Estimating Bearing Fault Type and Depth in Rotary Machines: A Comparison of AI Methods

Shubham Malhotra, Dan Moyer, and Shiva Kumar Tekumatla

Abstract—**Fault type and depth for bearings in rotary machines were estimated from vibration data using linear regression, support vector machines, random forest, and neural network approaches. Each method was used to classify inner race, outer race and rolling element bearing faults. The accuracy of each method for differentiating between bearing fault types and estimating fault depths using vibration data envelope spectrum analysis is compared. A novel combination of envelope spectrum preprocessing with neural networks achieved prediction accuracy exceeding 99% for fault type classification, and 95% for fault depth estimation. Classification accuracy exceeding 95% was also realized using random forest methods. The results indicate that performance could be further improved with a larger neural network and additional training data.**

I. Introduction

Bearing failures are a frequent and costly phenomenon in high-powered industrial rotary equipment. Machine failures can be detected using several methods, and significant effort has been put into standardizing these methods. The primary methods used in industry for periodic monitoring and maintenance include vibration analysis, motor current signature analysis, lubricant analysis, thermography, air leak detection and temperature monitoring. Most of these methods focus on identifying different types of failures and have been effective in helping the maintenance managers improve machine reliability. For detecting most types of mechanical failures in machines, vibration analysis is used extensively [1] and has proven to be one of the most efficient condition monitoring technologies [2].

Typically, vibration analysis is used to detect failures associated with unbalance, misalignment, looseness, rub crack, gear box failure and bearing failure. There are multiple possible underlying causes for these problems in rotary equipment. Uneven distribution of mass on a rotating shaft causes unbalance [3], while poor installation of shafts may cause either angular or parallel misalignment [4]. Low stiffness or loose bolts can result in issues with looseness [5]. Most bearing failures are the result of cavitation and erosion, often caused by the formation and collapse of vapor bubbles in the oil film during rapid pressure changes [6].

Machine vibration data are commonly captured using accelerometers, and signal processing techniques are employed to evaluate the vibration data at various frequencies and troubleshoot machine failure modes. Signal analysis techniques (including Fourier transforms, envelope spectrum, phase analysis, and Nyquist plots) are used by certified professionals to analyze the data and determine overall machine health. These technologies help the analysts identify failures even at the bearing rolling element level. However, these traditional methods require a detailed analysis of the complex data, and most of the results can be interpreted only by experienced engineers. Using existing vibration limit guidelines, certified professionals manually calculate the likelihood of possible failure modes by observing specific frequencies in the spectrum. Most major failure modes (such as unbalance, misalignment and structure looseness) can be identified by observing the frequency spectrum data, while most bearing failures can be detected by studying envelope spectrum data.

Predictive maintenance has seen increasingly wide acceptance because of its efficacy, with recent Industry 4.0 trends accelerating this growth. However, the use of artificial intelligence to analyze the machines' vibration data and predict failures is at an early stage. There have been some attempts within the last decade in this field, and the results from these studies show strong prospects for the role of AI in predictive maintenance.

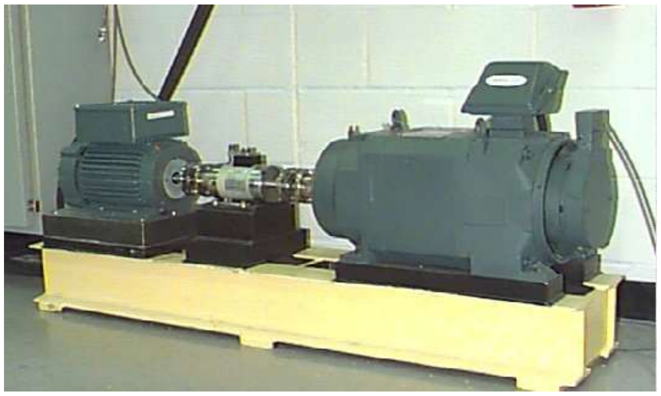

Figure 1: Apparatus used for vibration data capture at the Case Western Reserve University Bearing Data Center

Sampaio *et al* [7] designed an apparatus that can simulate machine vibration data and used various machine learning algorithms to predict machine failures. The authors considered the Fourier transform of the collected vibration data and are successful in achieving an RMSE of 0.0038 using artificial neural networks (ANNs). Kamat *et al* [8] provide a comparative analysis of ANNs and other methods applied to bearing fault data from the Case Western Reserve University Bearing Data Center (CWBDS) [9]. Korba *et al* [10] used empirical mode decomposition on this vibration data and evaluated performance of various machine learning algorithms. Zhang *et al* [11] presented results for several algorithms and discussed some deep learning methods used for bearing failure prediction.

Most of the previous work has focused on detecting the presence or absence of bearing failures, as well as predicting the vibration frequencies associated with these failures. We have not found any previous work wherein the model is used to directly predict the bearing fault depth from the vibration data. Furthermore, there have been no studies on the usage of envelope spectrum of vibration data to predict or classify bearing failures. Envelope spectrum, a frequently used digital signal processing technique in vibration analysis, is easy to use because of the established guidelines based on various bearing types and frequency modes. Multiple types of bearing failure, including inner race faults, outer race faults and rolling element faults, may be detected using envelope spectrum analysis. In this paper, we use envelope spectrum to classify and predict the fault type and depth for CWBDS vibration data.

II. Background

A. Dataset

The Case Western Reserve University Bearing Data Center (CWBDS) provides a series of datasets for vibration monitoring. These include vibration data taken during normal operation, as well as data taken from bearings with different types of faults. The data is gathered from a 2-HP Reliance electric motor with a fan connected to it through 2 bearings: the drive end (DE) bearing and the fan end (FE) bearing, both of which are deep groove ball bearings. The vibration data is captured using accelerometers, with FE sampled at 12000 Hz and BE sampled at both 12000 and 48000 samples/second. Fault depths of 7 mil, 14 mil, 21 mil, 28 mil and 40 mil are induced for each bearing element. Measurements for inner race and rolling element are captured at only one measurement location, whereas the outer race fault data is measured at centered, orthogonal and opposite locations. Data for both the fault modes and normal operation are measured at 4 different loads and speeds. We use these data to train and validate our models.

The data provided by CWBDS includes a total of 162 files, each containing vibration data measured at the drive end and fan end. The raw vibration data is split into multiple sets while keeping the frequency resolution constant to expand the data size and reduce overfitting when training our models. We do this by applying a rolling 1-second window to the data at 0.1 second intervals. This expands the length of the data by a factor of 10 and results in envelope spectrum data with 1 Hz resolution.

As a first step in predicting bearing fault type and depth, we compute the envelope spectrum of the vibration data, which first demodulates the vibration signal by convolution at the rotation frequency before taking the FFT. The envelope spectrum frequency data up to 1800 Hz, a frequency range that encompasses the harmonic frequencies relevant to bearing fault detection, is supplied to the estimation algorithms.

B. Envelope Spectrum Analysis of Vibration Data

Examples of CWBDS vibration data from bearings with and without faults are shown Figures 2 and 3.

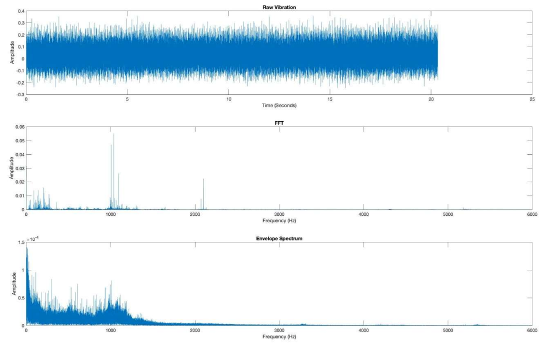

Figure 2: Vibration data, FFT and envelope spectrum of bearings in good condition. This vibration data was collected from a rotary machine where the bearings do not have faults, resulting in relatively few pronounced harmonics in the frequency domain data.

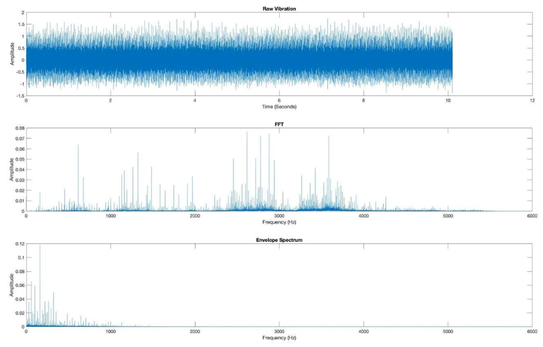

Figure 3: Vibration data, FFT and envelope spectrum of an inner race bearing fault. This vibration data was collected from a rotary machine where an inner race bearing has a fault with 7mil depth, resulting in significant signal power in certain harmonics in the frequency domain data.

To determine whether a bearing failure mode is present for any of the fault types mentioned above, analysis of certain frequencies is required. These frequencies depend on the type of bearing and type of failure to be detected. For example, these frequencies for the bearing 6205-2RS JEM SKF are given in Table 1. The frequencies of interest to detect the inner race, outer race and rolling element failures are given by Equations 1, 2, and 3 respectively. Figures 4 and 5 illustrate the envelope spectrum for rotary machines operating with bearing faults. We propose an automated method to directly estimate the bearing fault depth, eliminating the need for this time-consuming manual analysis.

$$F_i = n * I * Speed \quad (1)$$
$$F_0 = n * O * Speed \quad (2)$$
$$F_r = n * R * Speed \quad (3)$$

Where:

$$n = 1,2,3\ldots$$

I is the inner ring failure frequency
O is the outer ring failure frequency
R is the rolling element failure frequency
Speed is the rate of rotation of the bearing in Hz

Table 1: Bearing fault frequencies for two different types of bearings. Here, 6205-2RS JEM is used at drive end, and 6203-2RS JEM is used at the fan end.

Bearing	Inner Ring	Outer Ring	Rolling Element
6205-2RS JEM SKF	5.4152	3.5848	4.7135
6203-2RS JEM SKF	4.9469	3.053	3.9874

The values in this table are used to calculate which frequencies should be examined during failure analysis for these bearings.

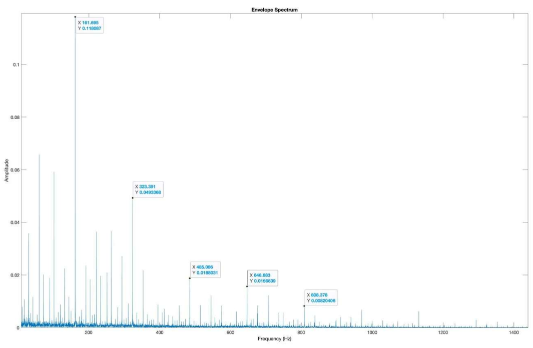

Figure 4: Envelope spectrum of a bearing with inner race failure. This failure data captured while the bearing is running at 1797 rpm. The pronounced peaks at frequency harmonics of the inner ring frequency indicate inner race bearing failure.

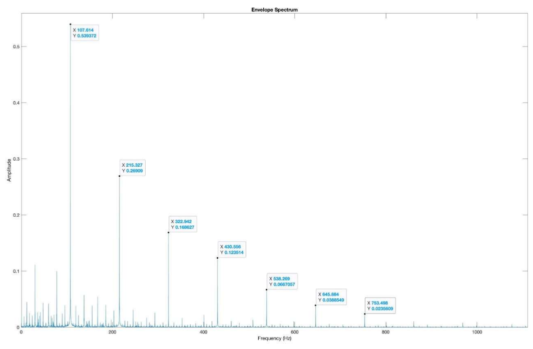

Figure 5: Envelope spectrum of a bearing with outer race failure. This failure data is captured while the bearing is running at 1797 rpm. The pronounced peaks at frequency harmonics of the outer ring frequency indicate outer race bearing failure.

C. Approach

Previous work has primarily focused on detecting the presence of bearing failures, as well as the vibration frequencies associated with these failures. We have not found any previous work wherein the model was used to predict the bearing fault depth directly from this vibration data; thus, we believe our approach of estimating bearing fault depth from vibration data is novel. Furthermore, we have not found any prior work on automated bearing fault detection or estimation algorithms that pre-process the data using the envelope spectrum. Because it can provide insight into multiple bearing failure modes— including inner race failure, outer race failure and rolling element failure—we exclusively use envelope spectrum data as the input for each method we explore here. ***Our novel contributions include the use of the envelope spectrum for vibration data pre-processing, as well as directly predicting the bearing fault depth.***

We test several methods to automate failure prediction and compare the performance of each. First, we apply linear regression to estimate the fault depth from a subset of the envelope spectrum data consisting of the harmonics of the frequencies of interest for each bearing fault type under the conditions present when the vibration data was collected. This method has the advantage of being simple to calculate, as well as being robust to overfitting since the number of model parameters is low. Second, we apply support vector machine and random forest classifier methods. These methods have a much greater ability to represent complex relationships within the data while still being resistant to overfitting. Third, we train a neural network using popular methods (including rectified linear unit activation functions and the *Adam* adaptive learning rate optimization algorithm)[10][12]. We leverage existing deep learning software packages (TensorFlow and Keras) to provide a baseline for what can be accomplished with state-of-the-art neural network algorithms.

Finally, we test a neural network using a novel method of calculating an input-data-dependent optimal learning rate. Neural networks are general function approximators and can be trained to learn highly nonlinear functions in high-dimensional state-space. The amount of computation required to train a neural network is heavily dependent on the learning rate. Several strategies for selecting either a fixed or adaptive learning rate have been proposed [13][14][15]. For problems requiring output over a continuous interval, the Least Mean Squares (LMS) filter algorithm can be thought of as a simplified single neural net node with a linear activation function [16]. The fastest stable learning for the LMS algorithm occurs when the learning rate is equal to the reciprocal of the sum of the squares of the inputs: $\frac{1}{\sum x^2}$. At this learning rate, the post-update error is reduced to zero, and the change in coefficients (i.e., neural net weights) is the minimum RMS distance in state space that fulfils this condition. Effectively, this learning rate changes the network weights as little as possible while making the output equal to the desired value for that single input. We apply this same method of calculating the learning rate to each neural network node. While this may initially seem computationally expensive for a large network, we propose a method wherein most of the calculation is already completed during the normal course of the backpropagation update.

III. Methods

A. *Linear Regression*

Because linear regression is limited to output that is a linear combination of the input features, we performed additional pre-processing that was not done for the other methods. For each data element, the frequencies of interest for each bearing fault were calculated as described previously. The envelope spectrum data for only these frequencies of interest, as well as their first 10 harmonics, were extracted and used as features for the linear regression model. As a result, different frequencies were selected depending on the rate of rotation of the bearing for each data element. This produced 30 input features for each data element, which were used to predict the fault depth.

The data were randomly shuffled before splitting the data into training and testing data sets to ensure there was no difference in fault type distribution between the training and testing data sets. Linear regression was performed on the training data set in Python using the NumPy library's linalg.lstsq() function. We then tested the performance on the reserved test data set and calculated the percentage of predicted depths within each of several tolerance bands.

B. *Support Vector Machines*

Envelope spectrum data from the raw vibration data is tested with various SVMs. For this problem, the Support Vector Machines classify the data into normal mode and three failure condition modes. Three failure condition modes are bearing inner race failure, outer race failure, and rolling element failure. The data is sampled for all these failure modes at different depths, loads and speeds. This data is split into 80 percent training data and 20 percent test data before being tested with SVMs using a radial basis function (RBF) kernel, polynomial kernels of different orders. We examined one vs one classification as well as one vs rest classification techniques.

C. *Random Forest Classifier*

As in the SVM approach, envelope spectrum data from the raw vibration data is tested with Random Forest Classifier from the sklearn library. We configured the classifier to use 100 estimators and the Gini impurity criterion. All nodes are set to expand until all leaves are pure or contain less than two leaves. Samples have equal weight, and the algorithm automatically the number of features to consider when looking for the best split, with no maximum number of leaf nodes. The data is split into 80 percent training data and 20 percent test data.

D. *Neural Networks*

To predict the bearing fault depth, we implemented and trained several neural networks. All networks used 1800 input nodes consisting of the envelope spectrum of the vibration data from 1 to 1800 Hz. The data were randomly shuffled before splitting the data into training and testing data sets to ensure there was no difference in fault type distribution between the training and testing data sets. We reserved 20% of the data for testing and used the remaining 80% for training. We tested networks with one and two hidden layers, where all hidden nodes used rectified linear unit (ReLU) activation functions. Because we had limited training data, we used fifty percent dropout for all hidden layer nodes to reduce the effects of overfitting.

E. *Classification using TensorFlow*

We used TensorFlow to train a classifier neural network to detect which fault type, if any, was present for each data set element. The classifier network output layer employs the softmax activation function with four outputs, representing a one-hot encoding of each of the four classes (no fault, inner race fault, outer race fault, and rolling element fault). We tried several network sizes and settled on a network with two hidden layers containing 256 and 32 nodes each. We trained the network to minimize the binary cross-entropy loss using the 'Adam' optimization algorithm on an Nvidia RTX 3070 GPU. After training the network, we tested the performance using the reserved test data set.

F. *Regression using TensorFlow*

We also used TensorFlow to train a regression neural network to estimate the bearing fault depth for each data set element. Because the goal was to estimate non-negative continuous values, we chose to use the ReLU activation function for the output layer nodes (in addition to the hidden layer nodes). The regression network output layer provides three outputs, representing the fault depth in thousandths of an inch for each of the bearing fault types (inner race fault, outer race fault, and rolling element fault). We assigned a fault depth of 0 mil for all data elements that did not exhibit that particular fault type: for example, elements with an inner race fault were assigned 0 mil depths for outer race and rolling element depths, while elements with no faults were assigned 0 mil for all three fault types. We used the same network size as the classifier network, with two hidden layers containing 256 and 32 nodes each, and trained the network to minimize the mean-square error using the 'Adam' optimization algorithm. We then tested the performance on the reserved test data set and calculated the percentage of predicted depths within each of several tolerance bands.

G. *Regression using NumPy*

Finally, we implemented a neural network in Python, primarily using the NumPy library for matrix multiplication functionality, to explore using the previously described adaptive learning rate. As with the TensorFlow regression network, this network used ReLU activation functions on all nodes and applied 50% dropout on the hidden layer; however, because we were not able to train the network using the GPU, we reduced it to a single hidden layer having 64 nodes to reduce the computation requirements.

The network is initialized with random weights, then trained on a small subset of the data for 5 epochs, at which point we record the final MSE training loss. We repeat this process 50 times, with different random weights each time, and finally select the weights which resulted in the lowest MSE loss after the training epochs. This increases the chance that the network is initialized near a solution with low error.

The training process is split into three steps: prediction, error backpropagation, and network weight update. The prediction is calculated using NumPy matrix multiplication and a ReLU function to calculate the output for each node on each layer. Each node is fully connected to the previous layer and additionally has an offset weight term. The prediction function applies 50% random dropout during training, but during testing applies a constant gain of 0.5 to all node outputs. The output error is calculated by subtracting the predicted values from the desired values. Because the gradient of a ReLU node is equal to the weight when the node is active and zero otherwise, the error is back propagated to each node in each previous hidden layer by multiplying it by the weight of any active connections and summing the effect for all nodes in the layer. Once the error term has been calculated for each node in the network, the weights are updated using gradient descent with either a constant or input-dependent, adaptive learning rate. The adaptive learning rate uses the reciprocal of the sum of the squares of the hidden node values as a maximum limit for the learning rate. While this is not the exact learning rate that minimizes the posterior prediction error, it serves as a close approximation for this network topology. As with the TensorFlow regression network, we then tested the performance on the reserved test data set and calculated the percentage of predicted depths within each of several tolerance bands.

IV. Results

A. *Linear Regression*

The linear regression method was able to achieve a root-mean-square prediction error (RMSE) of 5.675 mil for inner race, 6.907 mil for outer race, and 4.038 mil for rolling element fault types in the test data set. We calculated the percentage of predictions within ±1 mil, ±2 mil and ±3.5 mil of the true bearing fault depth, as shown in the Table 2.

B. *Support Vector Machines*

The SVMs were computationally expensive but did not perform well on the given data. The SVMs were tested with different kernels and different parameters. The SVMs with poly kernel performed well compared to RBF kernel. Poly kernel with order 1 performed better than other poly kernels with different orders. Order 1 poly kernel gave an accuracy of 66.72%, whereas order 3 poly kernel resulted in an accuracy of 64.33%. Order 5 poly kernel performed lowest out of three with an accuracy of 61.97%. The F1 score for the poly kernel with order 1 is 61.5%. Confusion matrix for this is given in Table 3. SVMs' one vs one multiclass strategy provided the classifications with 62.5% accuracy, and the confusion matrix for it is given in Table 4. One vs Rest classifier resulted in 67.27% accuracy, and the confusion matrix is given in Table 5. Finally, SVMs with the RBF kernel resulted in an accuracy of 64.5%. For the RBF kernel, the regularization parameter is set as 0.1 and gamma is set as 0.5.

Table 2: Linear Regression Fault Depth Prediction Accuracy. The percentage of training and test data where the model prediction was within the tolerance of the true fault depth.

	Inner Race	**Outer Race**	**Rolling Element**
Training data, ±1 mil	7.21%	4.79%	7.44%
Training data, ±2 mil	73.99%	11.69%	73.71%
Training data, ±3.5 mil	84.27%	69.94%	85.28%
Test data, ±1 mil	6.45%	9.68%	9.67%
Test data, ±2 mil	61.29%	19.35%	77.42%
Test data, ±3.5 mil	74.19%	80.65%	87.10%

Table 3: Confusion Matrix for SVM using a First-Order Polynomial Kernel

Predicted Actual	No Fault	Inner Race	Outer Race	Rolling Element
No-fault	5891	0	0	0
Inner Race	1074	652	0	9
Outer Race	2126	0	1064	6
Rolling Element	1054	0	7	493

C. *Random Forest*

Random Forest Classifier performed much better than the SVM. After fitting the data with Random Forest Classifier, it predicted the fault type with an accuracy of 95.27%. The F1 scores for the training set and testing set are 1.0 and 0.9526, respectively.

Table 4: Confusion matrix for a One vs One SVM

Predicted Actual	No Fault	Inner Race	Outer Race	Rolling Element
No-fault	5883	0	0	8
Inner Race	1102	609	2	22
Outer Race	2333	0	858	5
Rolling Element	1166	0	2	386

Table 5: Confusion Matrix for a One vs Rest SVM

Predicted Actual	No Fault	Inner Race	Outer Race	Rolling Element
No-fault	5755	0	0	136
Inner Race	1056	672	0	7
Outer Race	1835	0	1087	274
Rolling Element	740	0	3	811

Table 6: Random Forest Classifier Confusion Matrix

Predicted Actual	No Fault	Inner Race	Outer Race	Rolling Element
No-fault	5776	92	64	68
Inner Race	149	1510	3	0
Outer Race	85	0	30	0
Rolling Element	120	0	4	1441

D. Classification using TensorFlow

The TensorFlow classifier network with two hidden layers consisting of 256 and 32 nodes respectively was able to achieve sensitivity greater than 99% and specificity greater than 97% on the test data set for all three bearing fault types after training for 50 epochs on the full training data set, as summarized in the confusion matrix and prediction accuracy table in Tables 7 and 8 respectively.

Table 7: Confusion Matrix of TensorFlow Neural Network

Predicted Actual	No Fault	Inner Race	Outer Race	Rolling Element
No-fault	5993	6	21	87
Inner Race	96	1567	1	1
Outer Race	76	0	3013	0
Rolling Element	5	0	2	1508

Table 8: Prediction accuracy of TensorFlow Neural Network

	No Fault	Inner Race	Outer Race	Rolling Element
Training Data Classification Accuracy	96.88%	98.87%	99.24%	99.28%
Test Data Classification Accuracy	97.07%	99.16%	99.19%	99.23%

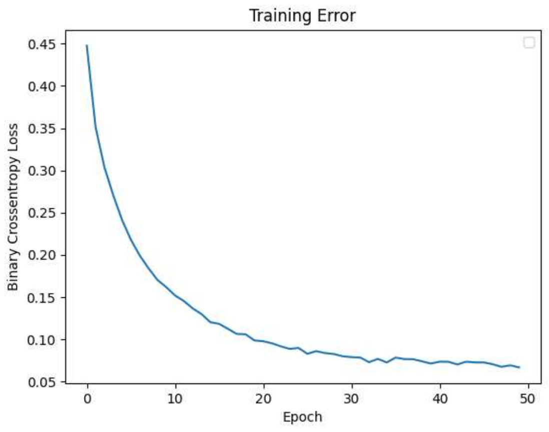

Figure 6: Binary Cross-Entropy Loss during Neural Network Training. The classification binary cross-entropy loss for the training set as the model weights evolved over the course of training the model.

E. Regression using TensorFlow

The TensorFlow regression network with two hidden layers consisting of 256 and 32 nodes respectively achieved a root-mean-square prediction error (RMSE) of 1.587 mil on the test data set (including all bearing fault types). We calculated the percentage of predictions within ±1 mil, ±2 mil and ±3.5 mil of the true bearing fault depth, as shown in Table 9.

Table 9: TensorFlow Fault Depth Prediction Accuracy. The percentage of training and test data where the model prediction was within the tolerance of true fault depth.

	Inner Race	Outer Race	Rolling Element
Training data, ±1 mil	90.94%	82.00%	88.51%
Training data, ±2 mil	93.72%	90.15%	91.41%
Training data, ±3.5 mil	95.87%	95.68%	95.11%
Test data, ±1 mil	91.16%	81.94%	88.56%
Test data, ±2 mil	94.09%	90.08%	91.23%
Test data, ±3.5 mil	96.11%	95.76%	95.01%

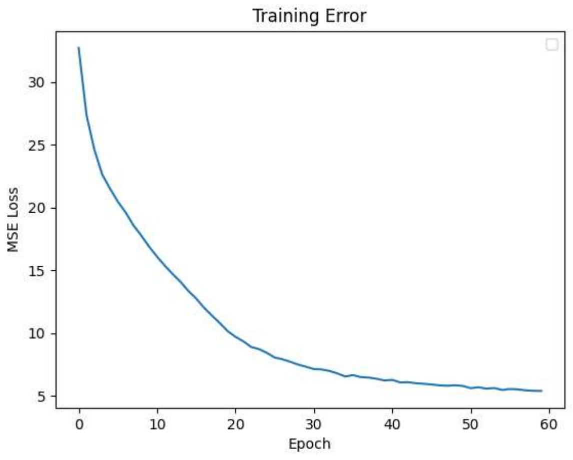

Figure 6: Mean-Square-Error Loss during Regression Network Training. The MSE for the training set as the model-weights evolved over the course of training the model.

F. Regression using NumPy

The NumPy regression network with a single hidden layer consisting of 64 nodes achieved root-mean-square prediction error (RMSE) of 5.538 mil for inner race, 6.247 mil for outer race, and 4.693 mil for rolling element fault types in the test data set. We calculated the percentage of predictions within ±1 mil, ±2 mil and ±3.5 mil of the true bearing fault depth, as shown in Table 10.

Table 10: NumPy Regression Neural Network Fault Depth Prediction Accuracy. The percentage of test data where the model prediction was within the tolerance of the true fault depth.

	Inner Race	Outer Race	Rolling Element
Test data, ±1 mil	0.48%	2.86%	2.17%
Test data, ±2 mil	1.79%	4.05%	84.46%
Test data, ±3.5 mil	72.47%	68.07%	87.59%

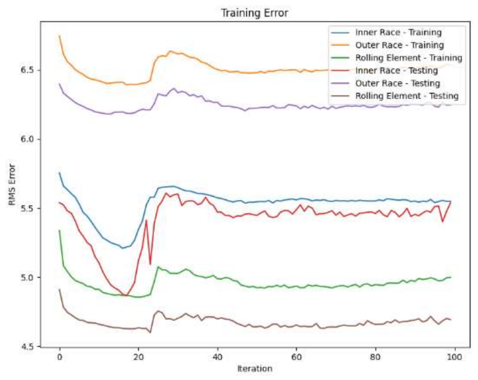

Figure 7: Root-Mean-Square-Error Loss during Network Training. The RMSE for the training and testing set as the model weights evolved over the course of training the model.

As shown in Figure 8, the training and testing error increased after training epoch 20, which may signify that the network was overfitting despite the 50% dropout and adaptive learning rate. While these results were significantly worse than the TensorFlow model, they are still competitive with the SVM classifier performance. Because it was not trained using GPU resources, this model took approximately 4 times longer to train per epoch than the TensorFlow models, despite having less than half the number of trained parameters.

V. Conclusion

We predicted bearing fault depth from rotary machine vibration data by first calculating the envelope spectrum as a pre-processing step and then applying several classification and regression approaches. We compared the performance of linear regression, support vector machines, random forests and neural networks for predicting bearing faults.

We found excellent classification and regression performance using the TensorFlow neural network models with two hidden layers, with classification accuracy exceeding 99% and more than 95% of regression predictions falling within 3.5 mil of the true bearing fault depth. The classification accuracy compares favorably with the best results found in previous published work. The regression model could additionally be used as a classifier for the bearing fault size data clusters by simply rounding to the nearest represented fault size (i.e., 0, 7, 14, 21 or 28 mils) while maintaining 94% to 96% classification accuracy on the test data set, as shown by the percentage of predictions falling within 3.5 mil of the true value. These models were very quick to train, since the implementation could take full advantage of GPU resources. Despite the model being significantly more complex and having far more parameters than any of the other methods, the similarity in performance between the training and test data seem to indicate that the model is not overfitting significantly. This is likely due in part to the use of dropout and regularization. It seems highly probable, therefore, that even better performance could be achieved with a larger network and/or more training data.

The simple linear regression model was able to perform relatively well when applied to rolling element failures, estimating the fault depth to within 3.5 mil for 87% of the test data. However, it performed significantly worse on inner race failures, with only 74% of the test data estimated within 3.5 mil. The large variability between the different failure types and between training and test data suggest that the linear regression model may not generalize well to other data, despite the resilience to overfitting typically seen in simpler models having fewer parameters.

Despite trying many different kernels and classifier strategies, we were unable to match the performance demonstrated in previous work using support vector machines. This approach tended to misclassify all three fault types as no-fault; none of the SVM models we tried achieved greater than 67% classification accuracy on the test data. It is possible that simple envelope spectrum pre-processing is insufficient for adequate performance with SVMs, since SVMs have been applied successfully to this application using other pre-processing methods. We found much better classification performance using random forest methods, which correctly classified 95% of the data in the test set.

The neural network implemented using NumPy achieved performance comparable to the linear regression model, with 87% accuracy on the rolling element failure. However, any potential benefit from the adaptive learning rate is insignificant compared to the advantages of the TensorFlow implementation, which is both much faster (due to GPU usage) and more accurate, presumably due to numerous regularization techniques. It is also not fully clear why the training error increased for the NumPy neural network after additional epochs.

To our knowledge, no previous work has attempted to directly predict bearing fault depth, so our 95% accuracy achieved using the TensorFlow neural network may serve as a benchmark for future work in this area. Additionally, we were not able to find any examples of previous work where the envelope spectrum data were used as inputs to a neural network model for classification. While it was not particularly successful in improving training efficiency or model accuracy, the adaptive learning rate in the NumPy neural network is also novel; it is possible that this method may be more successful with the addition of some of the regularization techniques employed by the TensorFlow neural network implementation.

Because the data only represented a few discrete fault diameters, future research is required to be certain that the regression model will generalize well between or beyond these values, for example preserving its level of performance on bearings with fault sizes of 5 mil or 50 mil, which were not included in our data set. There is also an opportunity to generalize this approach for different types of bearings, and to predict other types of failures in rotary machines. Applying a larger neural network similar to our TensorFlow implementation to a larger and more diverse set of training data appears to be a particularly promising avenue for future studies.

References

[1] J.K. Sinha, Health Monitoring Techniques for Rotating Machinery, Ph.D. Thesis, University of Wales Swansea, Swansea, 2002.

[2] Ameya R Mohanty, Machinery Condition Monitoring Principles and Practices, CRC Press International Standard Book Number-13: 978-1-4665-9305-3

[3] J.K. Sinha, A.W. Lees, M.I. Friswell,Estimating unbalance and misalignment of a flexible rotating machine from a single run-down,Journal of Sound and Vibration,Volume 272, Issues 3–5,2004,Pages 967-989,ISSN 0022-460X, https://doi.org/10.1016/j.jsv.2003.03.006.

[4] Saavedra PN, Ramírez DE. Vibration analysis of rotors for the identification of shaft misalignment Part 1: Theoretical analysis. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. 2004;218(9):971-985. doi:10.1243/0954406041991297

[5] Agnes Muszynska, Paul Goldman,Chaotic responses of unbalanced rotor/bearing/stator systems with looseness or rubs,Chaos, Solitons & Fractals,Volume 5, Issue 9,1995,Pages 1683-1704,ISSN 0960-0779, https://doi.org/10.1016/0960-0779(94)00171-L.

[6] D. Scott,Bearing failures diagnosis and investigation,Wear,Volume 25, Issue 2,1973,Pages 199-213,ISSN 0043-1648, https://doi.org/10.1016/0043-1648(73)90072-0.

[7] G. Scalabrini Sampaio, A. R. de A. Vallim Filho, L. Santos da Silva, en L. Augusto da Silva, "Prediction of Motor Failure Time Using An Artificial Neural Network", Sensors, vol 19, no 19, 2019.

[8] Kamat, Pooja & Marni, Pallavi & Cardoz, Lester & Irani, Arshan & Gajula, Anuj & Saha, Akash & Kumar V C, Satish & Sugandhi, Rekha. (2021). Bearing Fault Detection Using Comparative Analysis of Random Forest, ANN, and Autoencoder Methods. 10.1007/978-981-16-1089-9_14.

[9] Bearing Data Center: Case School of Engineering: Case Western Reserve University. https://engineering.case.edu/bearingdatacenter

[10] K. A. Korba en F. Arbaoui, "SVM Multi-Classification of Induction Machine ' s bearings defects using Vibratory Analysis based on Empirical Mode Decomposition", 2018.

[11] S. Zhang, S. Zhang, B. Wang and T. G. Habetler, "Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review," in IEEE Access, vol. 8, pp. 29857-29881, 2020, doi: 10.1109/ACCESS.2020.2972859.

[12] Scalabrini Sampaio, G., Vallim Filho, A. R. D. A., Santos da Silva, L., & Augusto da Silva, L. (2019). Prediction of motor failure time using an artificial neural network. Sensors, 19(19), 4342.

[13] Yu, C. C., & Liu, B. D. (2002, May). A backpropagation algorithm with adaptive learning rate and momentum coefficient. In Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290) (Vol. 2, pp. 1218-1223). IEEE.

[14] Behera, L., Kumar, S., & Patnaik, A. (2006). On adaptive learning rate that guarantees convergence in feedforward networks. IEEE transactions on neural networks, 17(5), 1116-1125.

[15] Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

[16] Luo, Z. Q. (1991). On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks. Neural Computation, 3(2), 226-245.