Deep Learning-based Vibration Analysis for Machine Health Monitoring

Submitted in Partial Fulfillment of the Requirements in
CPE313 Data Science Track Elective 3:
Advanced Machine Learning and Deep Learning

Submitted by
Jay-ann S. Alorro
Karl Josef Malaki

May 23 2025

Deep Learning-based Vibration Analysis for Machine Health Monitoring

Jay-ann S. Alorro
Computer Engineering
Technological Institute of the Philippines
Quezon City
qjsalorro@tip.edu.ph

Karl Josef Malaki
Computer Engineering
Technological Institute of the Philippines
Quezon City
qkjmalaki@tip.edu.ph

Roman M. Richard
Computer Engineering
Technological Institute of the Philippines
Quezon City
qrrichard@tip.edu.ph

ABSTRACT

This study is divided into two specific tasks that have a similar approach, to develop and build three deep learning models each that takes raw data and image spectrograms from that raw data. The objective is to compare which method is more effective in detecting machine faults through vibration data. Three Autoencoder models and another three CNN models are built and trained in order to meet the requirements. All models are analyzed and evaluated by specific metrics such as Mean Squared Error (MSE), AUC-ROC scores, and their overall accuracy. The best performing models from each task are then deployed using Streamlit in which can be accessed and used by the public. The results suggest that the base model autoencoder performed the best with the raw data being the input.

Keywords: Autoencoders, CNN, VAE, LSTM, raw vibration signals, spectrograms

---

I. Introduction

Machine vibration analysis is a technique used in analyzing oscillatory motions of machines. Their components are being inspected in order to detect and diagnose potential faults based on an established equilibrium point. Issues that might arise in these oscillations include imbalance, misalignments, looseness, bent shafts and bearing defects. This process is important for machine monitoring because vibrations are the earliest and most sensitive indicators of mechanical problems especially in rotating equipment. This prevents mechanical failures during production/usage which can delay the work flow and overall can cost more in maintenance.

As machine learning continues to progress, it has led to impressive results these past few years from areas including natural language processing and image recognition. While these examples are more present in the media, these algorithms also have great potential for industrial and mechanical applications such as the analysis of vibrations on parts of mechanical machineries. There are several studies that have performed different methodologies in analyzing vibration signals and patterns. Many methods were popularized from classical machine learning techniques such as, for example, using SVMs and Decision Trees for classification tasks, to complex algorithms such as using deep learning models like CNNs, or combinations of both models such as CNNs and LSTMs for time series analysis. Despite these advancements, there are still gaps present to the state of the art technologies.

In a journal, the researchers proposed three methods on classifying faults on an imbalanced rotating shaft. Imbalanced refers to the process in which a weight or a load is added to the rotating machine to simulate vibrations. Their results suggest that customized CNN models built with 2 fully connected layers and FFT values as inputs gained them an accuracy >95%. Moreover, they also highlighted how each of the models behave when subjected to a variety of speeds which could be exploited in further research [2]. Similarly, another journal proposed another method in which they used the Common Spatial Pattern (CSP) to identify faults in their bearings during rotational motion with an accuracy of >92%. Their findings suggest that CSP is an impactful algorithm because it overcomes the confusion created by the nonlinearity of classical methods which can be of use to fault detection not limited to bearings or rotating shafts [3]. To address these gaps the researchers will explore different models to learn the patterns on whether or not a machine is faulty based on its vibration signals and what models perform best when fed with two different types of input. The researchers trained 3 models each resulting in 6 different deep learning algorithms, the main difference being that the data being used for the 3 models are being turned into image spectrograms while the other 3 models are being trained with just the raw data itself. With this, the researchers used a public dataset that contains multivariate time-series data collected from sensors that is mounted on the SpectraQuest MFS ABVT (Alignment-Balance-Vibration Test Rig).

II. Related Work

A. Machine Health Monitoring (MHM)

Machine Health Monitoring is an important piece in maintenance in industrial systems. It aims at the longevity and reliability of mechanical equipment. It plays a major role in predictive maintenance (PdM) in which it anticipates the failures that may occur way before doing so in order to reduce downtime, maintenance costs, and potential hazards. It depends mainly on condition-based monitoring techniques that use real-time sensor data to assess the operational state of the machines [12].

Back in the day, traditional MHM depended on threshold-based systems and manual inspections. They are useful for detecting faults but they are also often reactive, imprecise, and not efficient in more complex industrial environments. More modern approaches in MHM include the use of machine learning and deep learning algorithms which are more effective for it is capable of learning complex fault patterns from large amounts of data being fed to them, enabling early and accurate prediction or fault detection [13]. Moreover, more recent studies have explored federated learning techniques for decentralized vibration monitoring, enhancing data privacy and model generalization across different industrial sites [11]. These advancements in intelligent, data-driven health monitoring increases maintenance efficiency and supports scalable and automated solutions for different industrial applications.

B. Vibration Analysis Techniques

Analyzing vibration data comes in different types and methods depending on what you want to achieve with that data. It has long been an important factor in predictive maintenance and machine monitoring conditions. Some conventional techniques may include time-domain, frequency-domain, and time-frequency domain analysis [8] [9]. Some make use of a transform-based architecture that is designed to denoise mechanical vibration signals. A model is designed to leverage multi-head attention mechanisms in order to filter out noise while preserving critical information effectively and enhancing the reliability of vibration-based diagnostics [6]. Some studies propose an autoencoder model for the condition monitoring of machine rotations. This is achieved by the model learning characteristics from the normal vibration signals [7].

C. Deep Learning in Machine Health Monitoring

Deep Learning in machine health monitoring has become more relevant in recent years due to its superior ability to extract hierarchical features from sensor data such as vibration signals. When compared to traditional machine learning techniques, they often require manual feature engineering while deep learning models can automatically learn relevant representations directly from raw or minimally processed data [14][15]. This can lessen the reliance on domain specific expertise and also improves diagnostic accuracy in more complex environments.

One main advantage of deep learning is that it can handle high-dimensional, non-linear, and noisy sensor inputs effectively. Deep learning models show great performance in identifying subtle patterns which indicate early-stage machine faults which traditional ML approaches can miss due to the limited representational capacity they have [16][17]. Some deep learning architectures that are commonly being used are: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Autoencoders (AEs), and more recently, Transformer-based models and Graph Neural Networks (GNNs).

Furthermore, end-to-end learning pipelines are increasingly adopted where deep models process raw vibration signals or lightly preprocessed inputs without requiring handcrafted features. This comprehensive approach streamlines the fault diagnosis pipeline and has been proven to be more effective in real-time scenarios industrially [11][15]. These models can operate directly on time-domain data or on data derived frequency-domain representations which offers flexibility in deployment across various sensor modalities and industrial setups.

III. Methods

This section will discuss the dataset used, the preprocessing and the method used for the raw data to be converted into image spectrograms. The deep learning models used will also be looked through, their architecture and overall approach on how the model and data worked together.

A. The Dataset

The dataset used is a public dataset composed of 1951 multivariate time-series which is acquired from SqectraQuests's Machinery Fault Simulator (MFS) Alignment-Balance-Vibration (ABVT). This dataset consists of six different simulation states which are normal, imbalance, horizontal and vertical misalignment, and inner and outer bearing faults. The sensors used are three Industrial IMI Sensors, Model 601A01 and Model 604B31 accelerometers on the radial, axial and tangential directions, IMI Sensors triaxial accelerometer, Monarch Instrument MT-190 analog tachometer, Shure SM81 microphone with a frequency range of 20-20.000 Hz, and two National Instruments NI 9234 4 channel analog acquisition modules with a sample rate of 51.2 kHz. The sequences are generated at a 50 kHz sampling rate during 5 seconds, which gives us a total of 250.000 samples [10].

Before building the models and actually training it, a comprehensive exploratory data analysis (EDA) was performed to understand the structure, the distribution, and the underlying patterns of the raw vibration data. The first step in the analysis was to evaluate the class distribution across the samples. The class imbalance plays an important role in the model's ability to generalize across faulty data types; you can see the distribution of the data in Table 2. From the distribution, the dataset appears to be an imbalance which has implications for the training process of the model. This imbalance can lead to problems such as biased learning, which would need to involve techniques like oversampling, undersampling, or weighted loss function.

B. Vibration-based Fault Classification using Autoencoders on Raw Data

This section will discuss the models used in classifying machine faults with the use of raw data. 3 autoencoder models are used, a base Autoencoder model, Variational Autoencoder (VAE) and a Long Short-Term Memory (LSTM) Autoencoder model. The main objective is to determine the best performing model with the use of the raw vibration data that is being used. The whole process is illustrated in Fig. 1 below.

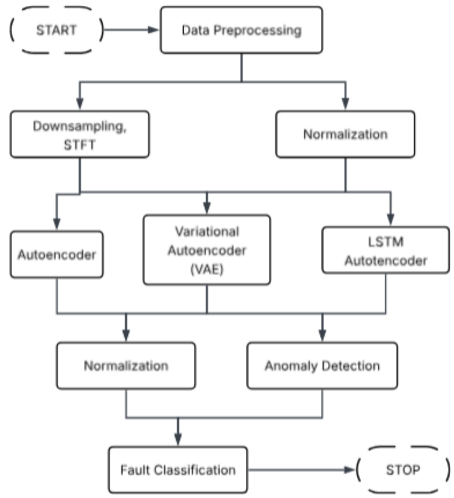

Figure 1. Flowchart of the whole task 1

a. Data Description and Processing

The dataset used in this study are time-series vibration signals which encompasses both normal operating states and another five conditions that have several fault conditions including imbalance, horizontal misalignment, vertical misalignment, overhang with outer race fault, and underhand ball fault.

For the base model autoencoder, the csv files were processed by downsampling with a factor of 5 in order to reduce the dimensionality of the data. The frequency-domain and time-domain features were then extracted and concatenated in order to form a comprehensive feature vector for each sample. These were then normalized using the Min-Max scaling technique to ensure that all features have equal contribution to the model training phase. The data process that took place for the Variational Autoencoder (VAE) is a bit different for the data were downsampled by a factor of 50 in order manage computational complexity and each channel of the data was normalized using the StandardScaler for the feature distributions. Lastly, for the LSTM Autoencoder, the raw sequences were normalized using Min-Max scaling and given the different lengths of the sequences, the data were padded to a fixed length of 1024 time steps which ensures uniformity in input dimensions for the model.

b. Autoencoder Models

The first model that is used is a Vanilla Autoencoder which consists of a feedforward neural network with an encoder-decoder structure. The network also includes Batch Normalization, Leaky ReLU for the activation function, and some Dropout layers in order to enhance training stability and to prevent overfitting. This model is trained using the Mean Squared Error (MSE) loss function and Adam for the optimizer.

The second model is a Variational Autoencoder (VAE) which introduces a probabilistic approach with its architecture. The encoder maps the input data to a latent space which is characterized by a mean and variance and this facilitates the generation of new data samples. The decoder on the other hand reconstructs the input data from samples from this latent space distribution. There are two components that comprise the loss function, the reconstruction loss which is measured by binary cross-entropy and the Kullback-Leible (KL) divergence which regularizes the latent space. A beta coefficient was also applied to balance the two components during training.

Lastly, the third model is a LSTM Autoencoder that is designed to capture temporal dependencies in the sequential data. This model employs Long Short-Term Memory (LSTM) layers in both the encoder and decoder. Table I summarizes the autoencoder models that are used as well as their general performance after training. All of these three models were trained exclusively on data representing simulated operating conditions to learn the standard behavior of the machinery. The summary of all the models and their overall performance is in Table 1.

Table 1. Model input shape and accuracy results

Model Summary			
Models	*Input Shape*	*Architecture*	*Top-1 Acc (%)*
Autoencoder	(1024, 1)	3 FC layers	84%
VAE	(1024, 1)	FC + Sampling	77.8%
LSTM	(1024, 1)	Encoder - Decoder LSTM	-

c. Model Evaluation

After building and training the three models using the raw vibration data, their performance was assessed using various metrics such as confusion matrices, recall, F1-score, and overall accuracy all based on the Mean Squared Error (MSE). The goal of the models was to detect anomalous patterns with high sensitivity and reliability.

C. Vibration-based Fault Classification using CNNs on Spectrogram images.

This section will discuss the second task of the project in which the main objective is to test which of the three CNN based models can provide the best outputs in performing binary classification tasks on vibration spectral images. The whole process from start to finish is illustrated in Fig.2.

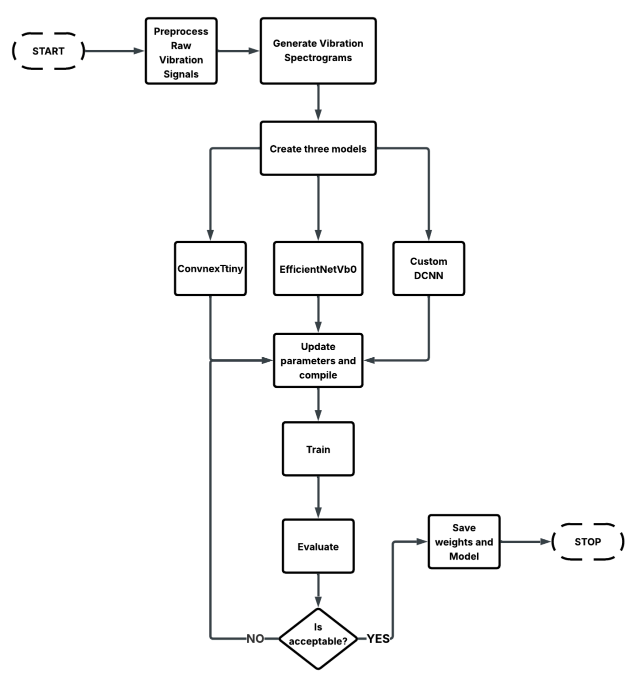

Fig.2. Flowchart of the whole component

a. Data Description and Processing

Inside the MAFAULDA database, there are six classes involved which are summarized in Table 2.

Table 2. Class distribution and their samples

Summary of Classes		
Classes	*Label*	*Samples*
Normal	1	49
Vertical-misalignment	0	301
Horizontal-misalignment	0	197
Imbalance	0	333
Underhang	0	186
Overhang	0	137

Among these classes, only the overhang class was used in this task. Moreover, Inside these folders are subfolders with CSV files that contain raw vibration signals. For example, the Overhang folder has subfolders that represent the imbalance or "weights" applied to the bearing during rotational motion. "0G" folder contains vibration csv files without any load applied to the bearing during rotational motion, hence this subfolder is used to represent the "healthy" class. Similarly "35G" is another folder that contains vibration csv files with an applied load of 35g, this will serve as the "faulty" class for this purpose. In total, there are eight columns with different specifications and measurements.Columns one and eight represent the tachometer and the microphone. The tachometer estimates the rotational frequency of the bearing while the microphone records the acoustic signals produced by the bearing during rotational motion. Columns two to four represent the underhang bearing and overhang bearing for columns five to seven.

The initial step would be to extract the underhang columns and transform them into a series format using the Pandas package in Python. The underhang columns are transformed into a mel-spectrogram converter function that will take a frequency sampling rate of 15KHz. The number of samples used in Fourier Transform is 1024 samples with a hop length of 512. Figure 3 shows a sample image of a healthy and faulty spectrogram with the applied aforementioned specifications.

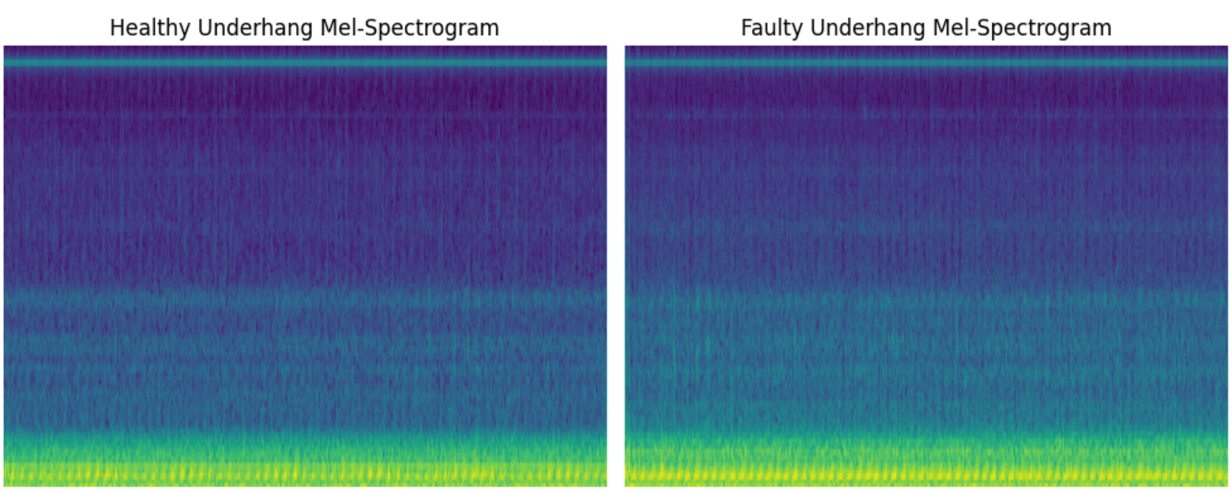

Figure 3. Healthy and faulty spectrogram sample.

Consequently, during the conversion process, the raw vibration signals are normalized using Z-score normalization shown in Eq. 1

$$\text{Eq.(1)} \quad X = (S - \bar{x})\frac{1}{\sigma(S)}$$

Where X represents the final signal, variable S represents the signal to be subtracted to the mean of the signal and multiplied to the inverse of its standard deviation. This will return values that are centered around zero which helps specific algorithms to scale with the input data better.

After the normalization process, the dataset is fed into another function that will reorganize the images into their corresponding train, val, and test splits. The main folder in which the images are stored will be splitted into 70 images alloted to train, 20 images allocated to validation, and the last 10 will be for final validation or predictions. The train data generator is also equipped with light data augmentation techniques such as rotation range with a value of 0.2, a zoom effect with an intensity of 0.005, and a brightness range of 0.5-0.7.

b. Convolutional Neural Network

The first CNN-based model that will be used is the ConvNeXt-tiny model, the model is composed of convolutional layers that's followed by a fully connected layer. This model is known for having the best performance especially in classifying images. The second model will be the EfficientNetV2B0 which is and is almost similar to the original EfficientNet variants. This model is also known to perform well in classifying images. The last model would be a custom DCNN model that was designed similarly with the VGG models. Table 3 summarizes the inputs and parameters of the model.

Table 3. Model summary and benchmarks

Model Description				
CNN Models	*Input Shape*	*Architecture*	*Trainable Parameters*	*Top-1 Acc (%)*
ConvNeXt-tiny	(224,224,3)	3 FC layers	27,926,881	82.1%
EfficientNetV2B0	(224,224,3)	2 FC layers	6,280,273	78.7%
Deep-CNN	(224,224,3)	3 Conv2D + Max Pooling	11,169,089	80.3%

c. Model Evaluation

The three models are evaluated using the AUC-ROC curve metric that is imported from the sklearn.metric package. This computes the overall ability of the model to predict the class of an image. Validation accuracy, and loss will also be illustrated to show how each of the models behaves after each epoch of training. Finally, the model with the best performance given the following metrics will be tested against the best model that was picked in task 1.

D. Model Deployment

For model deployment, an open-source Python library is going to be used, Streamlit. A python script is made that takes 2 inputs, either a csv file or an image spectrogram file which will be then passed to the best performing models that can predict whether or not the file uploaded is healthy or faulty in terms of machine health. This method of deployment makes certain that users, maintenance engineers, or even other researchers can easily assess machine health without the need to run the models manually, especially those who do not have a background in programming. This tool can be accessed by a public Streamlit link and a user-friendly interface will be presented, enabling real-time and remote anomaly detection with minimal technical overhead.

IV. Results and Discussion

This section presents the model results and discussion from the three tested models from the two tasks. Similarly, the best models from both tasks will also be compared and presented in this section.

A. Task 1 (Vibration-based Fault Classification using Autoencoders on Raw Data)

The first model, the base model Autoencoder, is the best performing model out of the 3 with an accuracy of 90% which demonstrates a strong ability to detect anomalies. It has a perfect recall which means that it identified all failed or faulty instances. The recall is only 70% which means that it misclassified some normal data. Overall, this model prioritizes safety by minimizing the missed anomalies. Figure 4 shows the distribution of error of the model.

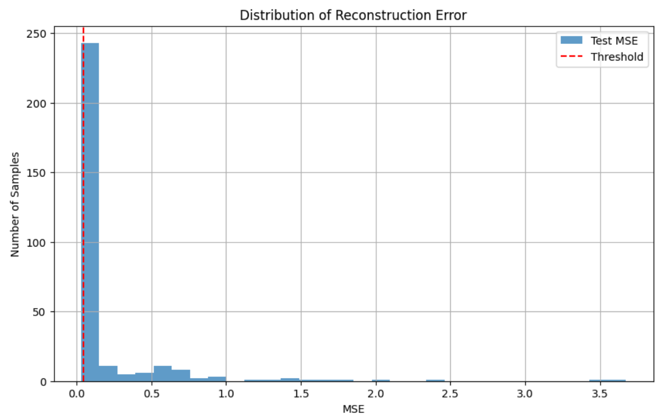

Figure 4. Distribution of Error - Vanilla Autoencoder

The Vanilla Autoencoder has the best performance out of the three Autoencoder models and it can be seen more clearly with the ROC curve in Figure 5. The AUC value of 0.9657 suggests that the model has a strong ability to differentiate between the positive and negative classes. The graph shows that the model is an effective fault classifier, it not only maintains a strong balance between sensitivity and specificity but also outperforms random guessing. While the performance is already good, it is necessary to work on the model because it definitely still has room for improvement.

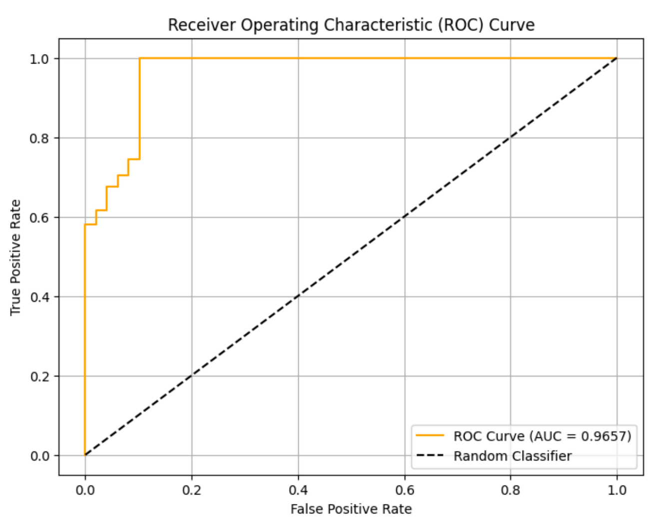

Figure 5. ROC AUC Curve of Vanilla Autoencoder

The Variational Autoencoder (VAE) model has a lower accuracy score of only 72% with a noticeable exchange between its precision and recall across the two classes. In terms of identifying normal classes, it performed particularly well with 0.90 but it struggled to accurately classify anomalies which had a higher false-negative rate. This suggests that the model might underfit the anomalous patterns or that it learned a less discriminative latent space. Figure 6 shows the distribution of the reconstruction error of the VAE model.

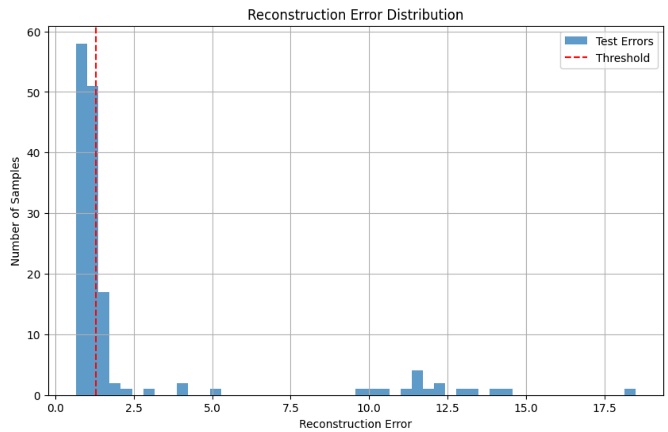

Figure 6. Distribution of Error - VAE

Lastly, the LSTM Autoencoder almost performed just as well as the base model with an accuracy score og 88%. It was good at detecting anomalies with the recall being 1.00 while still having a precision of 0.83 for the anomalous class. Just like the base model, it also misclassified some normal samples as faults. Still, this model might be particularly useful in capturing temporal dependencies in the vibration data which leads to a more robust anomaly detection. Figure 7 shows the distribution of reconstruction error for the LSTM model.

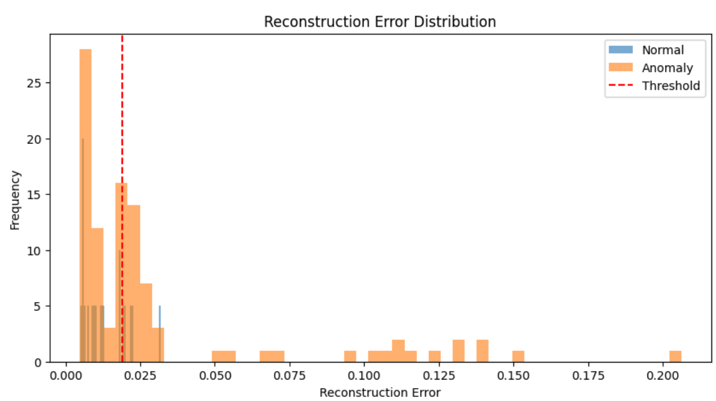

Figure 7. Distribution of Error - LSTM

Upon further improvement with the three models, Figure 8. Shows the overall ROC Curve of the models and how they compare with each other. The green curve represents the Vanilla Autoencoder with the highest AUC score of 0.9727 which shows the best overall classification performance among the three, it has the highest true positive rates for the lowest false positive rates across most thresholds. The blue curve represents the LSTM Autoencoder which has an AUC score of 0.9456, it also performs well but not quite reaching the level of the Vanilla Autoencoder. The red curve represents the Variational Autoencoder (VAE) which has the lowest AUC score of 0.9403 which is close to the performance of the LSTM model but it still slightly trails the other two models based on the graph.

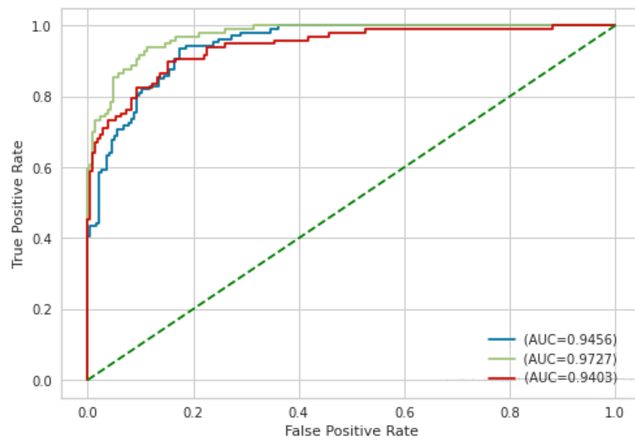

Figure 8. ROC AUC Curve Summary

B. Task 2 (Vibration-based Fault Classification using CNNs on Spectrogram images)

The ConvNeXt tiny model was trained initially with the “weights” parameter set to None. The goal here is to overfit the base model with the “faulty” dataset so that it can learn the patterns. After this step, another instance of the model is created with the same architecture of the base model. This is trained for 10 epochs. Initial training is illustrated in Fig 9.

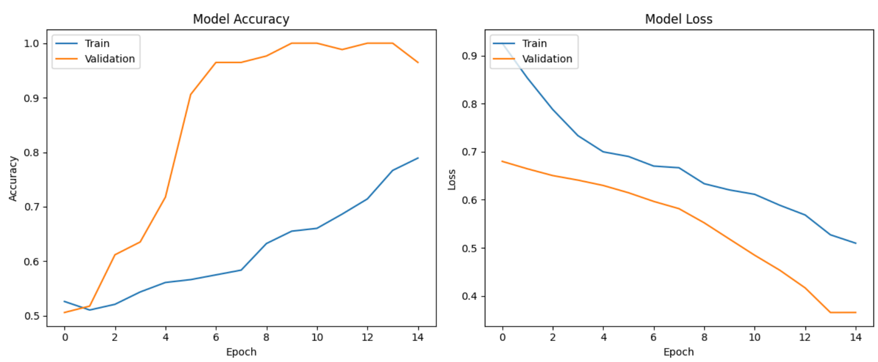

Figure 9. Initial Training of the ConvNeXt tiny model

The image above shows the line plot of the initial training of the ConvNeXt tiny model. The right graph shows the training and the validation accuracy of the model per epoch and the left shows its loss per epoch. On the right graph, the training of the model shows clear signs of poor generalization and overfitting. This is because of the huge gap found between the train and the validation accuracy. This is further emphasized by the higher loss of train than the loss in validation.

Similarly, the EfficientNet model also experienced a poor generalization and signs of overfitting as seen on Fig 10.

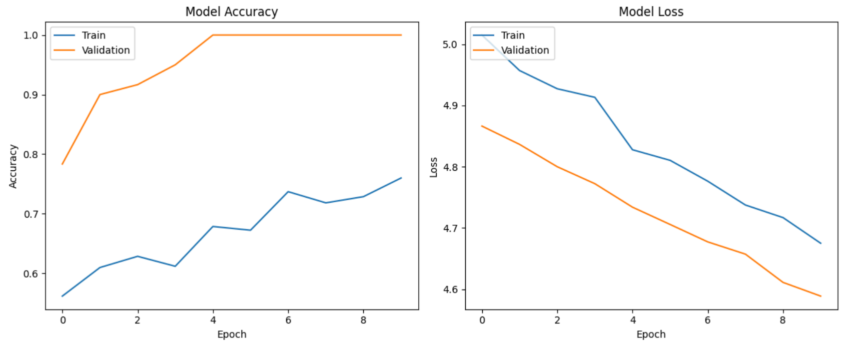

Figure 10. Initial Training of the EfficientNet2B0 model

Unlike the ConvNeXt tiny model, EfficientNetv2B0 showed fluctuating behavior in training indicating that the model is only random guessing and not efficiently learning.

Lastly, the custom DCNN model which is a model with customized interconnected layers. The model had undergone the same process of ConvNeXt tiny model where transfer learning was utilized on a base model which was overfitted using the "faulty" dataset.

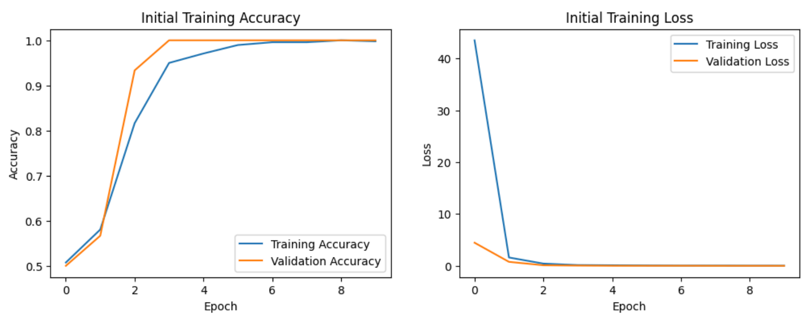

Figure 11. Initial Training of the DCNN model

As interpreted from the image, the graph on the left shows a similar trait to the previous models however the gap between the training and validation is not significantly big compared to the other models. Moreover, the validation accuracy was able to converge faster than training which could indicate small amounts of overfitting. Fig.12 shows the final predictions using the DCNN model.

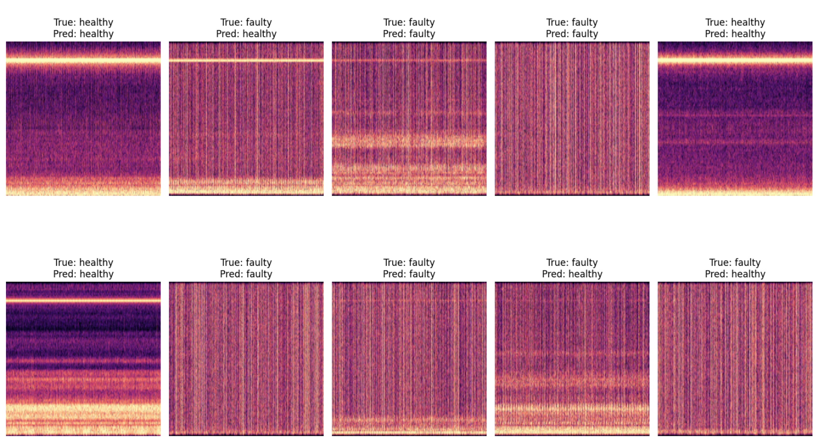

Figure 12. Final predictions using the DCNN model

The image above shows the predictions made on the test dataset. The image shows the true class and predicted class made by the DCNN model. Based on the image the model was able to predict 8 classes correctly.

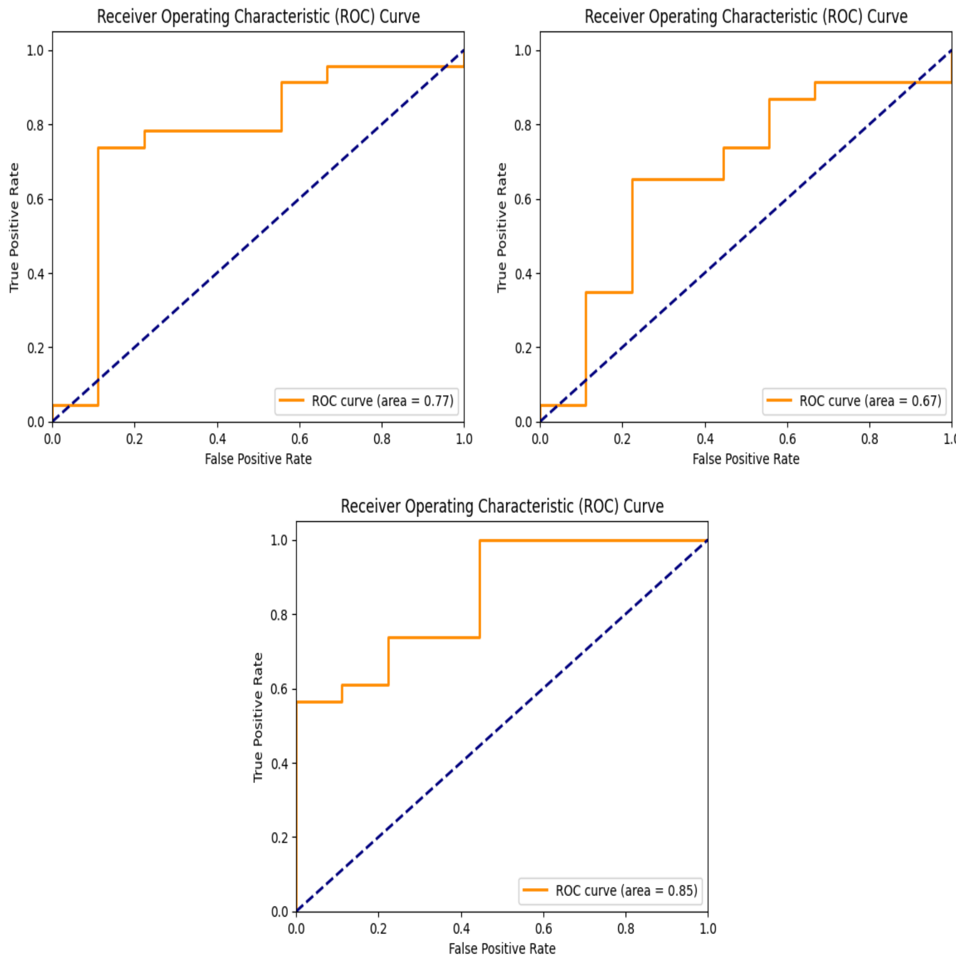

Figure 13. Final ROC graphs for the three models

Fig 13. shows the AUC curve of the three models side by side. As seen, the best performing model is the custom CNN in which compared to other three models, the curve was able to cover the majority of the area below the line. Similarly, both ConvNeXT and EfficientNet models which were pre-trained on imagenet, were not able to fully capture the area below the line posing a predictability issue with the model.

V. Conclusion and Recommendations

In conclusion, the researchers have found out that the best performing model is the base Autoencoder with the raw data being used. This suggests that you don't always need complicated models in order to have good results, sometimes the more simple, the better. It all boils down to the objective that you are trying to achieve.

The researchers recommend focusing more on the preprocessing of the raw data when turning them into image spectrograms in order to minimize the noise because it is still present even on the normal vibration data. Upon processing the data, the researchers also observed that the normal data available is limited compared to the faulty vibration data provided. The approach taken in this paper is to upsample the data by applying interpolation and duplicating existing data. It is better to have used a GAN model in order to generate synthetic data that can be used to train the model to have the model learn more of the patterns on the normal data.

VI. Acknowledgements

The authors would like to acknowledge the support of their instructor, Engr. Roman Richard for sharing his knowledge, expertise, and providing relevant guidance that helped them refine the approach they should take in this project.

VII. References

[1] “What is Vibration Analysis? | IBM,” Ibm.com, Aug. 21, 2024. https://www.ibm.com/think/topics/vibration-analysis
[2] “Machine Learning-Based unbalance detection of a rotating shaft using vibration data,” IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9212000
[3] J. Liu and X. Yang, “Learning to see the vibration: a neural network for vibration frequency prediction,” Sensors, vol. 18, no. 8, p. 2530, Aug. 2018, doi: 10.3390/s18082530.
[4] P. Gupta and M. K. Pradhan, “Fault detection analysis in rolling element bearing: A review,” Materials Today Proceedings, vol. 4, no. 2, pp. 2085–2094, Jan. 2017, doi: 10.1016/j.matpr.2017.02.054.
[5] S. Misra et al., “Fault detection in induction motor using time domain and spectral Imaging-Based transfer learning approach on vibration data,” Sensors, vol. 22, no. 21, p. 8210, Oct. 2022, doi: 10.3390/s22218210.
[6] H. Chen, Y. Yu, and P. Li, “Transformer-Based denoising of mechanical vibration signals,” arXiv.org, Aug. 04, 2023. https://arxiv.org/abs/2308.02166
[7] S. Ahmad, K. Styp-Rekowski, S. Nedelkoski, and O. Kao, “Autoencoder-based condition monitoring and anomaly detection method for rotating machines,” arXiv.org, Jan. 27, 2021. https://arxiv.org/abs/2101.11539
[8] “Mechanical Condition Assessment of Vacuum On-load Tap-changer for Converter Transformer based on time-frequency domain characteristics of Vibration Signal,” IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9279768
[9] “Anti-noise motor fault diagnosis method based on decision tree and the feature extraction methods in the time domain and frequency domain,” IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9446018
[10] “MAFAULDA :: Machinery Fault Database [Online].” https://www02.smt.ufrj.br/~offshore/mfs/page_01.html
[11] “Federated learning for autoencoder-based condition monitoring in the industrial internet of things,” IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/10020836
[12] L. Polverino et al., “Machine learning for prognostics and health management of industrial mechanical systems and equipment: A systematic literature review,” International Journal of Engineering Business Management, vol. 15, Feb. 2023, doi: 10.1177/18479790231186848.
[13] B. A. Tama, M. Vania, S. Lee, and S. Lim, “Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals,” Artificial Intelligence Review, vol. 56, no. 5, pp. 4667–4709, Oct. 2022, doi: 10.1007/s10462-022-10293-3.
[14] S. Ahmad, K. Styp-Rekowski, S. Nedelkoski, and O. Kao, “Autoencoder-based condition monitoring and anomaly detection method for rotating machines,” arXiv.org, Jan. 27, 2021. https://arxiv.org/abs/2101.11539
[15] Y. Jeon, K. Kim, Y. Lee, B. K. Kang, and S. W. Lee, “Development of Fault Diagnosis Model based on Semi-supervised Autoencoder,” PHM Society European Conference, vol. 8, no. 1, p. 7, Jun. 2024, doi: 10.36001/phme.2024.v8i1.4023.
[16] M. T. Singh, “Ensemble-Enhanced Graph Autoencoder with GAT and Transformer-Based Encoders for Robust Fault Diagnosis,” arXiv.org, Apr. 13, 2025. https://arxiv.org/abs/2504.09427
[17] Y. Lee and K. Park, “An experiment on anomaly detection for fault vibration signals using AutoEncoder-Based N-Segmentation algorithm,” PHM Society European Conference, vol. 8, no. 1, p. 8, Jun. 2024, doi: 10.36001/phme.2024.v8i1.4070.

Evaluation Metrics

Student Outcome 7							
Criteria	**Ratings**						**Pts**
T.I.P. SO 7.1 Acquire and apply new knowledge from outside sources threshold: 4.2 pts	**6 pts** **[Excellent] Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently and applies knowledge learned into practice**	**5 pts** **[Good] Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently**	**4 pts** **[Satisfactory] Look beyond classroom requirements, showing interest in pursuing knowledge independently**	**3 pts** **[Unsatisfactory] Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently**	**2 pts** **[Poor] Relies on classroom instruction only**	**1 pts** **[Very Poor] No initiative or interest in acquiring new knowledge**	6 pts
T.I.P. SO 7.2 Learn independently threshold: 4.2 pts	**6 pts** **[Excellent] Completes an assigned task independently and practices continuous improvement**	**5 pts** **[Good] Completes an assigned task without supervision or guidance**	**4 pts** **[Satisfactory] Requires minimal guidance to complete an assigned task**	**3 pts** **[Unsatisfactory] Requires detailed or step-by-step instructions to complete a task**	**2 pts** **[Poor] Shows little interest to complete a task independently**	**1 pts** **[Very Poor] No interest to complete a task independently**	6 pts
T.I.P. SO 7.3 Critical thinking in the broadest context of technological change threshold: 4.2 pts	**6 pts** **[Excellent] Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions**	**5 pts** **[Good] Evaluate information from a variety of sources; formulates a clear and precise perspective.**	**4 pts** **[Satisfactory] Analyze information from a variety of sources; formulates a clear and precise perspective.**	**3 pts** **[Unsatisfactory] Apply the gathered information to formulate the problem**	**2 pts** **[Poor] Gather and summarized the information from a variety of sources but failed to formulate the problem**	**1 pts** **[Very Poor] Gather information from a variety of sources**	6 pts
T.I.P. SO 7.4 Creativity and adaptability to new and emerging technologies threshold: 4.2 pts	**6 pts** **[Excellent] Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.**	**5 pts** **[Good] Ideas are creative and adapt the new knowledge to solve a problem or address an issue**	**4 pts** **[Satisfactory] Ideas are creative in solving a problem, or address an issue**	**3 pts** **[Unsatisfactory] Shows some creative ways to solve the problem**	**2 pts** **[Poor] Shows initiative and attempt to develop creative ideas to solve the problem**	**1 pts** **[Very Poor] Ideas are copied or restated from the sources consulted**	6 pts
							Total Points: 24

Evaluated by:

Engr. Roman M. Richard
Course Instructor