Bearing Fault Classification Using Multi-Class Machine Learning (ML) Techniques

C. Sujatha[1,*] and Aravind Mohan[2]

[1]Indian Institute of Technology Madras, Chennai 600036, India
[2]Google, Mountain View, California, 94043, USA

Abstract

Bearing elements are widely used in rotating machines and their failure results in a considerable amount of downtime of the machines. The aim of this work is to classify defects in a bearing. Three types of classification have been done: (i) Binary classification: classification as non-defective or defective bearing, (ii) 3-class classification such as non-defective, defective with inner ring defect and defective with roller defect and finally (iii) 7-class classification corresponding to no defect condition, three ring defect conditions pertaining to indentations of three different sizes on the inner ring and three roller defect conditions corresponding to indentations of three different sizes on the roller. The open-access data generated using a rolling bearing test rig from the Politecnico Di Torino, Italy, has been used for this work. The data had been obtained using 2 accelerometers on two bearing housings for multiple load and speed combinations. For classification, in the present work, classical ML algorithms such as logistic regression (LR), K-Nearest Neighbour (K-NN) classification algorithm, random forest (RF), support vector classifier (SVC) and kernel support vector machine (KSVM) have been used. All these techniques gave very promising results, the classification accuracy varying from 0.7969 to 0.9996 for all speed-load conditions. Such classification work across multiple operational conditions, with multiple fault conditions and multiple signatures with faulty components, has not been reported.

*Corresponding author. Email: sujathamohan5711@gmail.com

1. Introduction

In the last few decades, considerable progress has been made in the field of bearing fault diagnosis using vibration signals [1-5]. The use of sophisticated signal processing techniques like neural networks, discrete wavelet transform, spectral kurtosis, statistical methods, spectral methods, etc. was tried out by many researchers [6-9]. A good tutorial on rolling element bearing diagnostics has been presented by Randall et al. [10]. Singh et al. [11] have given a comprehensive review of the vibration modelling of rolling element bearings with defects. References [12-15] describe fault diagnosis of rolling bearings using specialized techniques like the K-NN algorithm, enhanced kurtogram, empirical mode decomposition,

principal component analysis and spectrogram. A good review of signal processing techniques used for bearing fault diagnostics including fast Fourier transform [FFT] algorithms has been presented in references [16] and [17]. Li et al. have presented the application of ensemble empirical mode decomposition (EEMD) and improved frequency band entropy in bearing fault feature extraction [18]. ML-based fault diagnosis has been discussed in references [19-20]. Early researchers looked at defect frequencies in the spectrum such as inner race defect frequency, outer race defect frequency, cage rotational frequency and ball or roller spin frequency. Later on, descriptors used in literature were based on statistical values like root mean square value, crest factor, kurtosis, probability density function, auto-correlation and cross-correlation functions, auto-spectral and cross-spectral density functions, transfer and coherence functions. Besides,

researchers have tried to obtain descriptors from time synchronous averaging, cyclic spectral analysis, cepstrum analysis, envelope analysis, spectral kurtosis, higher order spectral analysis, time-frequency analysis, spectrograms, spectral entropy, energy indicators, etc. They have also tried out feature extraction based on wavelet transforms, wavelet packet decomposition, neural networks and hidden Markov methods to name a few. Though these procedures are effective, they generally have to be custom-made for each machine, relying on human effort for the interpretation of the results. Also, they tend to be computationally expensive and difficult for real-time implementation. Hence in the present paper, five different ML algorithms have been tried out for rolling element and inner ring defects and the classification proves to be very promising.

2. Test Rig and Procedure

The present work makes use of the open access data [21] generated using a rolling bearing test rig by Politecnico Di Torino, Italy. The test rig had been set up at the Dynamic and Identification Research Group (DIRG) Laboratory in the Department of Mechanical and Aerospace Engineering for studying faults in high-speed aeronautical bearings.

2.1. Test rig

The test rig (Fig. 1) basically consisted of a high-speed spindle driving a shaft which had three appropriately lubricated roller bearings, specifically fabricated for this test [21]. The spindle speed was controlled through the control panel of an inverter. The body of the spindle was fixed to an extremely rigid support resting on a highly massive steel base plate which had a couple of supports for the outer rings of two identical roller bearings. The inner rings of these bearings were connected to a very short and thick hollow shaft, which was designed to run at speeds up to 35000 revolutions per minute (RPM). Provision had been made for the application of a load through a third and larger roller bearing at the centre of the shaft and the load was measured using a load cell. Triaxial accelerometers were fixed on the supports of the bearings and the electric spindle to measure accelerations. They were fixed at the two most significant locations on the structure, A1 and A2 (Fig. 1), located on the support of the damaged bearing under test, B1, and the support of the larger bearing used for the application of the external load, B2, respectively. The tests were done at variable rotational speeds, radial loads and different levels of bearing damage, along with temperature measurements. Table 1 gives the geometry of the roller bearings.

Data acquisition had been done [21] using an OR38 signal analyser and with analogue-to-digital converter having a 24-bit delta-sigma converter with synchronous acquisition (without multiplexing) from all channels. Every channel was set to a maximum of +/- 40 V. Six channels were recorded corresponding to the outputs of the two triaxial accelerometers placed on the two test bearings in the axial (X), radial (X) and radial (Y) directions as shown in

Table 2. The accelerations were measured for different stages of defect, at different speeds and under different loads. The time histories of the six channels mentioned in Table 2 were acquired with a sampling frequency of 51.2 kHz for a duration of 10 s for each test condition.
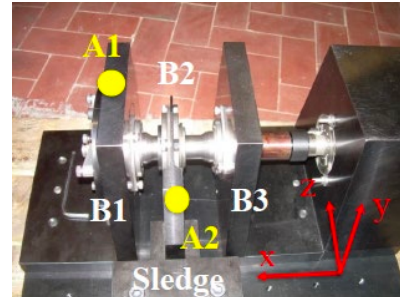

Figure 1. Test rig [21]

Table 1. Geometry of the roller bearings [21]

Bearing	Pitch diameter D (mm)	Roller diameter d (mm)	Contact angle ϕ (°)	No. of rolling elements (N)
B1 & B3	40.5	9.0	0	10
B2	54.0	8.0	0	16

Table 2. Direction of the measured accelerations (Refer Fig. 1) [21]

Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
Location A1			Location A2		
Axial, X	Radial, Y	Radial, Z	Axial, X	Radial, Y	Radial, Z

2.2. Test procedure

Every bearing was subjected to the same testing procedure for the different defect conditions [21]. The entire test took about 30 minutes. Due to the limited power of the inverter, the higher speeds could not be reached at the higher loading conditions. Localised conical indentations were created on the inner ring or on a single roller using a Rockwell tool, resulting in circular areas of sizes shown in Table 3 (defect conditions 0A to 6A). Table 4 gives the list of the speed-load combinations. There was a total of 4 load and 6 speed combinations. Data were not obtained for 3 load-speed combinations due to limitations in the inverter. Also, data were obtained from 2 accelerometers for 7 defect conditions. The test procedure was as follows.

(i) A short run at the minimum speed of 6000 RPM (100 Hz) at no load, to check the mounting arrangements.

(ii) Application of the static load in stages: 1000 N, then 1400 N and finally 1800 N.

(iii) Increase in the speed of the shaft from 0 RPM to 30000 RPM in steps of 6000 RPM.

(iv) Measurement of the accelerations at steady speeds of the shaft.

Table 3. Defect conditions [21]

Defect label	Defect location	Defect diameter
0A	No defect	
1A	Inner ring	450 μm
2A	Inner ring	250 μm
3A	Inner ring	150 μm
4A	Roller	450 μm
5A	Roller	250 μm
6A	Roller	150 μm

Table 4. Load and speed conditions for test [21]

Nominal load (N)	Nominal speed (RPM)				
0	6000	12000	18000	24000	30000
1000	6000	12000	18000	24000	30000
1400	6000	12000	18000	24000	--------
1800	6000	12000	18000	--------	--------

The datasets had been used to assess and compare the performance of different diagnostic algorithms [21]. Statistical tools like analysis of variance (ANOVA) had been applied to statistical features and linear discriminant analysis (LDA) had been carried out to see if the data were classifiable in a multi-dimensional space. Besides, an outlier analysis based on Mahalanobis distance had been formulated to distinguish between defective and non-defective states for various temperatures, speeds and loads.

3. Bearing Fault Classification

A faulty roller hits the outer race while rolling and produces a series of impulses, whose repetitive rate can be found solely from the geometry of the bearing if the rotational speed is constant. However, often, the faulty bearing frequencies are obscured in the spectra due to vibrations from other rotating parts. Many studies are reported on signal processing techniques for diagnostics and prognostics. The most commonly used methods are those based on statistical, spectral and probabilistic descriptors. In the present work, ML techniques have been used for defect classification. Each individual time series record of 10 s is broken down into 200 equally sized, contiguous sub-records, so as to artificially increase the number of sub-records available for analysis. The data points in each sub-record are further subsampled to select every third data point. Both these operations help reduce the computational effort and enable understanding of the volume of data needed to actually make the decisions for the defective and non-defective bearings. The features for these datasets are obtained using Time Series FeatuRe Extraction on basis of

Scalable Hypothesis tests (tsfresh) [23] with the relevant labels (binary or 3-class or 7-class) as the target. Tsfresh is an open-source Python package used for systematic feature extraction and feature selection. The number of features generated is ranked using an RF estimator [24] so as to select the most important features. Finally, five statistical models are trained using the datasets, the models being (i) LR, (ii) K-NN, (iii) RF, (iv) SVC and (v) KSVM. All five algorithms use the most important features extracted from the data using tsfresh and the RF estimator.

3.1. LR algorithm

LR is a supervised ML algorithm used for binary (can be extended to more) classification problems. LR, in spite of its name, is used more for classification. LR does not require a linear relationship between input and output variables because it takes a linear combination of input features and applies to them a nonlinear sigmoidal function. This method is used to map the input data to an output which has a probability between 0 and 1. The output represents the probability that the classification of the input data is 0 or 1 for binary classification. This algorithm has been extended for 3-class and 7-class classifications in the present work.

3.2. K-NN algorithm

This is one of the simplest and best-known classification algorithms. It is non-parametric in the sense that no assumptions are made regarding the underlying data. Known data are arranged in a space defined by the selected features. During the training process, this algorithm does not learn from the training set immediately but stores it instead. When a new dataset is presented, the algorithm transforms the data points into feature vectors or their mathematical values. The algorithm then compares it with the classes of the K closest classified data to assess the category of the new data. For this, it finds the Euclidean distance between the mathematical values of the new and the known data points. It then computes the probability of these new points being similar to the known data. Classification is done based on which points share the highest probabilities. The major advantage of the K-NN classifier is its simplicity and efficiency. However, it has the drawback that computation times can be long with large databases. Besides, finding the number of neighbours (K) to be used requires trial and error.

3.3. RF algorithm

RF, like LR, is a supervised ML algorithm used for classification, with labels for and mappings between inputs and outputs. The RF has multiple decision trees, each of which splits the data into smaller data sets based on the features of the data. When performing a classification task, each tree does a classification. The decision trees keep splitting the data into smaller data sets iteratively until a small set of data under a single classification is arrived at.

Once all the trees have done their classifications, the RF finds out which class has the largest number of votes and outputs it as a prediction. Because the RF utilizes the results of multiple decision trees, it is considered to be an ensemble ML algorithm. Many of the RF algorithms use bootstrapping, meaning that instead of training with the complete data set, each tree of the RF is trained with a subset of the complete data set, often called the bag. Multiple trees are trained using different bags, and later the results from all the trees are combined. So, it is quite possible that there are repetitions in the data that make it to a bag. Thus, the addition of a small number of extra training data can dramatically improve the prediction performance of a learned tree, though the training data do not change to any great extent. Such algorithms reduce the variance and the chance of overfitting. For classification, the decision is made based on "information gain", a measure of how much information is gained from a data set.

3.4. SVC Principle

A support vector machine (SVM) is again a supervised ML algorithm which can be used for both regression and classification. When used for classification, it is called an SVC and when used for regression, it is called a support vector regression (SVR) machine. The aim of an SVC is to find a hyperplane that maximally separates the two sets of data points present in the data set, the points being linearly separable, meaning the two classes can be separated by a straight line. In SVC, this line is determined by the margins and the support vectors. The margin denotes the area separating the two lines. The larger the margin, the better the classification. By support vectors are meant the data points through which each of the lines passes. These support vectors are nothing but the data points lying closest to the border of any one of the classes. Given labelled training data, the SVC outputs an optimal hyperplane which classifies new data into different classes. This hyperplane is then used to make predictions on new data points. The SVC is also called a maximum margin classifier since it finds the hyperplane with the largest distance to the nearest training data points of any class.

3.5. KSVM algorithm

KSVM allows the classification of data points which are not linearly separable. For such data, SVM supports the kernel method which allows one to implicitly map the input points into a high-dimensional feature space and then perform a non-linear classification. For this, one uses a complex mathematical mapping function that maps the lower-dimensional data points into a higher-dimensional space where they become linearly separable. This is a very powerful transformation. Then one finds a hyperplane that classifies the data points into two distinctive classes. Subsequently, the data points will be projected back to the initial lower-dimensional space using another function. In SVM, this is called the kernel method. The kernel function gives the similarity between the points in the original lower-dimensional feature space and the points in the newly transformed feature space. Many kernel functions like the Gaussian radial basis function (RBF), polynomial kernel, sigmoid kernel, etc. are used. A very interesting fact is that an SVM does not actually perform this transformation on the data points to the new high-dimensional feature space; rather the KSVM internally computes these complex transformations just in terms of similarity calculations between pairs of points in the original lower-dimensional feature space and the transformed feature space. This similarity function, which is a kind of complex dot product is actually the kernel of a KSVM. Though KSVM performs very well on a wide range of datasets, efficiency in terms of computer processor time and memory usage decreases as the size of the training set increases. Besides, it does not provide a direct probability estimator and it is difficult to interpret why a prediction was made.

3.6. Performance metrics for ML classifiers

The most commonly used performance metrics for ML classifiers are: (i) accuracy, (ii) precision, (iii) recall, (iv) F1 score and (v) receiving operator characteristic-area under the curve (ROC-AUC) metric. These performance metrics are defined as follows.

Accuracy is the simplest metric and is defined as the number of correct predictions divided by the total number of predictions. Accuracy lies between 0 and 1, with a value of 1 indicating a perfect model.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

Here TP = true positive, TN = true negative, FP = false positive and FN = false negative.

Precision is a measure of how good a model is at correctly identifying the positive class. In other words, out of all predictions for the positive class, it identifies how many were actually correct and lies between 0 and 1. A precision score close to 1 signifies that the model did not miss any true positives and is able to classify well between correct and incorrect labelling of say, a fraud. Using this metric alone for optimising a model would lead to minimising the false positives. This might be desirable for say a fraud detection case, but would be less useful for say, diagnosing cancer, as one would have little understanding of the positive observations that are missed.

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

Recall gives an indication of how good the model is at correctly predicting **all** the positive observations in the dataset. However, it does not give any information about the false positives. A recall value close to 1 implies that the model did not miss any true positives and is able to classify well between correctly and incorrectly labelling cancer patients, say. The recall is also a number less than 1. Usually, precision and recall are observed together by constructing a precision-recall curve. This can help to visualise the trade-offs between the two metrics at different thresholds.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

F1 score is defined as the harmonic mean of precision and recall and gives a number between 0 and 1. If the F1 score is 1, it indicates perfect precision and recall; if, however, the F1 score is 0, it means that either the precision or the recall is 0.

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

ROC-AUC is an evaluation metric for a binary classification problem at various threshold settings. An ROC curve is a graph showing two parameters: true positive rate (TPR) or recall and false positive rate (FPR) defined as FP/(TN+FP). AUC is a measure of separability. It is the area under the ROC curve with TPR on the Y-axis and FPR on the X-axis at various threshold values and essentially separates the 'signal' from the 'noise'. The AUC is a measure of the ability of a classifier to distinguish between classes. The larger the AUC, the better the performance of the model is in separating the positive and negative classes.

4. Results and Discussion

The classification was done using the five ML techniques namely LR, K-NN, RF, SVC and KSVM described in Sections 3.1. to 3.5. The input features for these were extracted using tsfresh as described in Section 3. The training was done across the complete acceleration data sets from both the accelerometers in X, Y and Z directions, details of which are given in Table 2, for all defect conditions given in Table 3 and for all combinations of loads and speeds given in Table 4. 70% of the data was used for training and 30% for testing, without any overlap between training and testing data. Results are shown in terms of distribution plots for binary, 3-class and 7-class classifiers. The distributions of the top 8 most important features are plotted. In the case of multi-defect classification, classification metrics and confusion matrices are obtained for the different classification techniques.

To visualise the results, a simple multi-layer perceptron (MLP) as shown in Fig. 2 is trained to predict the class, given a time series. An MLP is a type of artificial neural network that has multiple layers of interconnected nodes between input and output layers. The input layer receives the input data, the hidden layers process the data using activation functions, and the output layer produces the model's prediction. An MLP can model complex non-linear relationships between inputs and outputs and can be used for various tasks such as regression, classification and dimensionality reduction. The test portion of the dataset is encoded, and the results are visualized in 2 dimensions (to understand how the results can be classified). In Fig. 2(a) and 2(b), both f_1 and f_2 in Hidden layer 3 are nonlinear combinations of the top 128 most important features.

4.1 Results from binary classification

Figure 3(a) shows the scatter in the 2 most important features with binary classification. In Fig. 3(b) f_1 and f_2 are obtained from Hidden layer 3 of the MLP; each one is a nonlinear combination of the top 128 most important features. It can be seen that there is a clear demarcation between the two groups. Table 5 shows the performance of the 5 ML techniques in binary classification. Both KSVM and K-NN were able to achieve near-perfect classification with F1 scores of 0.9997 (P = 1.000, R=0.9994) and 0.9986 (P = 0.9987, R = 0.9985) respectively. Given the class imbalance, the F1 score, and ROC-AUC are leveraged over accuracy.
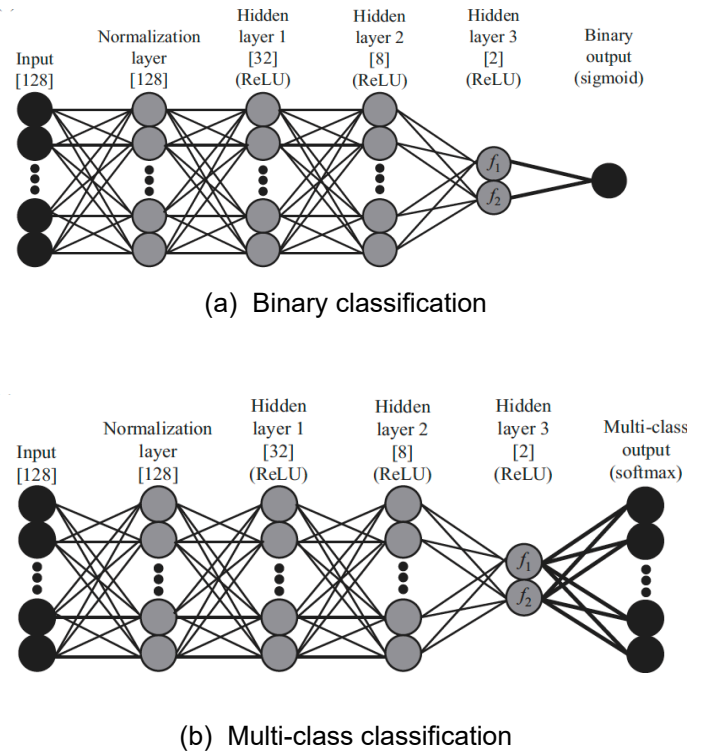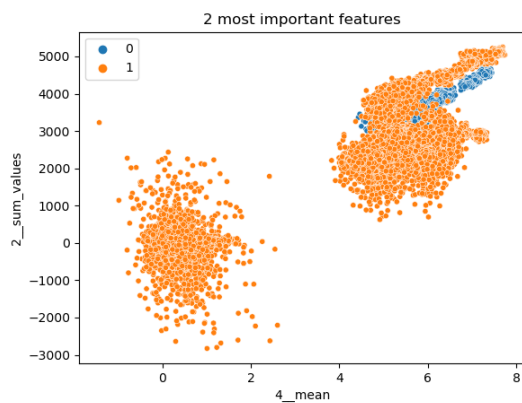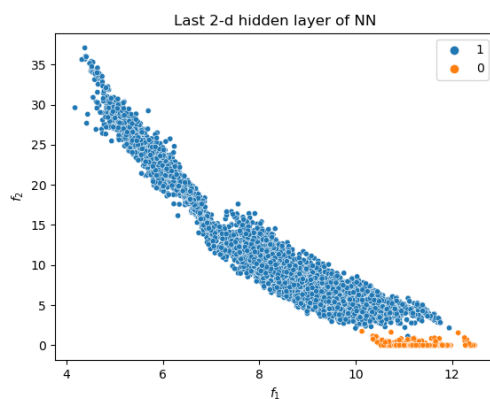

(a) Binary classification

(b) Multi-class classification

Figure 2. MLP configurations
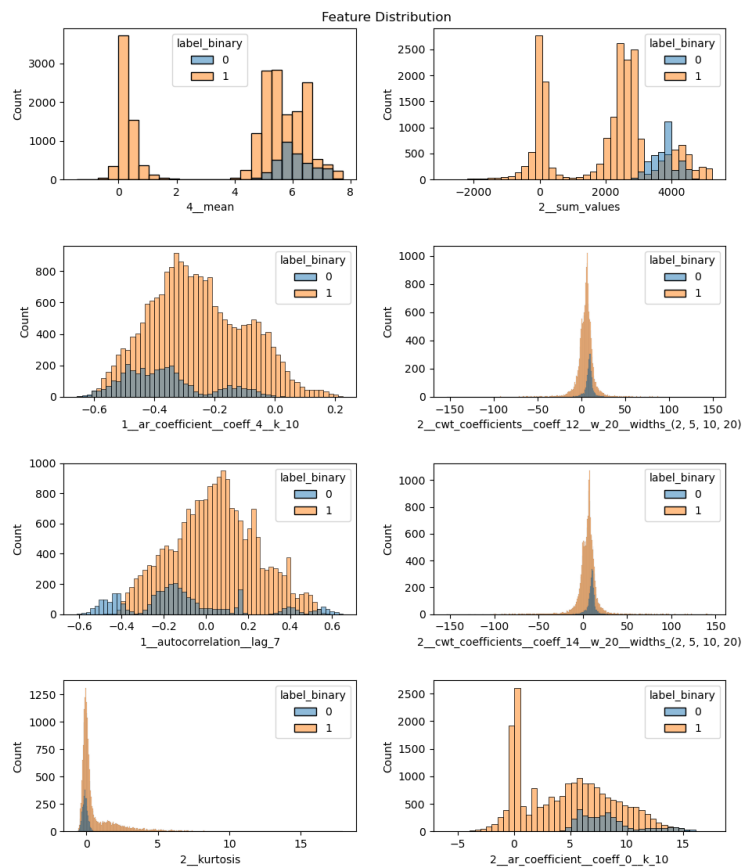
(a) 2 most important features

(b) Samples in the plane of f_1 and f_2

Figure 3. Scatter plot for binary classification (Note: 0-Non-defective; 1-defective)

Table 5. Binary classification performance

ML classifier	Accuracy	Precision	Recall	F1	ROC-AUC
LR	0.9628	0.9975	0.9592	0.9779	0.9722
K-NN	0.9976	0.9987	0.9985	0.9986	0.9953
RF	0.9277	1	0.9157	0.9560	0.9578
SVC	0.9671	0.9971	0.9644	0.9805	0.9739
KSVM	0.9994	1	0.9994	0.9997	0.9997

Figure 4 shows the histograms for the top 8 descriptors with binary classification. They are seen to overlap for 0-non-defective and 1-defective cases. They are essentially statistical quantities like mean, sum, autoregression and continuous wavelet transform coefficients, kurtosis, autocorrelation lag, etc. which researchers in literature have earlier intuitively tried to capture without using ML techniques.


Figure 4. Histograms of top 8 descriptors: binary classification

Note: 0-Non-defective; 1-defective

4.2 Results from 3-class classification

Figure 5(a) shows the scatter in the two most important features with 3-class classification. Figure 5(b) shows the samples in the f_1 - f_2 plane. Table 6 shows the performance of the 5 ML techniques in 3-class classification. KSVM and K-NN were able to achieve near-perfect results with accuracy scores of 0.9993 and 0.9926 respectively. Figures 6 (a-c) show the confusion matrices for LR, RF and KSVM classifications respectively. It can be seen that the values of the diagonal elements of the matrices are large, showing good classification. Figure 7 shows the histograms with 3-class classification. It is seen that the histograms for the 3 classes overlap as before. Most of the top 8 features for 3-class classification are the same as for binary classification. They are essentially statistical quantities like mean, sum, autoregression, continuous wavelet transform coefficients, kurtosis, autocorrelation lag, etc.
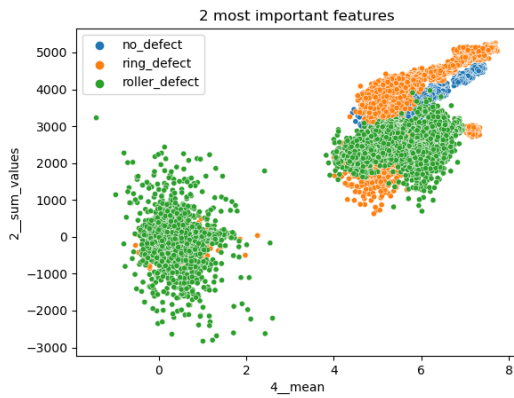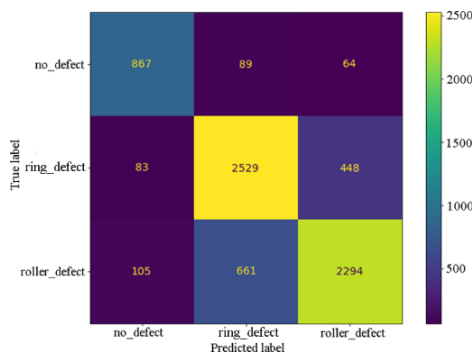
(a) 2 most important features
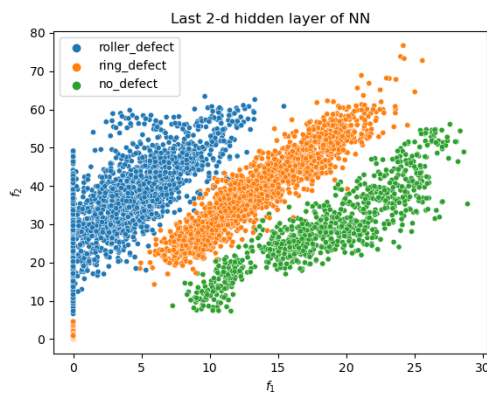
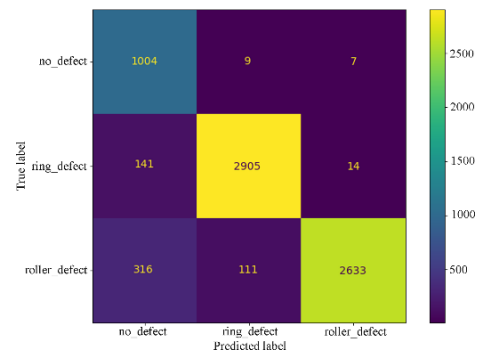(b) Samples in the plane of f_1 and f_2

Figure 5. Scatter plot for 3-class classification
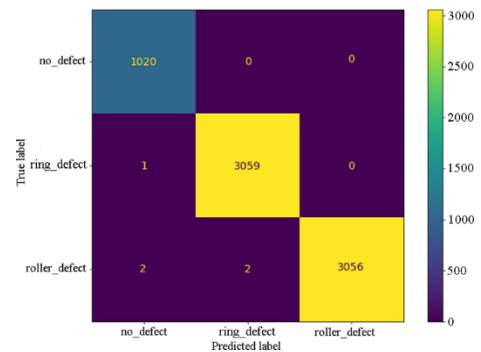
Table 6. Performance of 3-class classifiers

Name of ML classifier	Accuracy
LR	0.7969
K-NN	0.9926
RF	0.9162
SVC	0.8195
KSVM	0.9993

(a) LR

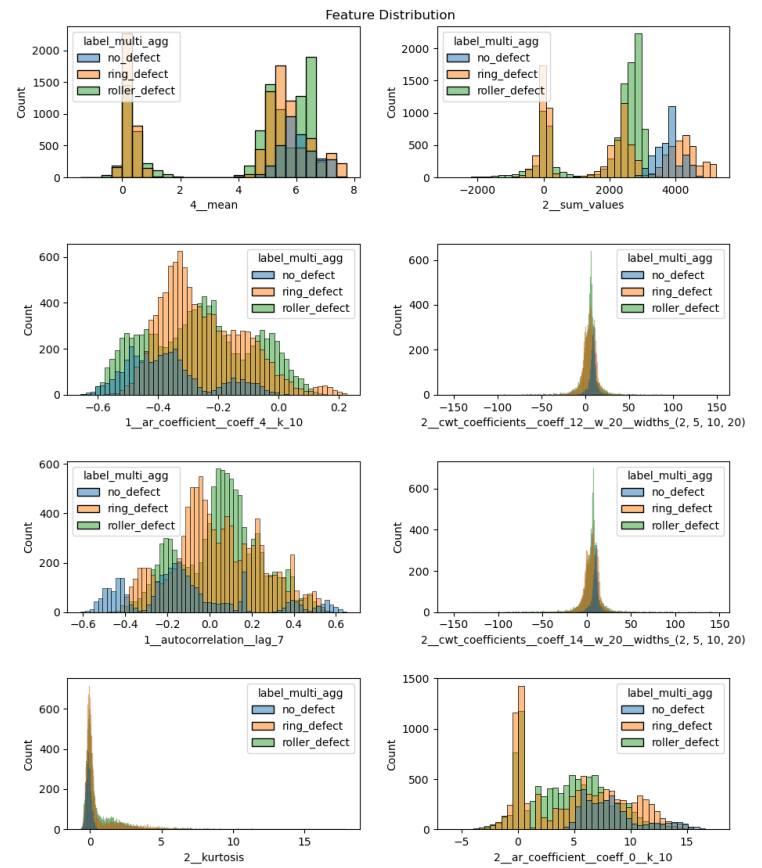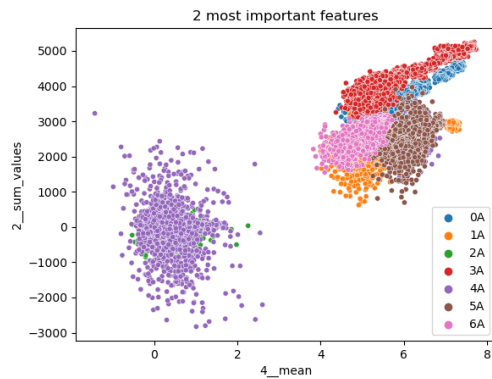

(b) RF

(c) KSVM

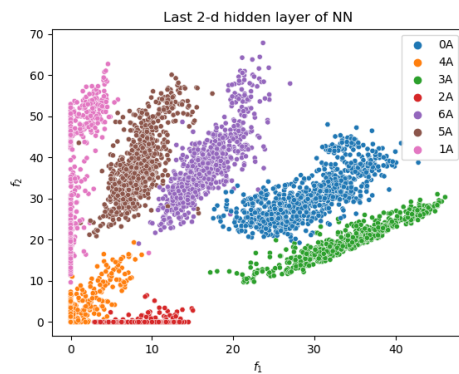**Figure 6.** Confusion matrices for 3-class classifiers

Figure 7. Histograms of top 8 descriptors: 3-class classification

4.3 Results from 7-class classification

Figure 8(a) shows the scatter in the classes with 7-class classification as defined in Table 3. In Fig. 8(b), the scatter in f_1 and f_2 are depicted in the plane of f_1 and f_2. Table 7 shows the performance of the 5 ML techniques in 7-class classification. KSVM and K-NN showed very good classification with accuracy scores of 0.9995 and 0.9955 respectively. Figures 9 (a-c) show the confusion matrices for LR, RF and KSVM respectively. It can be seen that the values of the diagonal elements of all confusion matrices are very large, showing extremely good classification, even better than for 3-class classification. Figure 10 shows the histograms with 7-class classification. It can be seen that most of the top 8 features for 7-class, binary and 3-class classifications are the same.
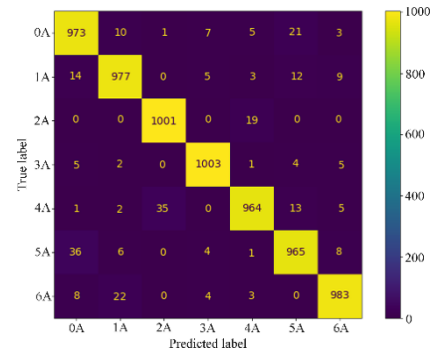
Table 7. Performance of 7-class classifier

Name of ML classifier	Accuracy
LR	0.9616
K-NN	0.9955
RF	0.9258
SVC	0.9807
KSVM	0.9996

(a) 2 most important features

(b) Samples in the plane of f_1 and f_2

Fig. 8. Scatter plot for 7-class classification

(a) LR
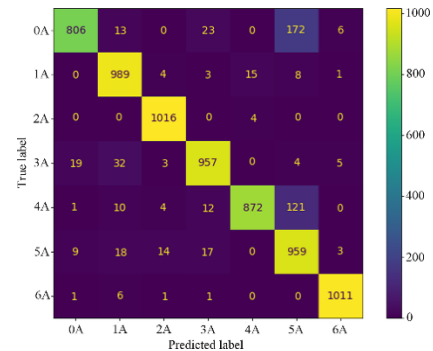
(b) RF
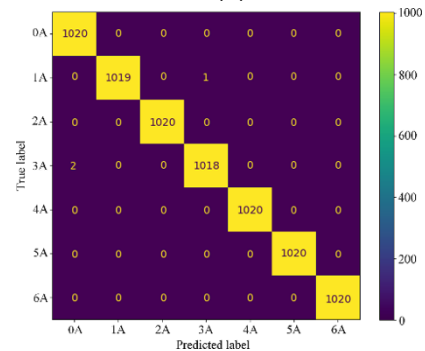
(c) KSVM

Figure 9. Confusion matrices for 7-class classifier

## 4.4 Comparison with Results from Polytecnico di Torino classification

Using the data generated using the rolling bearing test rig, researchers from the Politecnico Di Torino had also arrived at confusion matrices as shown in Fig. 11. For this, they had used Fisher's linear discriminant analysis (LDA), a simple multivariate classification tool to find linear combinations of features that characterize or separate two or more classes of objects. LDA is also used for dimensionality reduction; it is often used for feature extraction in pattern classification problems. LDA projects data from a D-dimensional feature space down to a D' (D>D') dimensional space in such a way as to maximize the variability between the classes and reduce the variability within the classes. As seen from Fig. 11, their LDA classification was not as good as the classification obtained in the present study shown in Figs. 6 and 9, proving that the ML algorithms used in the present study give very good classification.
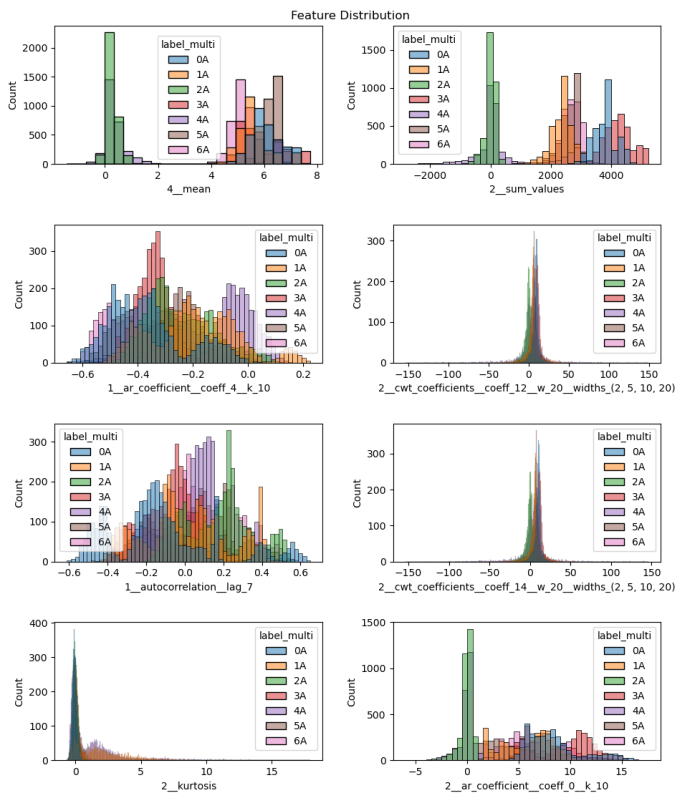
LDA	Training (in sample) Target Class							Validation (out of sample) Target Class						
Output Class	0A	1A	2A	3A	4A	5A	6A	0A	1A	2A	3A	4A	5A	6A
0A	51	0	10	12	0	9	15	48	0	9	15	0	12	14
1A	12	73	0	4	1	6	0	12	73	1	4	1	6	0
2A	14	0	59	13	0	4	7	16	0	52	17	0	5	7
3A	25	0	11	43	0	11	7	22	0	12	44	0	13	7
4A	1	4	0	0	81	10	0	1	4	0	1	80	11	0
5A	17	1	6	5	0	63	5	15	1	8	8	0	61	4
6A	13	0	8	12	0	7	57	14	0	8	12	0	7	57

Figure 11. LDA confusion matrices (%) [21]

5. Conclusions and Future Scope

This work has studied the classification of bearing defects for three classes of defects: 2, 3 and 7 over different load and speed conditions. The conclusions are as follows.

(a) Promising classification results were obtained regardless of whether the problem was considered as one of (i) binary classification with defective and non-defective bearings or (ii) 3-class classification considering a non-defective bearing, one with inner ring defect and one with roller defect or (iii) 7-class classification considering non-defective bearing, defective bearing with inner ring defects with indentations of 3 different diameters and roller defects with indentations of 3 different diameters.

(b) Considering binary classification, both KSVM and K-NN were able to achieve near-perfect classification results with F1 scores of 0.9997 (P = 1.000, R=0.9994) and 0.9986 (P = 0.9987, R = 0.9985) respectively.

(c) Considering 3-class classification with no defect, defective ring and defective roller, again KSVM and K-NN were able to achieve near-perfect classification with accuracy scores of 0.9993 and 0.9925 respectively.

(d) With 7-class classification, KSVM and K-NN were able to deliver accuracy scores of 0.9995 and 0.9955 respectively.

(e) In all cases, the most important descriptors of fault are essentially statistical quantities like mean, sum, autoregression and continuous wavelet transform coefficients, kurtosis, autocorrelation lag, etc.

The current work can be extended to detect faults that evolve in bearings naturally as they degenerate. Acceleration values could be sensed during various stages of their operational life and the data could be fed to the ML algorithms. These faults could also be obtained through accelerated tests. It is to be seen how the ML-based techniques described in this paper for man-made indentations work for naturally evolving degradation. Unsupervised ML techniques could also be tried.

Figure 10. Histograms of top 8 descriptors: 7-class classification

References

[1] Howard, IP.: A review of rolling element bearing vibration: Detection, diagnosis, and prognosis. Defence Science and Technology Organization, Canberra, Australia, DSTO-RR-0013 (1994)

[2] Tandon, N., Choudhury A.: A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. Tribology International, 32, 469–480, DOI:10.1016/S0301-679X(99)00077-8 (1999).

[3] Harris, TA.: Rolling Bearing Analysis, 4th ed., John Wiley & Sons Publication (2001).

[4] Kiral, Z., Karagulle, H.: Simulation and analysis of vibration signals generated by rolling element bearings with defects. Tribology International, 36(9), 667-678, DOI:10.1016/j.ymssp.2005.05.001 (2003).

[5] Tandon, N., Yadava, GS., Ramakrishna KM.: A comparison of some condition monitoring techniques for the detection of defect in induction motor ball bearings. Mechanical Systems and Signal Processing, 21(1), 244–256, DOI:10.1016/j.ymssp.2005.08.005 (2007).

[6] Subrahmanyam, M., Sujatha, C.: Using neural networks for the diagnosis of localized defects in ball bearings. Tribology International, 30(10), 739-752, DOI:10.1016/S0301-679X(97)00056-X (1997).

[7] Prabhakar, S., Mohanty, AR., Sekhar, AS.: Application of discrete wavelet transform for detection of ball bearing race faults. Tribology International, 35(12), 793–800, DOI: 10.1016/S0301-679X(02)00063-4 (2002).

[8] Antoni, J., Randall, RB.: The spectral kurtosis: Application to the vibratory surveillance and diagnostics of rotating machines. Mechanical Systems and Signal Processing, 20(2), 308-331, DOI:10.1016/j.ymssp.2004.09.002 (2006).

[9] Karacay, T., Akturk, N.: Experimental diagnostics of ball bearings using statistical and spectral methods. Tribology International, 42(6), 836–843 DOI:10.1016/j.triboint.2008.11.003 (2009).

[10] Randall, RB., Antoni, J., Rolling element bearing diagnostics-A tutorial. Mechanical Systems and Signal Processing, 25(2), 485–520, DOI:10.1016/j.ymssp.2010.07.017 (2001).

[11] Singh, S., Howard, C., Hansen, CH.: An extensive review of vibration modeling of rolling element bearings with localized and extended defects. Journal of Sound and Vibration, 357(3), 300–330, DOI:10.1016/j.jsv.2015.04.037 (2015).

[12] Zhu, X., Zhang, Y., Zhu, Y.: Intelligent fault diagnosis of rolling bearing based on kernel neighborhood rough sets and statistical features. Journal of Mech Science and Technology, 26, 2649–2657, DOI 10.1007/s12206-012-0716-9 (2012).

[13] Wang, D., Tse, PW., Tsui, KL.: An enhanced kurtogram method for fault diagnosis of rolling element bearings. Mechanical Systems and Signal Processing, 35(1-2), 176–199, DOI:10.1016/j.ymssp.2012.10.003 (2013).

[14] Xu, Z., Liu, K., Zhang, H., Wang, D., Zhang, M.: A fault diagnosis method for rolling bearings based on empirical mode decomposition and principal component analysis. Journal of Vibration and Shock, 33(23), 133–139, DOI:10.13465/j.cnki.jvs.2014.23.024 (2014).

[15] Wang, Y., Xiang, J., Markert, R., Liang, M.: Spectral kurtosis for fault detection, diagnosis and prognostics of rotating machines: A review with applications. Mechanical Systems and Signal Processing, 66-67, 679–698 (2016).

[6] Rai, A., Upadhyay, S.: A review on signal processing techniques utilized in the fault diagnosis of rolling element bearings. Tribology International, 96, 289–306, DOI:10.1016/j.triboint.2015.12.037 (2016).

[17] Lin, HC., Ye, YC.: Reviews of bearing vibration measurement using fast Fourier transform and enhanced fast Fourier transform algorithms. Advances in Mechanical Engineering, 11(1), DOI:10.1687814018816751 (2019).

[18] Li, H., Liu, T., Wu,. X, Chen, Q.: Application of EEMD and improved frequency band entropy in bearing fault feature extraction. ISA Transactions, 88, 170–185, DOI: 10.1016/j.isatra.2018.12.002 (2019).

[19] Singh, J., Azamfar, M., Li, F., Lee, J.: A systematic review of machine learning algorithms for prognostics and health management of rolling element bearings: Fundamentals, concepts and applications. Measurement Science and Technology, 32 (01), DOI:10.1088/1361-6501/ab8df9 (2020).

[20] Zhang, X., Zhao, B., Lin, Y.: Machine learning based bearing fault diagnosis using the Case Western Reserve University data: A Review. IEEE Access, PP(99):1-1, 155598–155608, DOI:10.1109/ACCESS.2021.3128669 (2021).

[21] Daga, AP., Fasana, A., Marchesiello, S., Garibaldi, L.: The Polytecnico di Torino, Rolling bearing test rig: Description and analysis of open access data. Mechanical Systems and Signal Processing, 120, 252–273. doi:10.1016/j.ymssp.2018.10.010 (2019).

[22] Christ, M., Braun, N., Neuffer J., Kempa-Liehr AW.: Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh - A Python package). Neurocomputing, 307, 72-77, Doi: 10.1016/j.neucom.2018.03.067 (2018).

[23] https://scikitlearn.org/stable/modules/generated/sklearn.feature_selection.RFE.html.