 21 were misclassified (5.25%) with the computation time of 1.51 s. This can be used in real time scenario for crack detection and localization on wind turbine blades due to low computational time involved. The classifier error chart is shown in Fig. 11. Here the squared dots represent the misclassification and the 'x' denotes the correct classification.

7 Conclusion

Wind turbines are very important structure in extracting wind energy. This paper displayed an algorithmic based classification of vibration signals for the evaluation of the wind turbine crack detection and localization. From the acquired vibration data, 2 models were developed using data modelling techniques (MLP and LMT) and these models were tested using 10-fold cross validation. These classifiers were compared with respect to their maximum correctly classified instances and were found to be 94.75% with multilayer perceptron (MLP) classifier with the computation time of 1.51 s. The error rate is relatively

less and may be considered for the blade crack detection and localization. Hence, multilayer perceptron (MLP) classifier can be practically used for the condition monitoring on wind turbine blade to reduce the downtime and to maximize the harvest of wind energy.

References

- Abouhnik, A; Albarbar, A.** (2012): Wind turbine blades condition assessment based on vibration measurements and the level of an empirically decomposed feature. *Energy Conversion and Management*, vol. 64, pp. 606-613.
- Barnard, J. C.; Wendell, L. L.** (1997): A simple method of estimating wind turbine blade fatigue at potential wind turbine sites. *Transactions-American Society of Mechanical Engineers Journal of Solar Energy Engineering*, vol. 119, pp. 174-186.
- Bouزيد, O. M.; Tian, G. Y.; Cumanan, K.; Moore, D.** (2015): Structural health monitoring of wind turbine blades: acoustic source localization using wireless sensor networks. *Journal of Sensors*, pp. 1-11.
- Ghoshal, A.; Sundaresan, M. J.; Schulz, M. J.; Pai, P. F.** (2000): Structural health monitoring techniques for wind turbine blades. *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 85, no. 3, pp. 309-324.
- Herp, J.; Ramezani, M. H.; Bach-Andersen, M.; Pedersen, N. L.; Nadimi, E. S.** (2018): Bayesian state prediction of wind turbine bearing failure. *Renewable Energy*, vol. 116, pp. 164-172.
- Cui, H. Y.; Ding, N.; Hong, M.** (2016): Crack diagnosis of wind turbine blades based on EMD method. *IOP Conference Series: Materials Science and Engineering*, vol. 157, no. 1.
- Joshuva, A.; Sugumaran, V.** (2016): Fault diagnostic methods for wind turbine: a review. *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 7, pp. 4654-4668.
- Joshuva, A.; Sugumaran, V.** (2017): A comparative study of Bayes classifiers for blade fault diagnosis in wind turbines through vibration signals. *Structural Durability & Health Monitoring*, vol. 12, no. 1, pp. 69-90.
- Joshuva, A.; Sugumaran, V.** (2018): A study of various blade fault conditions on a wind turbine using vibration signals through histogram features. *Journal of Engineering Science and Technology*, vol. 13, no. 1, pp. 102-121.
- Laino, D. J.; Hansen, A. C.** (1997): An integrated approach to wind turbine fatigue analysis. *Transactions-American Society of Mechanical Engineers Journal of Solar Energy Engineering*, vol. 119, pp. 242-247.
- Landwehr, N.; Hall, M.; Frank, E.** (2005): Logistic model trees. *Machine Learning*, vol. 59, pp. 161-205.
- Liu, C.; Jiang, D.; Chu, F.** (2015): Influence of alternating loads on nonlinear vibration characteristics of cracked blade in rotor system. *Journal of Sound and Vibration*, vol. 353, pp. 205-219.
- Malik, H.; Mishra, S.** (2017): Application of GEP to investigate the imbalance faults in direct-drive wind turbine using generator current signals. *IET Renewable Power Generation*.
- Manwell, J. F.; McGowan, J. G.; Rogers, A. L.** (2010): *Wind Energy Explained: Theory, Design and Application*. John Wiley & Sons.

Marin, J. C.; Barroso, A.; Paris, F.; Canas, J. (2009): Study of fatigue damage in wind turbine blades. *Engineering Failure Analysis*, vol. 16, no. 2, pp. 656-668.

Pratumnopharat, P.; Leung, P. S.; Court, R. S. (2014): Wavelet transform-based stress-time history editing of horizontal axis wind turbine blades. *Renewable Energy*, vol. 63, pp. 558-575.

Rosenblatt, F. (1958): The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, vol. 65, pp. 386-408.

Said, S. E.; Dickey, D. A. (1984): Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, vol. 71, no. 3, pp. 599-607.