Signal Processing Data Mining for Predictive Maintenance

Sruthi Keerthana Nuttakki

Ira A Fulton Engineering
Arizona State University
Tempe, Az, USA

ABSTRACT

This project focuses on Predictive Maintenance (PdM) applied to a bearing dataset, comprising signal snapshots recorded by the NI DAQCard 6062E. PdM leverages time-series data to anticipate machinery issues before they manifest as critical failures, enabling data-driven maintenance that significantly reduces downtimes and costs associated with equipment damage and repairs. Unlike reactive or preventive maintenance, which either responds after failure or schedules maintenance irrespective of actual machinery conditions, PdM optimizes resource utilization and prolongs equipment life.

CCS CONCEPTS

• Information systems → Data mining • Computing Methodologies → Feature Selection • Computing Methodologies → Modeling methodologies • Applied Computing → Forecasting

KEYWORDS

Predictive maintenance, signal processing, Neural Network, CNN, LSTM, RNN

1 Introduction and Related Work

The project uses NASA's IMS Bearing dataset which has three sets amounting to 2,156 files out of which the project utilizes Set No. 2 with approximately for further exploration, processing, transformation, analysis and investigation and the Set No. 2 contains 984 ASCII files when a test-to-failure experiment was conducted it is revealed to have defects in bearing 1 in the outer race [1] when kurtosis measure is performed after denoising and decomposing into a wavelet filter. Although the wavelet filter method paved the way for predictive maintenance, it requires high computational power in addition to varying parameters of wavelet which is not suitable for a real-time application.

Off-late, various hybrid models are devised to support advanced analytics. In this approach parameters present in 2 or more algorithms are combined and implemented to develop a robust system which is accommodating to real-time data and varying datasets which is computationally difficult to achieve by a standalone algorithm. Hybrid models yield better performance although the initial setup might be on a higher end in comparison to a single model. For a time-series analysis machine learning model can be combined with generic statistical methods for a better accuracy in predictions. Hybrid models also reduce bias of a standalone algorithm mitigating errors by generalizing the model. Survey in [2] presents state-of-art techniques and Deep Learning models such as CNN, LTSM, RNN and LTSM with CNN and RNN which are evaluated on the basis of anomaly detection, failure prediction, and time-to-failure estimation. Although it identifies a plethora of key challenges, the limitations of standalone models are not contrasted to hybrid approaches using real-time data.

Leveraging LSTM networks for building a model allows for an automatic feature extraction which can be trained to detect patterns in a time series dataset. Citations [3,6], delve into LSTM implementation that enhances model accuracy, optimized feature extraction, practicality in PdM has its challenges with computational efficiency and adaptability to varying datasets. In recent times Recurrent neural networks are coming into limelight as they enable working in complex patterns in the data. RNNs have activation functions for inducing non-linearity. The papers [4,5,7] corroborates high accuracy in fault classification and forecast rate. It is to be noted that while implementing RNNs the dataset needs to be chosen appropriately which narrows down the generalization as it requires sequential data.

A Convolutional Neural Network (CNN) is an extension to Artificial Neural Network which has different kernels to extract features. CNN model learns subtle changes in patterns during training [10] projects results of CNN along with Fuzzy logic controller and PCA for fault prediction and contrasts to EEMD method where the CNN takes lesser time for fault diagnosis and distinguishes. The shortcomings of this model relates to the details of pre-processing and data transformation which is automated in its functionality providing less flexibility.

The following report is organized starting with processing and feature extraction followed by model development, training, evaluation, testing, corresponding results and conclusion in the upcoming sections.

2 Processing and Feature Extraction

Fast Fourier Transform is an algorithm which is helpful in signal processing. In this project FFT is implemented to obtain frequency domain features from time domain on each bearing data. Peak frequency is computed to identify patterns and distinguish different signals of the bearings. To process it further this project also employs a tailored bandpass filter which attenuates signals outside the range of the filter. Wavelet transformation is performed to reduce the dimensionality of the refined signal to extract features such as entropy and skewness.

2.1. Initial Investigation of Data and EDA

Each file in the dataset has 20,480 data points which are consolidated into a single dataframe and are indexed to a date time format for readability

Table 1: First 4 values of the four bearings from the consolidated data frame

Timestamp	Bearing _1	Bearing _2	Bearing _3	Bearing _4
2004-02-12 10:32:39	0.05833	0.07183	0.08324	0.0430
2004-02-12 10:42:39	0.05899	0.07400	0.08443	0.04454
2004-02-12 10:52:39	0.06023	0.07422	0.08392	0.04444
2004-02-12 11:02:39	0.06145	0.07384	0.08445	0.04508

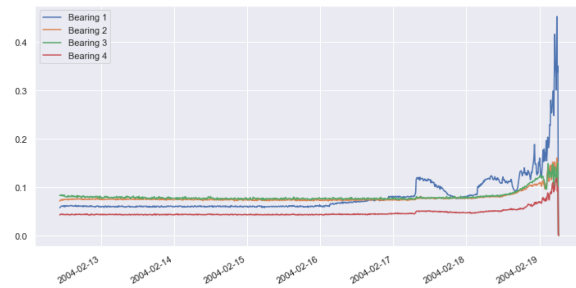

Figure 1: Plotting of vibration signal against time of the four bearings

Summary statistics of the merged data such as mean, standard deviation, minimum and maximum values of the bearings are computed.

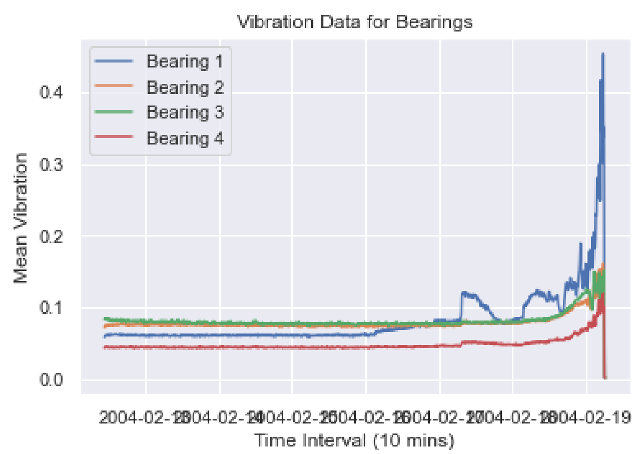

Figure 2: Mean values of bearings against time

In order to detect outliers in the data a box-plot is used. Outliers, are the data points that show irregularities and a box plot visualizes the extremities when the points are present outside the whiskers

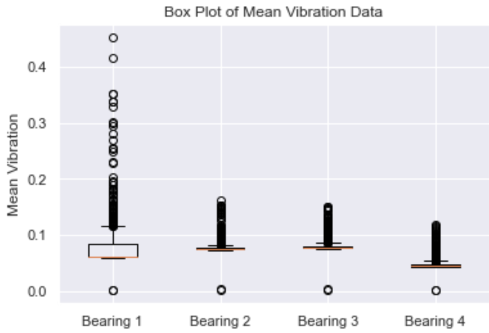

Figure 3: Box-plot of mean vibration of bearings depicting outliers in Bearing 1.

From the above figure, it’s evident that as bearing 1 has points lying outside the whiskers of the plot it has outliers corroborating the result of the test-to-failure experiment.
By using z-score with a threshold value of 3 points an estimate of outliers was obtained.
Z-score is a statistical measure used to determine the standard deviation of the data point from the mean
$Z = (raw\ data - population\ mean)/standard\ deviation$
By setting the threshold value as three there are 42 outliers in the consolidated dataset.

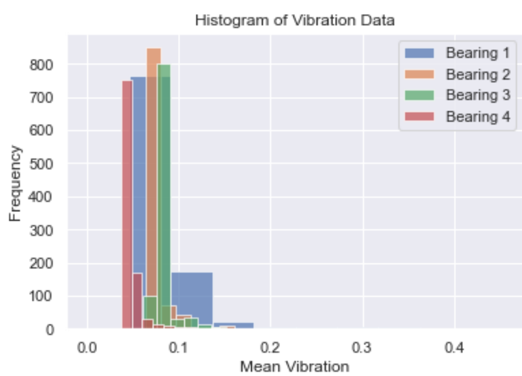

Figure 4: Histogram of mean vibration of bearings against frequency.

From the above histogram it can be seen that there are similar vibration characteristics as they have overlapping distributions. The bearings generally experience low mean vibration levels, suggesting they are in relatively stable condition.However, the slight differences in their distributions could point to early signs of wear or varying operational conditions for specific bearings.

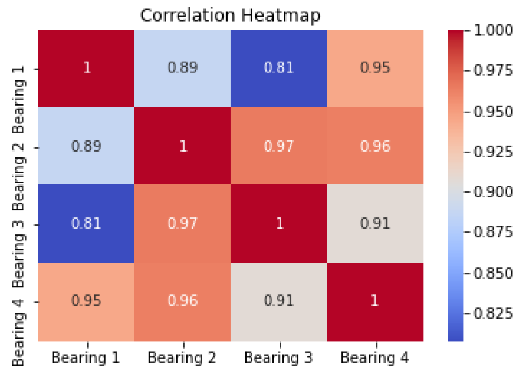

Figure 5: Correlation heatmap consisting of mean vibration values amongst the bearings.

The strong correlation between bearings 2, 3 & 4 suggests that the vibration pattern of these might be influenced by similar conditions and we can validate this with other test sets.

2.2. Feature Extraction

	mean	std	skew	kurtosis	max	min	\
Bearing 1	0.080905	0.040171	4.187934	24.727763	0.453335	0.001168	
Bearing 2	0.078532	0.011779	3.106714	19.736684	0.161016	0.000767	
Bearing 3	0.081356	0.011596	2.484647	15.261315	0.151299	0.000716	
Bearing 4	0.047822	0.009541	3.729213	20.338313	0.119047	0.001699	

	range
Bearing 1	0.452167
Bearing 2	0.160249
Bearing 3	0.150583
Bearing 4	0.117349

Figure 5: Projection of statistical features of the bearings

The high skew & kurtosis values of both bearings 1 & 4 indicates that there might be some anomalies in those bearings and is to be closely considered for further analysis.

After the above statistical implications the project proceeds with z-score normalization post which frequency features are extracted.
The parameters of Band pass filters are 100 and 9500 Hz (lower and higher cutoff frequencies) with a sampling rate of 20kHz. The upper cutoff frequency is taken slightly higher than the Nyquist frequency.

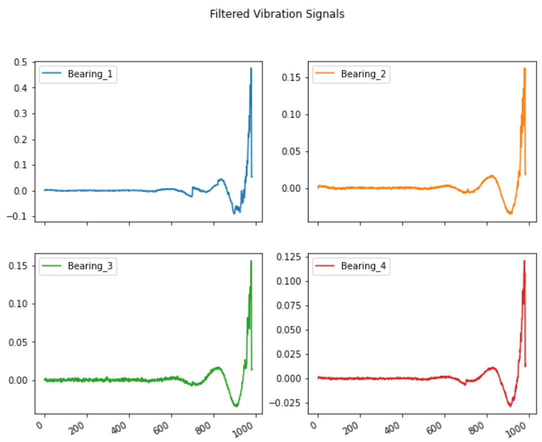

Figure 6: The above figure captures plots of FFT Magnitude Spectrum.

To analyze any low frequency dominance and filtered vibration signals for the bearings. From the previous implementations noise has been removed by attenuating out of range frequencies.

Table2:Following is the tabulation of variance in vibration data of the original consolidated data frame. The variance is pretty insignificant

Bearing 1	0.001614
Bearing 2	0.000139
Bearing 3	0.000134
Bearing 4	0.000091

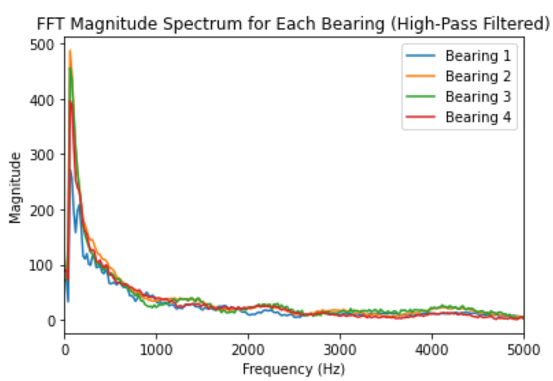

Figure 7: High pass filter magnitude spectrum

The cutoff frequency considered for the high pass filter is 50 Hz with a sampling rate of 20kHz. The FFT plot is derived using the normalized data frame instead of the original concatenated data frame employed on a high pass filter.

As introduced previously, wavelet transformation is performed on the normalized data frame to reduce dimensionality. pywt is a popular library for wavelet transformation Discrete Wavelet Transform is used to extract wavelet features from the normalized dataset. As the data contains both time and frequency domain features, DWT is helpful in decomposing the features and the decomposition level can be chosen as required. This step is essential for detecting anomalies and the extracted features can be further used in PdM tasks. At decomposition level 1 the number of rows are reduced to half of the original data set.

(495, 4)

	Bearing_1_wavelet	Bearing_2_wavelet	Bearing_3_wavelet	Bearing_4_wavelet
0	-0.677205	-0.481266	0.281692	-0.425807
1	-0.731222	-0.492694	0.365259	-0.448326
2	-0.796177	-0.790565	0.240381	-0.688228
3	-0.764490	-0.599558	0.348119	-0.541316
4	-0.694703	-0.466147	0.301266	-0.413816
5	-0.677836	-0.547205	0.277077	-0.486987
6	-0.677225	-0.449108	0.205474	-0.551553
7	-0.689774	-0.368005	0.308878	-0.537835
8	-0.700586	-0.450833	-0.040124	-0.625806
9	-0.654936	-0.384237	-0.141164	-0.524287

Figure 8: First 10 rows of Wavelet transformation for a low frequency approximation.

The results in the above figure gives both frequency and time domain information due to utilization of DFT while the standard Fourier Transform provides only frequency information. By considering low level frequencies the high level frequencies such as sharp noises are disregarded providing a more accurate interpretation of the wavelet.

Following the wavelet transformation, statistical properties such as energy, entropy, skewness, kurtosis and others were extracted from the wavelet coefficients. For example, in the case of Bearing_4, the energy from 813.49 at Level 1 to 1160.36 at Level 3, indicating that the signal contains multiple frequency components. Additionally, the low entropy values, such as 0.8679 at Level 1 and 0.8691 at Level 4, suggest that the signal exhibits periodic behavior, meaning it follows a consistent, recurring pattern over time. These properties are crucial for diagnosing issues and detecting anomalies in the bearing, as they provide valuable insights into the signal's characteristics at different scales.

```
   Bearing_4_energy  Bearing_4_mean  Bearing_4_std  Bearing_4_skew  \
0        813.486202       -0.170499       3.454557        3.558772
1       1119.232059       -0.463083       4.030490        2.657294
2       1160.361526       -0.288696       4.120776        1.808055
3        958.371345       -0.354081       3.737422        3.919192

   Bearing_4_kurtosis  Bearing_4_entropy  Bearing_4_max  Bearing_4_min  \
0           16.887866           0.867902      20.540018      -2.087116
1           19.304599           0.697948      23.448003     -15.898960
2           13.397451           0.765784      21.069723     -16.754054
3           23.855894           0.869145      23.694296     -10.939808

   Bearing_4_range  Bearing_4_rms  ...  Bearing_4_energy  Bearing_4_mean  \
0        22.627134       3.458762  ...         41.397525       -0.002314
1        39.346963       4.057006  ...         96.052120        0.008439
2        37.823776       4.130876  ...         95.977653       -0.001000
3        34.634104       3.754157  ...         55.121522        0.003732

   Bearing_4_std  Bearing_4_skew  Bearing_4_kurtosis  Bearing_4_entropy  \
0       0.289182        0.058189          128.521515           0.246750
1       0.440424        0.613269          130.937859           0.672238
2       0.440333       -1.213584          142.905081           0.140795
3       0.333681        0.581209          163.033437           0.468588

   Bearing_4_max  Bearing_4_min  Bearing_4_range  Bearing_4_rms
0       3.912422      -3.750611         7.663033       0.289191
1       5.927130      -5.834841        11.761971       0.440505
2       5.705554      -6.308618        12.014172       0.440334
3       4.795315      -4.666153         9.461467       0.333701

[4 rows x 50 columns]
```

Figure 9: Wavelet transformed Statistical Features of Bearing_4: Energy, Entropy, Skewness, and Kurtosis Across Levels.

2.3. Data Labelling

From the vibration data for every bearing, the time-domain and frequency-domain features were extracted.The data has been labeled using these attributes, which include RMS, Energy, Entropy, and Kurtosis.The IQR and a multiplier of 1.5 for the aforementioned statistical features have been used to determine the threshold values.The table below shows the class distribution of each of the bearings.

Bearing #	# Healthy	# Faulty
Bearing_1	806	178
Bearing_2	885	99
Bearing_3	893	91
Bearing_4	875	109

3 Classical Machine Learning Models

The goal of our dataset, which included vibration data from four machine bearings, was to determine how well the classical models classified the bearings as either healthy or faulty based on the attributes that were derived from the vibration data that was captured.Since simpler models are always preferable to more complicated ones, the goal of analyzing the classical machine learning models was to determine which of them would be appropriate for a dataset like ours, which is imbalanced in nature. We employed the 5-fold cross validation technique to assess these models rather than a particular train-test split of the dataset because the latter could result in overfitting.

3.1 Logistic Regression

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Logistic regression is a useful analysis method for classification problems, where one is trying to determine if a new sample fits best into a category.

Although the model's accuracy is 98%, it had a problem with "False Positives," which occurred when the model failed to accurately detect the bearings' "faulty" state. This is a critical problem in a dataset where there are few instances of "faulty" data.This misclassification problem has mostly been seen in bearing_4.

3.2 Random Forest

A random forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. Random Forest is considered to be a good algorithm when it comes to binary classification.

This model also demonstrated a 98% accuracy rate, which is similar to that of logistic regression; however, we can observe that the rate of incorrectly classifying "faulty" sections is lower than that of LR, which makes it a preferable option.The ROC demonstrates that here bearing_3 has the maximum number of misclassification while bearing_1 has the least.

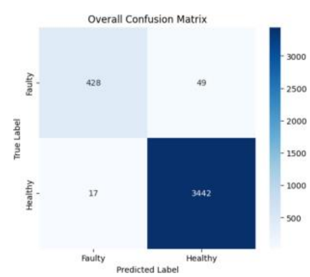

Figure 10: RF Confusion Matrix

3.3 Decision Tree

A Decision Tree is a popular machine learning algorithm used for both classification and regression tasks.

Even though the model showed 99% accuracy, misclassification remained a problem. However, it has been noted that the model is correctly predicting the healthy class in this instance, which was also problematic for logistic Regression and Random Forest.

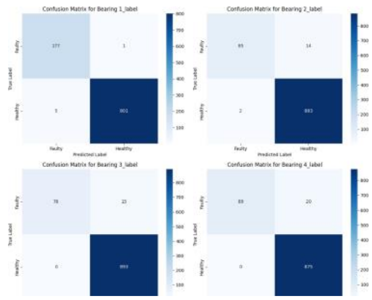

Figure 11: DT Confusion Matrix

3.4 Naive Bayes

Naive Bayes is a machine learning algorithm based on Bayes theorem. Gaussian Naive Bayes is a specific type of the Naive Bayes algorithm designed for classification tasks with continuous features. The scikit-learn library is used to implement the Gaussian Naive Bayes classifier.When compared to the other models, this one's 93% accuracy rate makes it evident that it does not suit our data well.The total confusion matrix shows that the rate of misclassification is rather high.

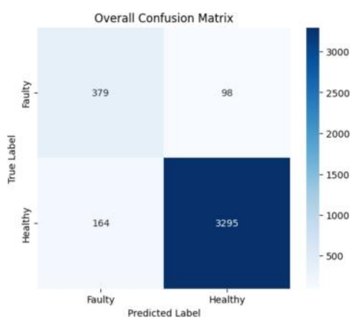

Figure 12: NB Confusion Matrix

3.5 XGBoost

XGBoost is a boosting classification algorithm that helps improve the machine learning model's accuracy. It is a powerful open-source tool that works by combining decision trees and gradient boosting algorithms. It is mainly used to handle large datasets. We tried to implement XGBoost to our dataset to see if we could elevate the model performance.

Even though XGBoost is regarded as a potent technique, we were unable to accomplish our goal of accurately identifying the classes for every bearing. Additionally, this model demonstrated a 99% accuracy rate, resulting in a perfect categorization for bearing_1. For bearing_4, the misclassification is most prevalent.

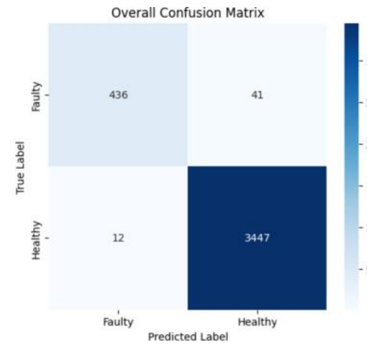

Figure 13: XGBoost Confusion Matrix

3.6 K-Nearest Neighbor (KNN)

KNN is an algorithm used for both classification and regression tasks. In the context of fault prediction, KNN works by classifying a data point based on the majority class of its 'k' nearest neighbors in the feature space. By calculating distances between data points, KNN determines the similarity between instances and assigns labels accordingly. This does not need any training. KNN has the lowest accuracy rate of 91% of all the traditional models tested on this dataset, suggesting that it is not appropriate for our dataset.Additionally, models trained on historical data are a better fit for fault prediction like ours.It is failing to accurately classify the healthy class as well.

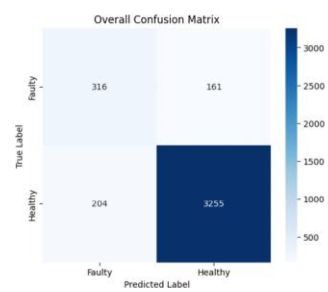

Figure 14: KNN Confusion Matrix

3.7 Support Vector Machine (SVM)

A support vector machine (SVM) is a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space.

This showed a 92% accuracy rate and a high probability of both classes being misclassified. Compared to the other kernels, the "rbf" kernel that we utilized appeared to perform better.

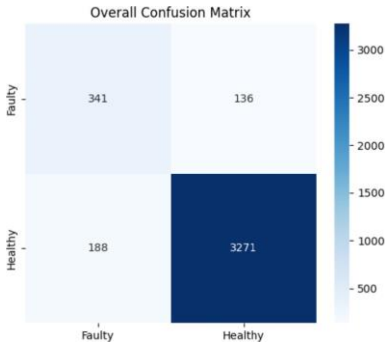

Figure 15: SVM Confusion Matrix

4 ML Model Evaluation summary

The Table demonstrates the model performance

Model	Accuracy	Precision	Recall	F1
LR	98%	98%	99%	99%
KNN	91%	95%	94%	95%
RF	98%	99%	100%	99%
DT	99%	99%	100%	99%
NB	93%	97%	95%	96%
SVM	92%	96%	95%	95%
XGBoost	99%	99%	100%	99%

5 Algorithm choice and Implementation

The training processes for CNN and LSTM models are outlined in the algorithms below. The CNN focuses on learning spatial features from data such as images, whereas the LSTM is intended to grasp sequential patterns in data such as time series. To increase performance, both models use forward and backward passes to modify their weights across different epochs.

Algorithm 1: CNN Model Training

Require: The full datastore *U*, the size limit of the on-device datastore subset *L*, the number of random sampling times *T*, the size of candidate key-value pairs per sampling *B*, and the size of selected pairs per sampling *b*.

Input:

- Training dataset Xtrain
- Labels Ytrain
- Number of epochs E
- Learning rate η
- Batch size BatchSize

Output:

- Trained CNN model

Steps:

1. Initialize CNN model with layers (Cond2D, MaxPooling, Flatten, Dense)
2. For e ← 1 to E do:

   2.1. Shuffle Xtrain and Ytrain

   2.2. For each batch in the training set:

   2.2.1. Perform forward propagation

   2.2.2. Compute the loss L = LossFunction(ytrue, ypred)

   2.2.3. Perform back propagation to update weights using η

3. Return the trained CNN model.

Algorithm 2: LSTM Model Training

Require: The full bearing fault detecting *U,* the size limit of the on-device datastore subset *L*, the number of random sampling times *T*, the size of candidate key-value pairs per sampling *B*, and the size of selected pairs per sampling *b.*

Input:

- Training dataset Xtrain
- Labels Ytrain
- Number of epochs E
- Learning rate η
- Batch size BatchSize
- Number of LSTM units NLSTM

Output: T

- Trained LSTM model

Steps:

1. Initialize LSTM model with layers (LSTM, Dense)
2. For e ← 1 to E do:

2.1 Shuffle Xtrain and Ytrain

2.2 For each batch in the training set:

2.2.1. Perform forward propagation through the LSTM layers.

2.2.2 Compute the loss L = LossFunction (ytrue, ypred)

2.2.3. Perform backpropagation to update weights using η

3. Return to trained LSTM model.

6 Model Development

After the wavelet features decomposition, labels for the bearings are generated in terms of healthy and faulty by using entropy, rms, energy and kurtosis values. These values are taken by employing a cutoff range with IQR to determine the labels.

The generated label file is further split into training, testing and validation data. As per the previous section, LSTM and CNN models are built. The LSTM model is built by considering the parameters Relu, Sigmoid and compiled using adam optimizer. These inputs provided a training accuracy of 99.94% and validation accuracy of 99.28. CNN model is built with relu activation and optimized with adam. The training accuracy is 100% and the validation accuracy is 99.28% for the devised CNN model.Figure 13 clearly demonstrates that the accuracy of the model increases rapidly with increasing epochs and similarly the loss curve decreases.

```
Epoch 23/50
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 1.0000 - loss: 0.0045
Epoch 23: val_loss did not improve from 0.03939
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 1.0000 - loss: 0.0045 - val_accuracy: 0.9928 - val_loss: 0.0574
Epoch 24/50
11/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 1.0000 - loss: 0.0021
Epoch 24: val_loss did not improve from 0.03939
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9994 - loss: 0.0025 - val_accuracy: 0.9928 - val_loss: 0.0578
Epoch 24: early stopping
Restoring model weights from the end of the best epoch: 14.
```

Figure 16: Training and validation accuracies of LSTM

```
Epoch 22/50
 1/18 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - accuracy: 1.0000 - loss: 4.1424e-04
Epoch 22: val_loss did not improve from 0.02587
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 1.0000 - loss: 0.0043 - val_accuracy: 0.9928 - val_loss: 0.0349
Epoch 22: early stopping
Restoring model weights from the end of the best epoch: 12.
```

Figure 17: Training and validation accuracies of CNN

	precision	recall	f1-score	support
faulty	1.00	0.98	0.99	57
healthy	1.00	1.00	1.00	237
accuracy			1.00	294
macro avg	1.00	0.99	0.99	294
weighted avg	1.00	1.00	1.00	294

Figure 18: Classification report of the findings

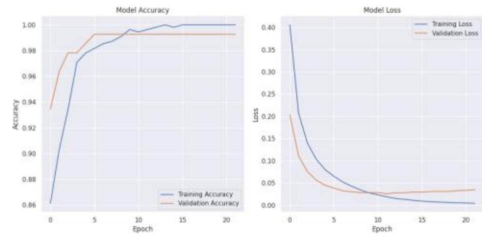

Figure 19: Plots of testing and validation accuracy and loss function

The model when evaluated on test data projected 99.659% accuracy

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9913 - loss: 0.0171
Test Loss: 0.01549276988953352
Test Accuracy: 0.9965986609458923
```

Figure 20: Evaluation of testing data

6.1 RUL Prediction using LSTM

We used the LSTM model to predict RUL of Bearing 1. The model was trained using time-series data, and its performance was measured using Test Loss and Test MAE. The results are as follows:

```
Test Loss: 0.0001499634381616488, Test MAE: 0.008524426259100437
```

Figure 21: Test Loss and Test MAE for LSTM Model (RUL Prediction)

The results indicate that the LSTM model predicted the RUL with reasonable accuracy, while minor errors were noted, particularly as the bearing approached failure.

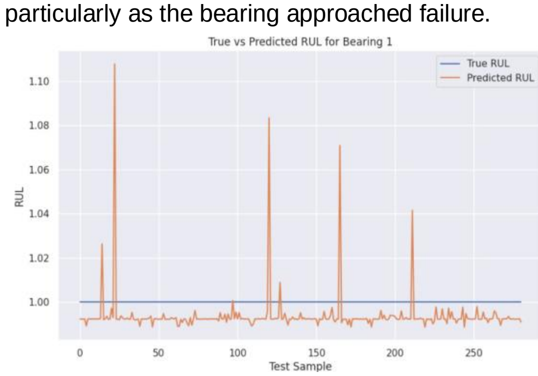

Fig 22: True vs. Predicted RUL for Bearing 1 (LSTM)

6.2 RUL Prediction using Random Forest

We also applied the Random Forest model to predict RUL for Bearing 1. The model was more effective than LSTM at capturing true RUL trends. The Random Forest model's results are as follows:

```
Validation R²: 0.959130764106168
Validation RMSE: 59.212435416650564
Validation MAE: 36.90746376811594
Test R²: 0.9398758835300689
Test RMSE: 67.02507356317223
Test MAE: 42.67503378378378
```

Fig 23: Performance metrics for Random Forest Model (RUL Prediction)

The Random Forest model outperformed other models in predicting RUL, with higher R^2 scores and lower errors.

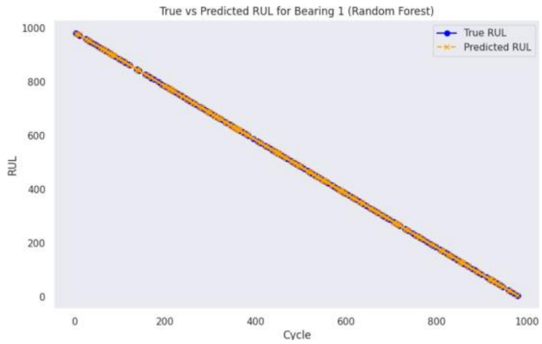

Fig 24: True vs. Predicted RUL for Bearing 1 (LSTM)

6.3 RUL Prediction using CNN-LSTM

Hybrid CNN-LSTM model leverages the feature extraction capabilities of CNNs and the temporal pattern recognition strengths of LSTMs to predict the Remaining Useful Life (RUL) of bearings. The model achieved an R2R^2R2 score of 0.8855

To predict RUL using the hybrid model approach, optimizer Adam with learning rate 0.001 and ReLU activation along with MaxPooling size of 2 for CNN has been implemented.

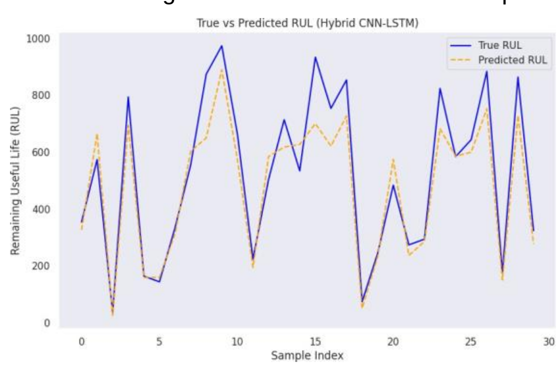

Fig 25: True vs Predicted RUL using CNN-LSTM

7 Conclusion:

Upon exploring both classical ML models and neural networks it is observed that Neural networks (CNN & LSTM) are a better choice when data is imbalanced in nature**.** Also, opportunity lies in exploring different sets of parameters and better fine tuning the classical ML models to test their performance in future.

REFERENCES

[1] Hai Qiu, Jay Lee, Jing Lin. "Wavelet Filter-based Weak Signature Detection Method and its Application on Roller Bearing Prognostics." Journal of Sound and Vibration 289 (2006) 1066-1090
DOI: https://doi.org/10.1016/j.jsv.2005.03.007.

[2] Oscar Serradilla, Ekhi Zugasti, and Urko Zurutuza. 2020. "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospect". 1, 1 (October 2020)
DOI:https://doi.org/10.1007/s10489-021-03004-y

[3] Zhang, S., Zhang, S., Wang, B., & Habetler, T. (2020). Deep Learning Algorithms for Bearing Fault Diagnostics – A Comprehensive Review. TR2020-034.
DOI:https://doi.org/10.1109/DEMPED.2019.8864915

[4] "Prediction of Machine Deterioration Using Vibration Based Fault Trends and Recurrent Neural Networks" - P.W.Tse Department of Manufacturing Engg. D. P. Atherton School of Engineering.
DOI: https://doi.org/10.1115/1.2893988

[5] Kavana V; Neethi M. "Fault Analysis and Predictive Maintenance of Induction Motor Using Machine Learning." 2018 Third (ICEECCOT) 14-15, December 2018.
DOI: https://doi.org/10.1109/ICEECCOT43722.2018.9001543

[6] Dey, D., & Jana, R. (2022). Bearing Fault Predictive Maintenance using LSTM. Proceedings of the 2022 3rd International Conference on Computing, Analytics, and Networks (ICAN), Rajpura, Punjab, India, 1-6. IEEE.
DOI: https://doi.org/10.1115/1.2893988

[7] T. T. Reza, M. M. Arif, and M. S. Islam. Bearing Fault Detection Using an Effective Convolutional Neural Network Architecture. In Proceedings of the 7th International Conference on Mechanical Engineering and Renewable Energy 2023 (ICMERE 2023), Chattogram, Bangladesh, November 16--18, 2023, pp
DOI: https://doi.org/10.1515/mt-2023-0334

[8] Rehab, A., Ali, I., Gomaa, W., and Fors, M. N. 2023. Bearings Fault Detection Using Hidden Markov Models and Principal Component Analysis Enhanced Features. Production Engineering Department, Alexandria University, Alexandria, Egypt, and Department of Industrial and Manufacturing Engineering, Egypt-Japan University of Science and Technology (E-JUST), Alexandria, Egypt.
DOI: https://doi.org/10.48550/arXiv.2104.10519

[9] "Oche A Egaji , 2 Tobore Ekwevugbe, 1 Mark Griffiths - "A Data Mining based Approach for Electric Motor Anomaly Detection Applied on Vibration Data"
DOI: https://doi.org/10.1109/WorldS450073.2020.9210318

[10] Dan Zhang , Senior Member, IEEE, Yongyi Chen , Fanghong Guo , Member, IEEE, Hamid Reza Karimi , Senior Member, IEEE, Hui Dong , and Qi Xuan , Member, IEEE - "A New Interpretable Learning Method for Fault Diagnosis of Rolling Bearings"
DOI: https://doi.org/10.1109/TIM.2020.3043873