Detection of Fault in Bearings using SVM, RF, NN

Love Kush Tak
Dept of Civil Engineering
IIT Bombay
Mumbai, India
170040113@iitb.ac.in

Harshita Agarwal
Dept. of Chemical Engineering
IIT Bombay
Mumbai, India
180020034@iitb.ac.in

Rishabh Sharaff
Dept of Civil Engineering
IIT Bombay
Mumbai, India
180040081@iitb.ac.in

Shantanu Chhaparia
Dept of Civil Engineering
IIT Bombay
Mumbai, India
170040078@iitb.ac.in

***Abstract*—This document presents various machine learning models applied to vibration signals collected from bearings of different health conditions under time-varying rotational speed conditions and then compare accuracies of the models. The main aim is to use the most accurate model in real-life to find health status of machines. This would help industries save a lot of money and time from breaking down of machines. Early detection of fault will also lead to have timely repair and maintenance of machine. This paper would also discuss about how to extract and select features, data visualization, selecting best hyper-parameters for the model.**

***Index Terms*—healthy, inner race fault, outer race fault, fault detection, accuracy**

I. Introduction

Fault detection in machines using vibration monitoring has been increasing due to easy availability of contact or contactless vibration sensors. Breakdown of machines is a problem for users, for specialised machines it is even more difficult to get it repaired fast. Specialised machine needs highly qualified technicians which would lead to high cost of repairing as well as a lot of time getting spent in repairing. Early fault detection plays a key role here as it could prevent breakdown of machines, early repairing and maintenance could be done. This saves a lot of money and time for the users. The vibration data in real-life is a combination of frequencies which makes it difficult to predict the healthy and fault machines. Therefore machine learning is considered as a best alternative to predict the health status based on the parameters obtained from training the data. Accuracy should be very high in such cases as if our machine is healthy and it predicts it faulty, then users will be wasting time and money in unnecessary check from technicians.

II. dataset explanation

The bearing vibration data is collected under time-varying rotational speed conditions for different bearing health conditions. The bearing vibration data is important for bearing fault detection. The health conditions of the bearing considered are healthy, inner race fault and outer race fault. The operating rotation speed of the dataset include increasing speed, decreasing speed, increasing then decreasing speed, and decreasing then increasing speed. An accelerometer was used was used to collect vibration data and an incremental encoder was used to collect the rotational data. The data is recorded in Ottawa, Canada and its source is mentioned here [16]. The original data consisted of only one column, i.e., vibration data thus it was required to extract features from it in order to implement models. The features identified were 12 different statistical measure like mean, rms, std dev, kurtosis etc. Henceforth the data has been divided into thousand different batches having two thousand vibration data each in order to get these statistical measures. The frequency of the dataset was 200kHz and the total time for which the 200kHz vibration data was obtained is 10 sec, hence each data point is associated with a time of 5 micro seconds.

III. Literature Review

Bearing failure is one of the major causes of equipment downtime in any manufacturing industry using rotating equipment. Most of these bearing failures can even cause the whole system to perform erroneously resulting in economic and human losses (Li et al. 2018[1]; Goyal and Pabla 2015[2]). Vibration signature is a widely use method for machine health monitoring and accelerometer, being the most commonly used vibration sensor. After data acquisition, (Goyal 2019[3]) adopted Discrete Wavelet transform (DWT) and Mahalanobis Distance (MD) criteria for signal denoising and feature selection from the time-domain vibration features respectively. Most time domain features are statistical features such as mean value, root mean squares, standard deviation, kurtosis, skewness, peak-to-peak and others. Time domain and frequency domain features usually easy to calculate, are effective in extracting the features of the original rotating machinery signals and so, considered as classification features. Although, most of the vibration signals may have non-stationary characteristic. Wavelet transform, one of the most useful signal analysis methods, has been proved a superior method than the conventional Fourier Analysis in handling non-stationary signals (Konar and Chattopadhyay 2011 [4]; Zhang et al. 2013 [5]). Another method to solve the non-stationary characteristic is the Empirical Mode Decomposition (EMD) technique, where the vibration signal of a rotating machine is decomposed into a set of intrinsic mode functions (IMFs)[6]. Each IMF may be considered as a basic function of the signal. EMD energy entropy is calculated using the first few IMFs which contain more energy. With these features extracted, classification models are trained for diagnosis. Mahalanobis distance is a useful statistical measure to evaluate the resemblance of an unknown data set to a known one

(Lebaroud and Clerc2008[7]; Niu et al. 2011[8]; Wu et al. 2013 [9]). (Hu et al. [10] and Sreejith et al. [11]) combined time domain features with artificial neural network (ANN), in bearing fault diagnosis. In [12], time domain features and frequency domain features were combined using information fusion and an ANN model was trained for fault diagnosis. In (2018 Elsevier B.V.[13]), three techniques (i.e. statistical analysis, Fast Fourier Transform(FFT), and Variance Mode Decomposition(VMD)) are adopted to extract multi-domain feature contents aimed to fully reveal the intrinsic property of the raw signal. Afterwards, Laplacian Score (LS) algorithm, is used to evaluate classification sensitivity of extracted features and rearrange the feature space to get a low-dimensional feature set. At last, particle swarm optimization-based support vector machine (PSO-SVM) classification model is presented to differentiate the fault category. Cao et al. in [14] trained a SVM model with feature extraction using PCA method. Ali et al. [15] adopted empirical mode decomposition (EMD) energy entropy to extract a feature vector as the input of artificial neural network (ANN) for achieving the automatic detection of bearing fault severity.

Random Forest Classifier has also been widely used by researchers (Bo-Suk Yang, et al. [17]) because of its fast execution, high performance and as it does ensembles decision trees. By increasing various number of trees we get much better accuracy while using single decision trees. For classification problem it works on maximum votes.

In (Sanyam Shukla, et al. [18]), it has been observed that change in vibration in faulty machines is more as compared to healthy machines which can be seen by plotting Root Mean Square (RMS) or standard deviation (stdev.) for healthy and faulty data. And also to extract statistical feautres, data is divided into samples as more the number of samples more features we will have to train and better the accuracy we will get. But increasing too much number of sample increases computational time to so it is necessary to choose optimal number of samples.

Convolution Neural Network (CNN) has widespread application in fault diagnosis, it is compared much easier to train than Deep Neural Networks (DNN).CNN has the ability to extract useful and robust features for the monitoring signals. 1D-CNN can be applied directly on the raw signals but due to noise interference components increases the feature extraction capability of CNN and increases the training cost of CNN (Ince et al. [19]). Time-frequency domain analysis methods like Empirical Mode Decomposition (EMD), Wavelet Transform (WT), Hilbert-Huang transform and S-transform (ST) algorithms has been used for fault detection. These convert the 1D data into 2D on which CNN can be applied [20-24].

IV. Procedure & Experiment

A. Data Visualization

Vibration data were plotted as scatter plots for all three bearing conditions, i.e., healthy, inner fault and outer fault. The visualizations were observed for the four cases of speed which are increasing speed, decreasing speed, increasing then decreasing speed and decreasing then increasing speed. The visualizations lead to a conclusion that inner race fault could be easily identified using vibration data in increasing rotational speed condition.

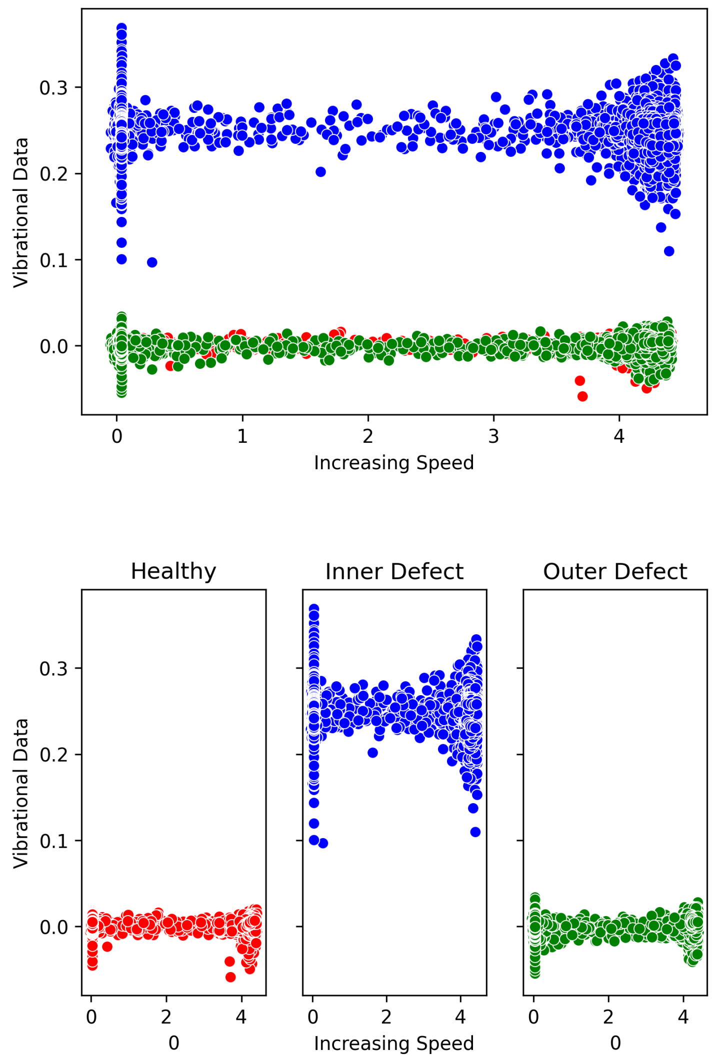

Fig. 1. Vibration data in increasing speed condition

The amplitude was plotted in time and frequency domain respectively for Inner fault and healthy, Outer Fault and Healthy conditions for the four rotation speed conditions. The visualizations clearly demonstrated the amplitudes were differing for different health conditions, and were convincing of the fact that we could predict health of the motor using machine learning models.

In frequency domain, the plots obtained showed similar conclusions as that in the time domain-

B. Feature Extraction

It is defined as a process to estimate some measures which will express the signal. From the raw accelerometer data to use it for classification we extracted some features from

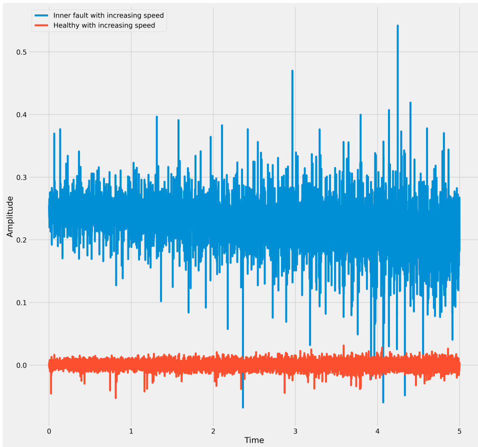

Fig. 2. Amplitude in *time domain* for Increasing Speed of Inner Fault(Blue legend) and Healthy(Red)

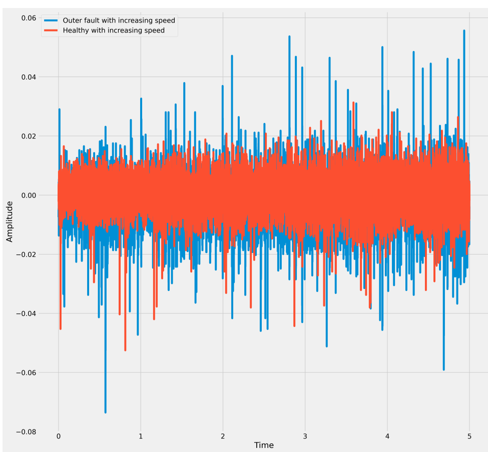

Fig. 3. Amplitude in *time domain* for Increasing Speed of Outer Fault(Blue legend) and Healthy(Red)

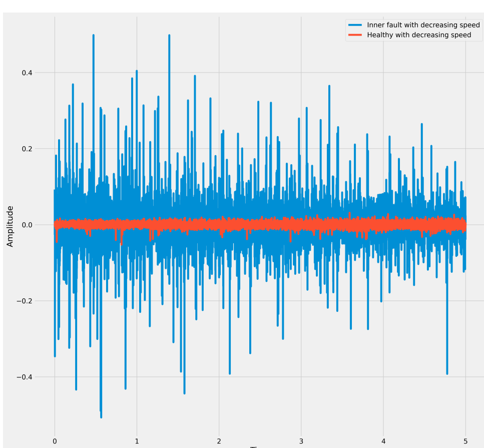

Fig. 4. Amplitude in *time domain* for Decreasing Speed of Inner Fault(Blue legend) and Healthy(Red)

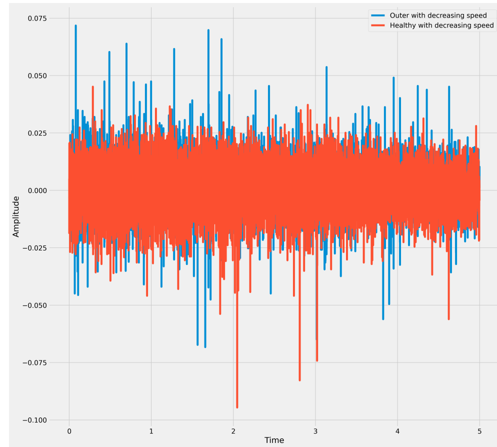

Fig. 5. Amplitude in *time domain* for Decreasing Speed of Outer Fault(Blue legend) and Healthy(Red)

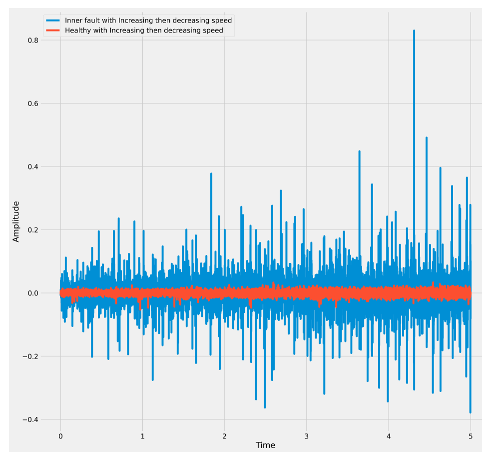

Fig. 6. Amplitude in *time domain* for Increasing then decreasing condition of Inner Fault(Blue legend) and Healthy(Red)

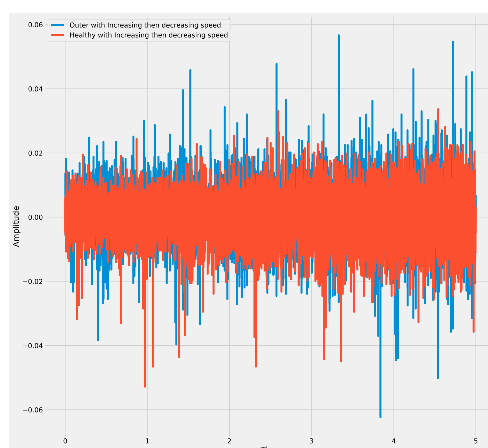

Fig. 7. Amplitude in *time domain* for Increasing then decreasing condition of Outer Fault(Blue legend) and Healthy(Red)

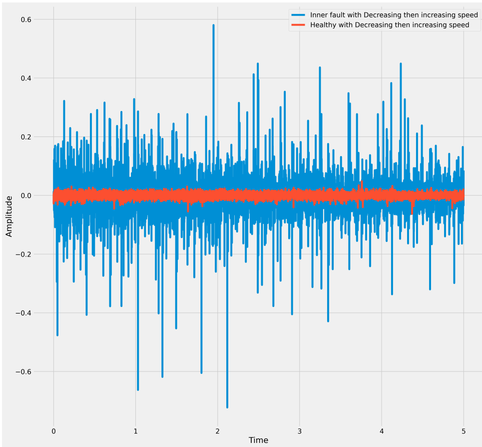

Fig. 8. Amplitude in *time domain* for Decreasing then increasing condition of Inner Fault(Blue legend) and Healthy(Red)

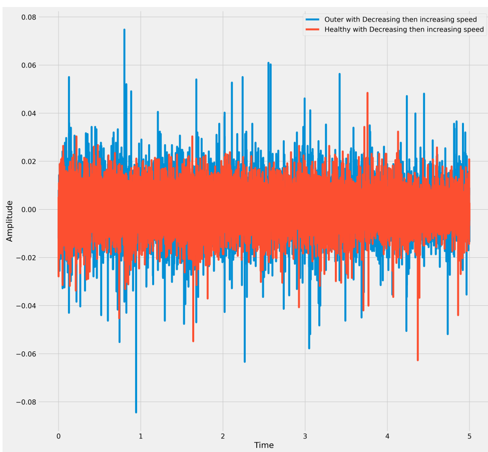

Fig. 9. Amplitude in *time domain* for Decreasing then increasing condition of Outer Fault(Blue legend) and Healthy(Red)

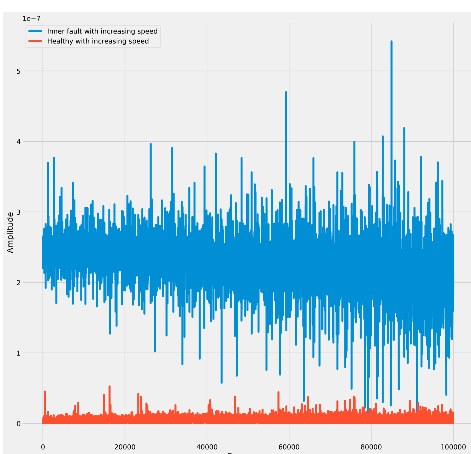

Fig. 10. Amplitude in *frequency domain* for Increasing Speed of Inner Fault(Blue legend) and Healthy(Red)

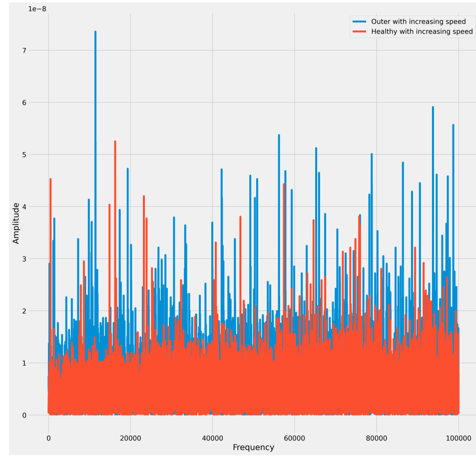

Fig. 11. Amplitude in *frequency domain* for Increasing Speed of Outer Fault(Blue legend) and Healthy(Red)

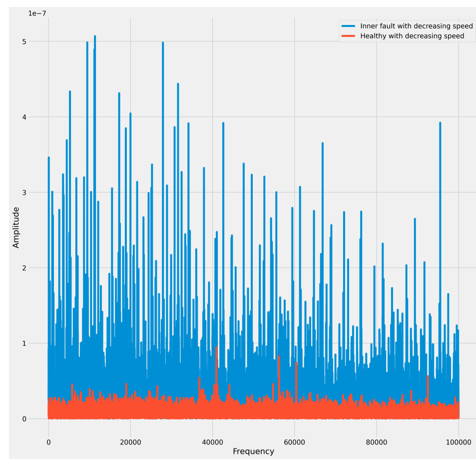

Fig. 12. Amplitude in *frequency domain* for Decreasing Speed of Inner Fault(Blue legend) and Healthy(Red)

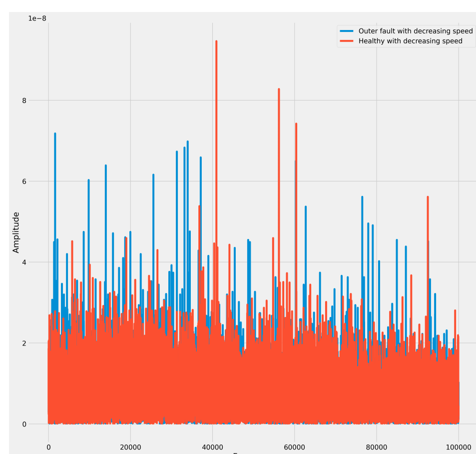

Fig. 13. Amplitude in *frequency domain* for Decreasing Speed of Outer Fault(Blue legend) and Healthy(Red)

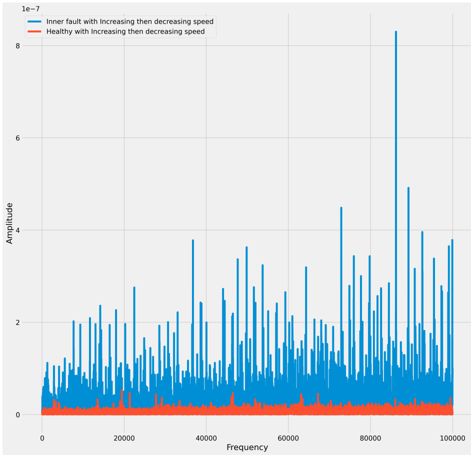

Fig. 14. Amplitude in *frequency domain* for Decreasing Speed of Inner Fault(Blue legend) and Healthy(Red)

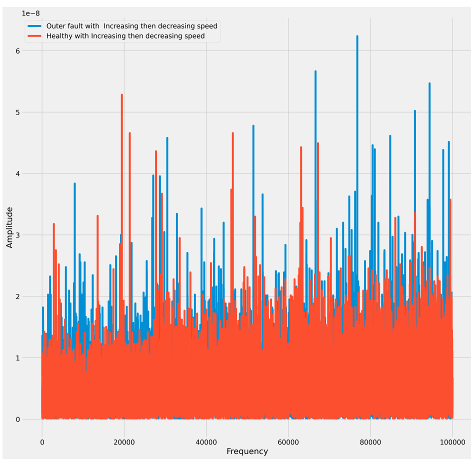

Fig. 15. Amplitude in *frequency domain* for Increasing then decreasing condition of Outer Fault(Blue legend) and Healthy(Red)

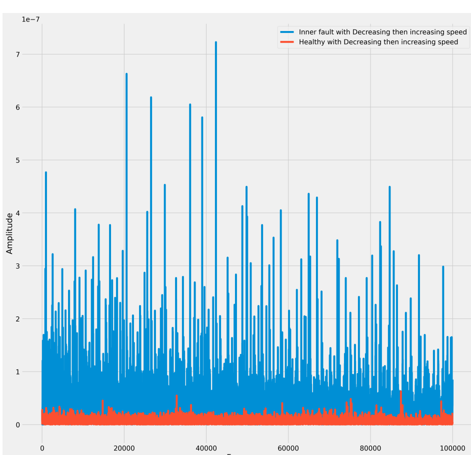

Fig. 16. Amplitude in *frequency domain* for Decreasing then increasing condition of Inner Fault(Blue legend) and Healthy(Red)

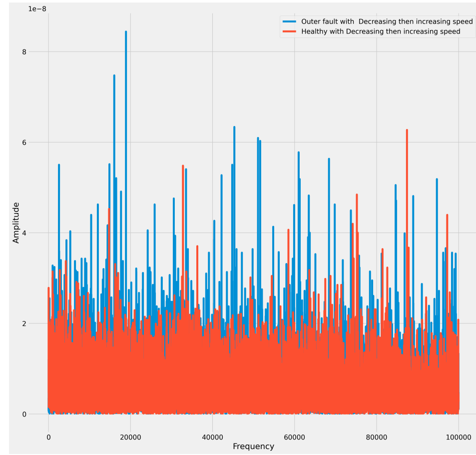

Fig. 17. Amplitude in *frequency domain* for Decreasing then increasing condition of Outer Fault(Blue legend) and Healthy(Red)

the data. A fairly wide set of statistical features viz. mean, Standard Deviation, peak-peak amplitude, energy, entropy, crest factor, kurtosis, skewness, margin factor, impulse factor, variance, RMS were calculated from the accelerometer data using statistics as depicted in the table I.

TABLE I
STATISTICAL FEATURES
(CONDITION MONITORING PARAMETERS)

Parameter	Formula
Mean	$\bar{y} = \frac{1}{N}\sum_{k=1}^{N} y_k$
Root Mean Square (RMS)	$y_{rms} = \sqrt{\frac{1}{N}[\sum_{k=1}^{N}(y_k)^2]}$
Standard Deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \bar{y})^2}$
Kurtosis	$K = \frac{\sum_{k=1}^{N}(y_k-\bar{y})^4}{N\sigma^4}$
Skewness	$S_k = \frac{\sum_{k=1}^{N}(y_k-\bar{y})^3}{N\sigma^3}$
Peak to Peak (pk-pk)	$y_{pk-pk} = y_{max} - y_{min}$
Crest Factor	$C = \frac{\lvert y_{peak}\rvert}{y_{rms}}$
Shape Factor	$W = \frac{y_{rms}}{\frac{1}{N}\sum_{k=1}^{N} y_k}$
Impulse Factor	$I = \frac{max\lvert y_k\rvert}{\bar{y}_{abs}}$
Margin Factor	$L = \frac{max\lvert y_k\rvert}{\left(\frac{1}{N}\sum_{k=1}^{N}\lvert y_k\rvert^{\frac{1}{2}}\right)^2}$
Energy	$\sqrt{\sum_{k=1}^{N} \mid y_k \mid^2}$

Characteristics of statistical features for vibration signals:
Standard Deviation: It measures the power or energy content of the vibration signal that shows deterioration in the health of bearing.
Root Mean Square (RMS): It characterizes the detection in unbalanced rotating elements by evaluating power content of signal. But it does not provide any information on incipient defect stage while it increases with the development of effect.

Crest Factor: It detects damage in a rotating machine at an early stage by measuring the spikiness of the vibration signal. It estimates the impact occuring during the rolling element and raceway contact.
Kurtosis: Bearings in faulty condition gives high value of kurtosis due to spiky or impulsive peaks whereas it is low for healthy bearings.
Impulse Factor: Like crest factor, it is also used to estimate how much impact bearing fault produces.

V. Applied Models

We have implemented Support Vector Classification and Random Forest Classifier, Artificial Neural Network. We have also implemented an ensemble of these three models.

A. *Support Vector Machine (SVM)*

The basic principle of SVM is the search for optimal separating hyperplane so that the classification problem becomes linearly separable. For two dimensions, a line is the hyperplane. For a space of n dimensions we have a hyperplane of n-1 dimensions separating it into two parts. According to the SVM algorithm, the points closest to the line from the classes are support vectors. The distance between the line and the support vectors is the margin. The hyperplane for which the margin is maximum is the optimal hyperplane. C and gamma are the tuning parameters that is passed while creating the classifier.
C: C controls the trade off between smooth decision boundary and classifying the training points correctly. Large value of c will get more intricate decision curves trying to fit in all the points. Different values of c need to be checked for getting a perfectly balanced curve and avoid over fitting.
Gamma: Gamma defines how far the influence of a single training example reaches. For high values of gamma, the decision boundary is dependent on the closer points and ignore the far points. Closer points get more weights resulting in a more wavy curve. For lower values of gamma, even the far points get considerable weight leading to a more linear curve.
Advantages: SVM can efficiently handle non-linear data using the kernel trick. It is very efficient even with high dimensional data. It works relatively well when there is a clear margin of separation between classes. Both classification and regression can be done.
Disadvantages: SVM algorithm is not suitable for large datasets because take long training time on large sets. It does not perform very well when the data set has more noise i.e. target classes are overlapping. That's why we have considered our batches separately for fault detection data. It is not a probabilistic model so the classification cannot be explained in terms of probability. Choosing an appropriate kernel is a difficult task.
The feature matrix was calculated for two different combination of training and testing data. a) Training data extracted in batches from Trial 1, and testing data from Trial 2, b) Training data extracted in batches from Trial 2 & 3, and testing data from Trial 1. The input matrix for SVM with 9 features was reduced to 4 features using Principal Component Analysis (PCA). This was trained for different values of the tuning parameters (C, gamma, kernel). *Table III* shows different accuracy scores on the testing data for both the combinations.

TABLE II
Description of two cases a and b

Case	# Training sample	# Testing samples	# Classes	# Features
a	12000	12000	3	9
a	24000	12000	3	9

TABLE III
Accuracy score of SVM with different parameters (C, gamma, kernel)

	(1000, 0.01, sigmoid)	**(1000, 0.0001, linear)**	**(1000, 0.0001, rbf)**
a	0.92	0.81	0.64
b	0.82	0.74	0.73
	(1, 0.12, rbf)	**(93, 0.01, sigmoid)**	**(1000,1, poly)**
a	0.97	0.96	0.57
b	0.61	0.43	0.59

B. *RFC*

Random Forest Classifier is considered as an extended version of decision tress in which we train the data for more than one decision trees and then we assign the class to the data having maximum number of votes. The parameters which are used in RFC are as follows:
Maximum features: These are maximum number of features that can be taken in Random Forest. 12 features were calculated for the data. The maxim number of features are generally calculated by taking square root, log2 or 0.x times which means x percentage of the number of features available. Increasing maximum number of features increases higher options available but decreases the diversity and also the speed of algorithm.
Number of trees: The more number of trees more the better is the performance and the slower is the code.
Minimum sample leaf size: Leaf is the last node of the decision tree. Leaf size means number of data that can be present at the leaf node. Lesser the size of leaf more is the variance and more prone to capture noise in the data. It also leads to over fitting of the model.
Maximum depth of tree: Higher the depth of tree the more it splits into different nodes and the more information it captures about the data. The model leads to over fitting for large depth values so optimal value needs to be selected for best performance.
Minimum number of samples to split: It constrains the model by limiting minimum number of samples required to split at internal nodes. This model can lead to under fitting because of not allowing the node to split when required.
Criterion: It is the function to measure quality of slit. Common criteria used are gini impurity and information gain. For training the data, we used number of trees, criterion,

maximum depth of tree, maximum features and minimum sample leaf size as our parameters. For different values of these parameters an accuracy of 98% is achieved which would vary generally between 97.5% to 98.1%. For inner fault 100% accuracy is achieved whereas for healthy and outer fault condition 96.85% and 97.15% of accuracy is achieved.

C. Artificial Neural Network

An artificial neural network is essentially comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node in a layer is connected to every node of the previous and next layer. Each connection between an input and an output node is assigned a weight which represents its relative importance.

Commonly used libraries for implementation of artificial neural network algorithm are TensorFlow and PyTorch. They handle all the tasks such as defining the layers and network architecture, activation functions, loss functions, optimiser, updating the parameters using backpropogation, etc

Activation function: It defines the output of that node corresponding to a set of inputs. Important use of Activation function is to introduce non linearity in the neural network. It decides whether a node should be activated or not. Ex. tanh, ReLU, sigmoid, etc

Loss function: It calculates the difference b/w the output and target values. Ex. CrossEntropyLoss, MSELoss, NLLLoss, etc.

Optimiser: It takes in the existing parameters, uses the learning rate and various other possible parameters as per requirement and performs the update of parameters. It updates the model in response to loss function. Ex. SGD, Adam, Adamax, etc.

Learning rate: It is a hyper-parameter which determines the step size while approaching a minimum. It controls how much to change the model each time the gradients are calculated. It may be the most important hyper-parameter. There is always a trade-off between the learning rate and overshooting. Too small values of learning rate may result in a long training process while a large value may produce sub-optimal results due to over-crossing of minima.

Dropout Layer: Using dropout layers is a computationally cheap and an effective way of regularising a neural network to prevent overfitting and improve generalisation to unknown dataset. During training certain nodes are randomly ignored or dropped out, thus effectively making the layer with different number of nodes.

We have implemented the artificial neural network using PyTorch library. It consists of an input layer, three hidden layers and an output layer (Table IV). The activation functions used over the hidden layers is ReLU due to its faster convergence. With the use of other activation functions viz tanh, sigmoid similar results were being obtained. The Optimizer used is SGD (implements Stochastic Gradient Descent). Loss function taken in use is CrossEntropyLoss which combines LogSoftmax and NLLLoss in one single class. From the literature review of ANN implemented previously in this field and our implementation of the neural network, we arrived at the following architecture of the Neural Network (Table IV)

Different learning rate was tried to ensure a good convergence and high accuracy. A learning rate of 0.005 has finally been used. Dropout layer has been introduced to prevent the overfitting over training points.

TABLE IV
NEURAL NETWORK ARCHITECTURE

Layer	# Input	# Output	Activation function
Input layer	-	12	-
Hidden layer 1	12	32	ReLU
Hidden layer 2	32	128	ReLU
Hidden layer 3	128	32	ReLU
Output layer	32	3	-

First we tried to achieve the overfitted neural network to guage the parameters. Different learning rate was used to ensure a good learning rate and high accuracy. A dropout layer was then introduced to prevent the overfitting.

Result: An overall accuracy of 95 percent is achieved using Artificial Neural Network with the distribution shown in Table V.

TABLE V
RESULT(ANN)

Class	Percentage accuracy	#correct/#total
Healthy	93%	3759/4000
Inner fault	100%	4000/4000
Outer fault	91%	3659/4000
Overall Accuracy	95%	11418/12000

D. ENSEMBLE

An ensemble classifier trains on ensemble of numerous models and predicts an output based on the most probable chosen class. It simply predicts the output class based on the majority of predicted class using voting. The idea is to create separate models and predict output based on the combined majority of output of each of the models. Voting Classifier are of two types-

The basic principle of Ensemble is the search for optimal separating

Hard Voting: In hard voting, the predicted class is the class with the majority of votes from the separate models. The class which has the highest probability of being predicted by separate classifiers gets predicted.

Soft Voting: In soft voting, the predicted class is the class with the maximum average probability given to that class by all classifiers.

For example, suppose for a binary classification, the probabilities of correctly predicting a negative class by three different classifiers are 0.45, 0.45 and 0.90. In hard voting, the ensemble would give voting against the negative class as the two classifiers favour the positive. In soft voting, the ensemble gives the average of the probabilities, which is 0.60 and hence

this ensemble would also predict the positive class.
We have used hard voting ensemble to predict the output and obtained the accuracy score of 98.09 percentage. *Table II* shows the confusion matrix of the ensemble for healthy, inner race fault and outer race fault respectively.

TABLE VI
CONFUSION MATRIX OF ENSEMBLE CLASSIFIER

3949	0	51
0	4000	0
178	0	3822

E. *Results and Conclusion*

Highest accuracy is achieved using Random Forest Classifier (98%) and Ensemble (98%), followed by Artificial Neural Network (94%) and Support Vector Classification (82%) respectively. 100% accuracy is achieved for inner race fault which could also be validated by plotting graphs in time-frequency domain as well as the features like mean and standard deviation of healthy, inner race defect and outer race defect. The wrong predictions for healthy goes into outer race and the wrong predictions for outer goes into healthy. Healthy and outer race defect does not have 100% accuracy due to overlapping of data and noise present in the data. In the graphs we could see sudden change in data points, which could be due to noise (white noise, brown noise, etc.) coming from the sensor while recording the data. In our model the noise has not been accounted to make our model robust and save computation time.

TABLE VII
RESULT

Model	Percentage accuracy
Support Vector Classification	82%
Random Forest Classifier	98%
Artificial Neural Network	94%
Ensemble	98%

RFC could be implemented over the real-time data in machines. The prediction could be done for one second data as for the data we will have 2 lacs of data points i.e. 100 batches. If the model even predicts 90% accuracy (less than what we got) i.e. 90 times healthy and 10 times unhealthy or vice-versa then we could say the health status of machine with full surety. Therefore, we could consider 1 second as minimum period for which the data would be required to predict health status of machine. Decreasing the duration of data would decrease surety and vice-versa. Increasing the frequency of sensor would reduce time taken for the model to predict the accuracy. Therefore, frequency of sensor plays a major role in determining speed of prediction.

F. *Acknowledgement*

We would like to thank Prof. Amit Sethi (Department of Electrical Engineering, IIT Bombay) for teaching us machine learning concepts and Prof. Siddharth Tallur (Department of Electrical Engineering, IIT Bombay) and his students: Vaibhav Malviya (B.Tech) and Indrani Mukherjee (M.Tech) for guiding us in the project. We would also like to thank "Mendeley Data, ELSEVIER" for providing the data.

TABLE VIII
BEST ACCURACY ACHIEVED
OUT OF ALL 4 MODELS

Class	Percentage accuracy	Model
Healthy	96.85%	RFC
Inner fault	100%	RFC, ANN
Outer fault	97.15%	RFC

G. *Statement of Contribution*

The project involved these parts and following represent the corresponding team members who worked on the same.
Data visualisation and Feature engineering: Time-domain: Love Kush, Frequency-domain: Rishabh, Variation of Vibration data with variation in rotational speed: Rishabh, Literature Review: All have contributed in literature review Feature extraction and visualisation: Shantanu.
Model Implementation: Support Vector Classification: Harshita, Random Forest Classifier: Lovekush, Artificial Neural Network: Shantanu, Ensemble: Rishabh.
Report making: All have contributed to the report corresponding to their work done in the project.

REFERENCES

[1] Li, X., Zhang, W., Ding, Q., & Sun, J. Q. (2018). Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. Journal of Intelligent Manufacturing,. https://doi.org/ 10.1007/s10845-018-1456-1.
[2] Goyal, D., & Pabla, B. (2015). Condition based maintenance of machine tools—A review. CIRP Journal of Manufacturing Science and Technology, 10, 24–35.
[3] Support vector machines based non-contact fault diagnosis system for bearings https://doi.org/10.1007/s10845-019-01511-x
[4] Konar, P., & Chattopadhyay, P. (2011). Bearing fault detection of induction motor using wavelet and support vector machines (SVMS). Applied Soft Computing, 11(6), 4203–4211.
[5] Zhang, Z., Wang, Y., & Wang, K. (2013). Fault diagnosis and prognosis using wavelet packet decomposition, fourier transform and artificial neural network. Journal of Intelligent Manufacturing, 24(6), 1213–1227.
[6] C. Junsheng, Y. Dejie, and Y. Yu, "A fault diagnosis approach for roller bearings based on EMD method and AR model," Mechanical Systems and Signal Processing, vol. 20, no. 2, pp. 350– 362, 2006
[7] Lebaroud, A., & Clerc, G. (2008). Classification of induction machine faults by optimal time–frequency representations. IEEE Transactions on Industrial Electronics, 55(12), 4290–4298.
[8] Niu, G., Singh, S., Holland, S. W., & Pecht, M. (2011). Health monitoring of electronic products based on mahalanobis distance and weibull decision metrics. Microelectronics Reliability, 51(2), 279– 284.
[9] Wu, S.-D., Wu, C.-W., Wu, T.-Y., & Wang, C.-C. (2013). Multi-scale analysis based ball bearing defect diagnostics using mahalanobis distance and support vector machine. Entropy, 15(2), 416–433.
[10] Q. Hu, S. Zhang, and S. Yang, "Variable condition bearing fault diagnosis based on time-domain and artificial intelligence," Applied Mechanics and Materials, vol. 203, pp. 329–333, 2012.
[11] B. Sreejith, A. K. Verma, and A. Srividya, "Fault diagnosis of rolling element bearing using time-domain features and neural networks," in Proceedings of the IEEE Region 10 Colloquium and 3rd International Conference on Industrial and Information Systems (ICIIS '08), pp. 1–6, Kharagpur, India, December 2008.

[12] Z. Jiang, W. Jiao, and S. Meng, "Fault diagnosis method of time domain and time-frequency domain based on information fusion," Applied Mechanics and Materials, vol. 300, pp. 635–639, 2013.
[13] A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing-https://doi.org/10.1016/j.neucom.2018.05.002
[14] M. Cao, H. Pan, and X. Chang, "Research on automatic fault diagnosis based on time-frequency characteristics and PCASVM," in Proceedings of the 13th International Conference on Ubiquitous Robots and Ambient Intelligence, (URAI '16), pp. 593–598, Xi'an, China, August 2016.
[15] Application of EMD and ANN for automatic bearing fault diagnosis-http://refhub.elsevier.com/S0925-2312(18)30523-X/sbref0009
[16] Bearing vibration data collected under time-varying rotational speed conditions-https://www.sciencedirect.com/science/article/pii/S2352340918314124?via
[17] Bo-Suk Yang, Xiao Di1 and Tian Han, "Random forests classifier for machine fault diagnosis"-https://link.springer.com/content/pdf/10.1007/s12206-008-0603-6.pdf
[18] Shukla S, Yadav RN, Sharma J and Khare S 2015 Analysis of statistical features for fault detection in ball bearing. In: 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) pp. 1–7
[19] Ince, T.; Kiranyaz, S.; Eren, L.; Askar, M.; Gabbouj, M. Real-time motor fault detection by 1-D convolutional neural networks. IEEE Trans. Ind. Electron. 2016, 63, 7068–7075.
[20] He, M.; He, D. Deep learning based approach for bearing fault diagnosis. IEEE Trans. Ind. Appl. 2017, 53, 3057–3065.
[21] Wang, L.H.; Zhao, X.P.; Wu, J.X.; Xie, Y.Y.; Zhang, Y.H. Motor fault diagnosis based on short-time fourier transform and convolutional neural network. Chin. J. Mech. Eng. 2017, 30, 1–12.
[22] Ding, X.; He, Q. Energy-fluctuated multiscale feature learning with deep ConvNet for intelligent spindle bearing fault diagnosis. IEEE Trans. Instrum. Meas. 2017, 66, 1926–1935.
[23] Zhao, M.; Kang, M.; Tang, B.; Pecht, M. Deep residual networks with dynamically weighted wavelet coefficients for fault diagnosis of planetary gearboxes.IEEE Trans. Ind. Electron. 2018, 65, 4290–4300.
[24] Guoqiang Li, et al., "Sensor Data-Driven Bearing Fault Diagnosis Based on Deep Convolutional Neural Networks and S-Transform"