A Study on Vibration Detection and Rejection: Enhancing IMU Data Accuracy

B. Özbay, D. Erdoğan

***Abstract*—Inertial Measurement Unit (IMU) is used in many application areas. Aircraft, ground robots and submarines are at the forefront of these application areas. Vibration analysis is very important to ensure and maintain correct behavior for these vehicles. This analysis can be done with signal processing methods or artificial intelligence.**

***Index Terms*—Vibration analysis, Inertial measurement unit (IMU), Fast Fourier Transform (FFT), K-Means Clustering, Denoising Autoencoder**

I. Introduction

Measurements made using IMU are very important for the stability of the system. For example, when it is determined that the sensor data is a vibration signal, the necessary maintenance and repair should be done. Vibration signals can be caused by engines or environmental factors. Accelerometer and gyroscope data can be negatively impacted by vibration of the vehicle, which can in turn degrade the vehicle's ability to navigate accurately. Therefore, systems that tolerate vibration are being developed. In this study, a working environment was created using MATLAB imuSensor, and datasets containing vibration signals were observed. Signal processing techniques are utilized to create the artificial vibration signal and machine learning models were used to detect the vibration [1]. With this research, it was aimed to have more data by multiplying the vibration data and thus to train the machine learning model for more situations. Detection of artificial or real vibration signals and damping of vibration are also carried out with machine learning techniques. Thus, it can contribute to the design of more reliable navigation systems.

II. Literature Review

In a study conducted using IMU sensor, which includes a three-axis accelerometer, gyroscope and magnetometer, a Gaussian vibration profile was used and applied to the external surface of the device in the experiment. As a result of this study, it was determined that although the vibration was applied only on one axis, the IMU sensors also showed sensitivity on other axes, which is shown in Fig. 1 . This shows that the devices can produce signals on unwanted axes. Therefore, the results emphasize that the data processing algorithms of the IMU sensors should be optimized to filter or compensate for the effect of vibrations [2].

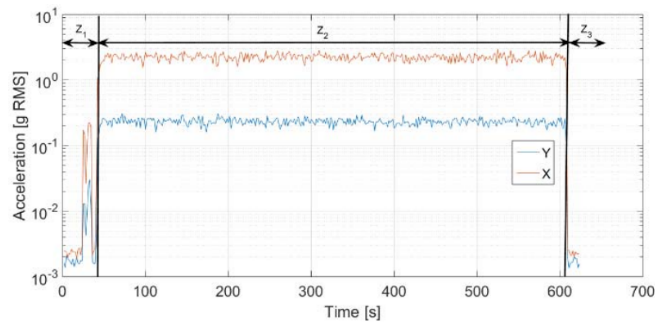

Fig. 1. Accelerometer sensor output (the vibration is applied on the x-axis).

In the study that developed an adaptive vibration filter using Micro Electro - Mechanical System (MEMS) based on low cost IMU sensor, Adaptive Noise Canceller (ANC) was designed to prevent errors caused by vibration. In ANC, a noise estimate is made using various coefficients based on the reference signal associated with the noise. This noise is removed from the raw sensor data and an error is found and a coefficient calculation is made based on this error. As a result, as stated in Fig. 2 and Fig. 3, it was observed that the data obtained from ANC and Kalman filter were successful [3].

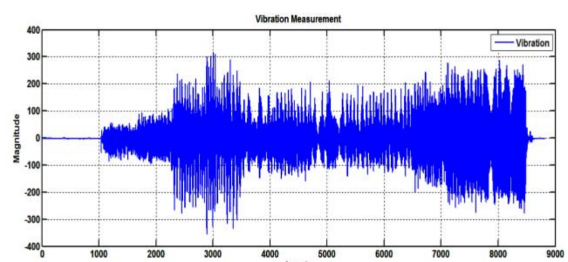

Fig. 2. Vibration measurement.

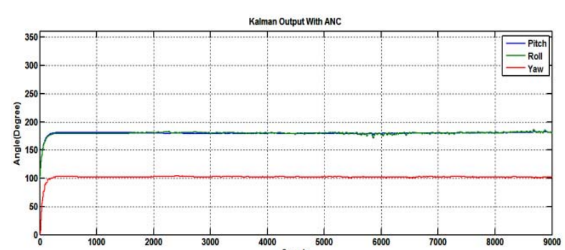

Fig. 3. Kalman filter output with ANC.

III. Dataset Information

Used dataset in this study is collected with an accelerometer in a research, using vibration data to predict motor failure time, where artificial neural networks were used. While producing the time domain features as used in dataset, the speed of fan is increased from 20 percent to 100 percent with the increment of 5 percent. So in total 17 times speed is changed and 3000 data is collected per speed. To cover different vibration levels, 3 weight configurations are used. For 3 of them, collection of 3000 data for different speeds are made, resulting in data with the length of 153000. To understand the behavior of IMU under vibration properly, the time domain and especially the frequency domain representations of the signal is analyzed. As shown in Fig. 4. During both the first and second peaks, there are many fast up-down oscillations around the acceleration values, which can be described as noise due to the increase in speed. In particular, the experimental conditions such as sensor sensitivity and the effect of mechanical weights can increase the intensity of this noise. Frequency domain representation is shown in Fig. 5. This representation illustrates that in the x-axis, largely dominant at frequencies near zero (static or very low frequency vibration), and no significant peaks or resonances are seen at other frequencies since x-axis is the axis of height. Since there is no valuable vibration information, x-axis does not take into consideration. For the y-axis, a pronounced peak around 15 Hz indicates a mechanical resonance or vibration source on that axis. This may mean that the system is vibrating more on this axis around 15 Hz. For the z-axis, there is a noticeable small spike at medium frequencies (10–15 Hz). From this inference, y-axis is more similar to a vibration signal by its frequency domain analysis. [4].

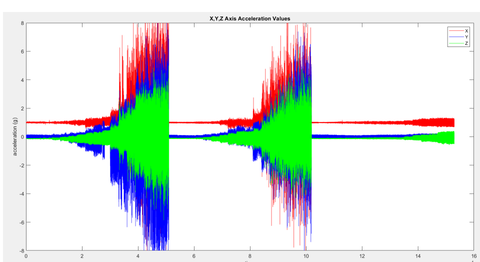

Fig. 4. 3-axis acceleration data in time domain.

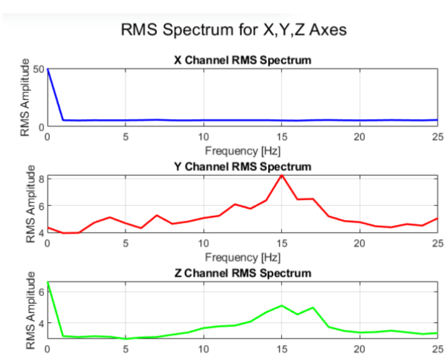

Fig. 5. 3-axis acceleration data after short time fourier transformation (STFT).

IV. Problem Description

The objective of the project is to simulate vibrating IMUs subject to build systems that can tolerate vibrations. The problem consists of two main parts.

In the first part, the aim is to create a model that generates signals that mimic IMU data under vibration. The model is developed by signal processing techniques. It takes the dataset as training data and analyzes the input signal from the vibrating IMU's output. The model is expected to generate signals that align with vibrating IMU's output as similar as possible to the vibration dataset as in Fig. 6.

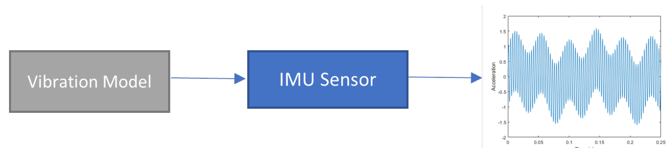

Fig. 6. IMU + vibration model

The second part of the problem is compensating for the signals generated in the first part. First, whether there is vibration or not is identified. Then, the effect of vibration-induced signals is reduced or completely extracted as in Fig. 7. Thus, the accuracy of sensor data is increased.

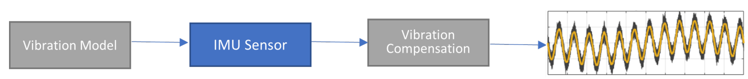

Fig. 7. Vibration compensation

V. Solution Strategy

The solution strategy aims to solve two tasks: generating vibration signals similar to the dataset and compensating these vibration signals in the IMU output.

A. Data Preparation

The dataset mentioned in [4] involves the acceleration data on three axes that read from a vibrating IMU. This data was collected as 3060 windows and 50 measurements per window. The measurements for each axis are read from the dataset one by one and separated to align with the number of windows and samples per window. Then they are normalized to reduce variations in the scale of the data. The data in only 1 window (50 samples) after normalization is illustrated in Fig. 8.

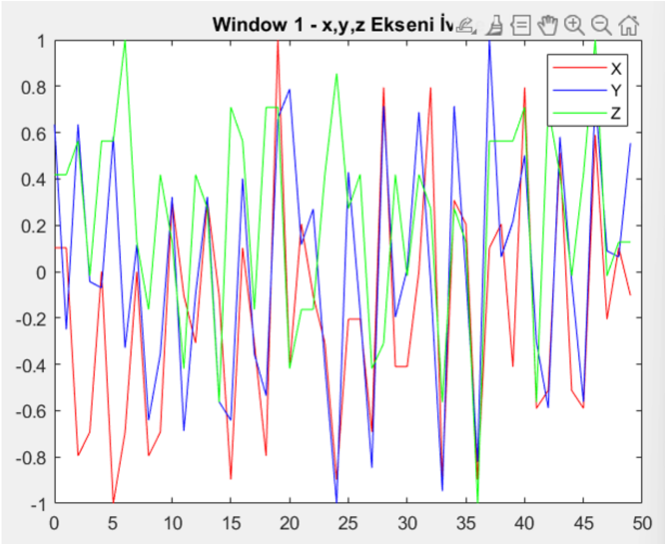

Fig. 8. 1 window (50 samples) acceleration data after normalization

B. Signal Generation Using Signal Proccessing Techniques

To develop the vibration model classical signal processing techniques are used. The aim is to develop a vibration model that learns from the dataset and mimics the real vibration data. The signal is created by taking a baseline created with random noise and then adding a vibration similar to the vibration data in the used dataset.

C. K-Means Clustering

In this part the aim is to classify the data as vibrating and non-vibrating. An unsupervised technique such as K-means can be used for this. Vibrating and non-vibrating signals have different characteristics in frequency domain. Thus, data can be classified with K-means by their frequency characteristics.

D. Vibration Compensation Algorithm

In this part the aim is to detect and compensate for the vibration signals. To detect whether there is vibration or not, the Fourier transform is taken of the signals generated in the previous step and vibration frequencies are extracted. The properties acquired from frequency analysis can be given as input to a ML model (e.g. power spectrum density, dominant frequency, energy ratio in frequency bands). Thus, vibrating and non-vibrating signals are classified using K-means clustering as an unsupervised machine learning model. Then, the classified vibration signals are subjected to compensation by denoising autoencoder, a deep learning technique. The output of the DL model make the signal noise and vibration free.

VI. Vibration Signal Generation

The purpose of generation a synthetic acceleration signal is to pretend the real world IMU behavior to contribute the variety of the samples and to lower the cost since the setup of collection of data may be costly. Firstly, envelopes are produces with the formula of $1-\cos(x)$ to obtain a bell-shaped data like in the dataset used. This scale, provides a realistic and natural change in the data. Especially, the intervals of high vibration occurs such as 40,000 to 60,000 seconds and 100,000 to 120,000 seconds are defined. These intervals simulate the vibration bursts may stems from external mechanical events or the motor operation. If there is no external effects to IMU, the data may be affected from self noise of the sensor. To simulate this baseline is created. This baseline merges a constant offset with Gaussian noise, which represents the small, random fluctuations of IMUs with imperfections like in the real-world. Gaussian noise models the inherent imperfections and variability in sensor measurements, such as thermal noise or electrical interference. The envelope and the baseline are combined to generate a whole IMU acceleration data. The result is a synthetic signal that mimics the output of an real IMU under vibration closely, with the transitions between periods of vibration and rest. Finally, to evaluate the generated signal characteristics in terms of vibration precisely, frequency domain analysis is needed.Generated signal is shown in Fig. 9 and Fig. 10 with their frequency domain representation. As illustrated, the frequency is not constant with high values, meaning that the object is subjected to the vibration.

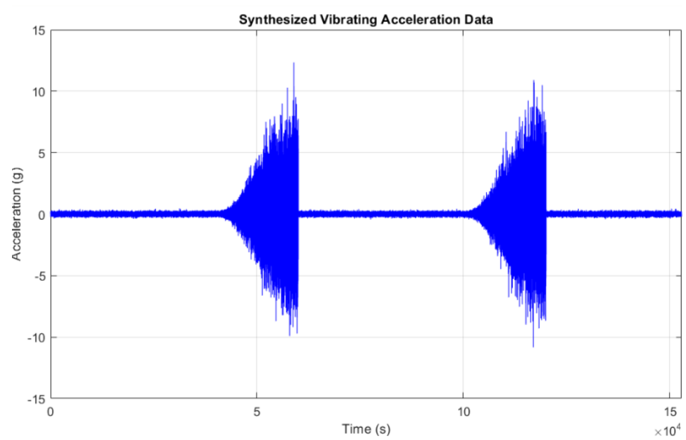

Fig. 9. Synthesized vibrating acceleration data

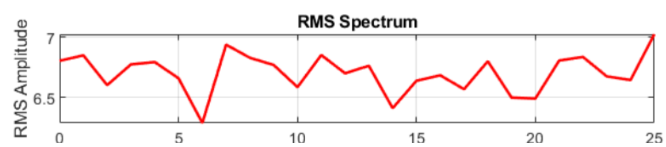

Fig. 10. Synthesized vibrating acceleration data in frequency domain

VII. Vibration Data Classification

K-means clustering is an unsupervised ML technique that divides N data objects into k clusters to minimize the L_2 Loss. The aim is to maximize the intra-cluster similarity while minimizing similarity between clusters. First, k objects representing the center of each cluster are randomly selected. The remaining objects are included in the closest cluster according to their distance from these cluster centers. According to the clustering, a new center is determined and these steps are repeated until it becomes stable by examining the distance of the objects again [6].

$$E = \sum_{i=1}^{k} \sum_{x \epsilon S_i} ||x - \mu_i||^2 \tag{1}$$

S_i : i-th cluster
μ_i: Average of points in set S_i

$$\mu_i = \frac{1}{|S_i|} \sum_{x \epsilon S_i} x \tag{2}$$

This part of the project aims to classify the generated data into two groups: vibrating and non-vibrating. Analyzing signal frequency features helps to detect where there is a vibration. The y-axis of the generated data was used to classify. The data is divided into windows to handle shorter and more manageable segments. First, a Fast Fourier Transform (FFT) is applied to each window of the data to analyze it in the frequency domain. Then, the power spectrum is calculated and the RMS value is computed. These RMS values are chosen as the key feature for the K-means algorithm to separate the data into two clusters. The RMS value represents the energy of the signal. Vibrating signals will have higher RMS values, while non-vibrating signals will generally have smaller RMS values [5].

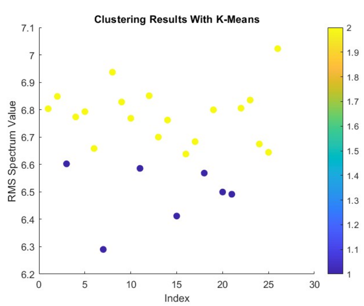

Fig. 11. K-Means classification visualization plot (yellow points are vibrating points)

To evaluate the accuracy of the model we calculated Silhouette Value. This is a metric that measures how much each data point belongs to its cluster. To assess the accuracy of the model these values are averaged. As closer these values to 1 the better the clustering. We calculated the average silhouette value as 0.71133 for y- axis. This result is good enough.
Silhouette Score

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{3}$$

Average Silhouette Value

$$S = \frac{1}{N} \sum_{i=1}^{N} s(i) \tag{4}$$

$a(i)$: Average distance to other data points in the same cluster

$b(i)$: Average distance to the nearest other cluster

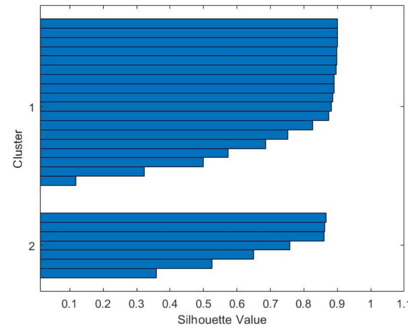

Fig. 12. Silhouette plots of each cluster

VIII. Vibration Compensation

A. Denoising Autoencoder

To denoise the generated signal a deep learning model which is called **denoising autoencoder** is used. Denoising autoencoder has 3 layers essentialy. The first one is encoder that compresses the input data, the second one is bottleneck as a lower dimentional representation of the input, and the last one is the decoder that returns the compressed data back. The input of the network is partially corrupted data which is the noisy data. This corrupted data is created from the clean data. The output of the network is the less noisy data. This output is ideally the same as the clean data. For this study, the referred clean data is obtained from passing the generated vibration data in previous step to the low-pass filter. The corruption is done by removing the parts that are equal to zero when modulated by 7. This corruption probability distribution is a *bernoulli distribution* with the p value 0.857, shown in Fig. 13. As a result, 14.3 percent of the clean data is removed or corrupted.

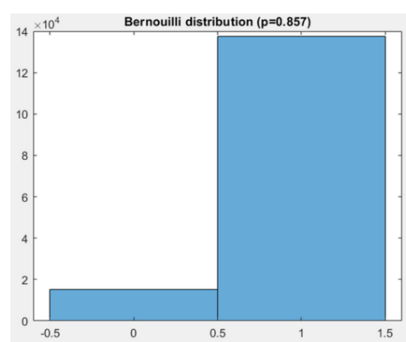

Fig. 13. Corruption probability distribution

B. Network Layers

The layers used in this research are as follows: Image input layer, fully connected layer and regression output. The image input layer is given to the encoder to be compressed, which is also the layer that the input data is zero-centered normalized. Bottleneck is a part of the fully connected layer, and the regression output is used after decoding the input.

C. Training Process

For tranining options of the network, Stochastic Gradient Descent with Momentum optimization method is used. This algorithm takes account the effect of previous descent. The formula of this method is illustrated in the equation (5). γ refers to the momentum coefficient, and v_t refers to the velocity term which is the weighted running sum of all previous gradient vectors.

$$\begin{aligned} v_{t+1} &= \gamma v_t + \eta \nabla L(w_t) \\ w_{t+1} &= w_t - v_{t+1} \end{aligned} \tag{5}$$

Training Option	Value
Momentum coefficient	0.09
Learning rate	0.001
Epoch number	10

TABLE I
TRAINING OPTIONS OF DENOISING AUTOENCODER

D. Result of Deep Learning Model

Resulting loss is nearly $1.6e{-3}$ which is calculated from Mean Squared Error (MSE) between the clean input and less noisy output, as in the equation (6). This type of loss is commonly used for regression tasks.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

After examination of the denoised signal in the frequency domain, it is realized that there is a decrease in the amplitude of the vibration frequency. The decrease is nearly 7 to 5.5 as shown in Fig. 14, comparing the Fig. 10 frequency spectrum before the denoising.

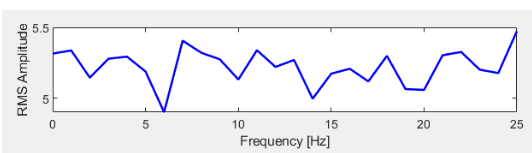

Fig. 14. Y-axis data frequency spectrum after denoising

After looking at the K-Means clustering resuts, an increase in the non-vibration, an thus a decrease in vibration is noticed as in Fig. 15. This is an indicator showing the successful denoising process.

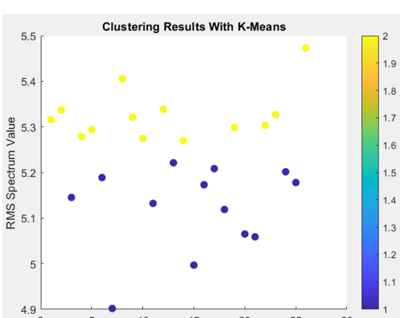

Fig. 15. Classified y-axis data after denoising

Comparison of Fig. 11. and Fig. 15. shows that the denoised data contains less vibration data.

IX. CONCLUSION AND EVALUATION

The resulting graph of the generated y-axis acceleration signal compensation is shown in Fig. 16.

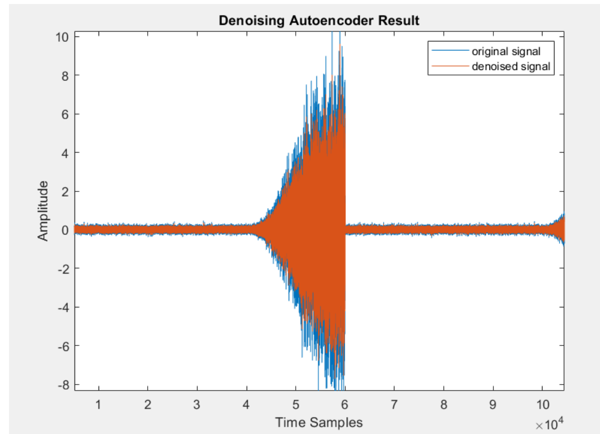

Fig. 16. Synthetic y-axis data after denoising

The resulting graph of the original y-axis acceleration signal compensation is shown in Fig. 17.

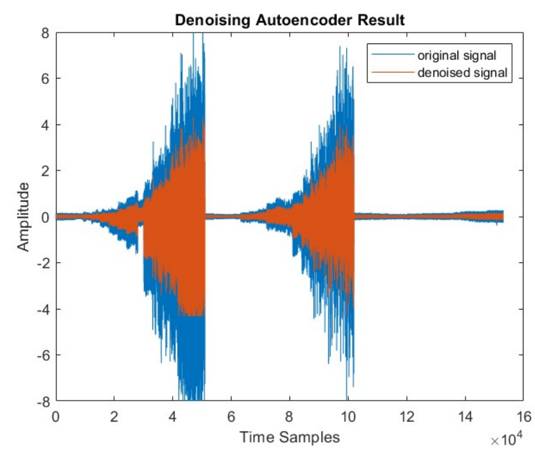

Fig. 17. Original y-axis data after denoising

This study successfully demonstrates the ability to distinguish between vibration and non-vibration states using frequency-based characteristics extracted from acceleration data obtained from IMU sensors. By utilizing the K-means algorithm, an unsupervised learning method, we effectively classified the data into distinct clusters. Building on this classification, the detected vibration data was further processed using a deep learning-based denoising algorithm. This approach not only reduced the noise and vibration effects but also significantly improved the accuracy and quality of the sensor data.

Major Findings:

Frequency Analysis: FFT analysis, can provide important information for machine health monitoring, structural analysis or vibration control.

Feature Extraction: The amplitude values and root mean square (RMS) values obtained from the frequency spectrum were used to distinguish between vibrating and non-vibrating states.

Classification: With the K-means algorithm, the data were classified into two main clusters and these clusters were labeled as vibrating and non-vibrating.

Vibration Compensation: Vibrating data denoised by denoising auto-encoder which is a deep learning technique.

Analyzing the results confirms that the proposed methods effectively addressed the problem. The combination of K-means clustering for classification and denoising auto-encoder for noise reduction offers a comprehensive solution for enhancing the usability and reliability of IMU data in environments subject to vibration with less cost.

X. Future Work

Vibration signal generation process can be also done by Generative Artificial Intelligence (AI) techniques. In a paper on Generative AI, a Generative Adversarial Network(GAN) based advanced model, 1D-WDCGAN, was used to generate vibration data. This model focused on the problem of generating vibration data in the context of structural health monitoring (SHM). As a result, synthetic data with the same statistical properties as real data but with increased diversity is generated, examples of these data are in Fig. 18. [8].

Fig. 18. AI model generated data.

This allows the development of realistic vibration models suitable for different scenarios. This research is helpful as an example of generative AI applications.

References

[1] *Vibration Detection and Rejection from IMU Data*. https://github.com/mathworks/MATLAB-Simulink-Challenge-Project-Hub/tree/main/projects/Vibration%20Detection%20and%20Rejection%20from%20IMU%20Data

[2] D. Capriglione, *Experimental analysis of IMU under vibration* in Testing, Diagnostics and Inspection as a comprehensive value chain for Quality and Safety, Sep. 4AD.

[3] U. Güner, H. Canbolat and A. Ünlütürk, *Design and Implementation of Adaptive Vibration Filter for MEMS Based Low Cost IMU*.

[4] G. S. Sampaio et al., *Prediction of Motor Failure Time Using An Artificial Neural Network*, Sensors, 2019.

[5] I. Muhlisin, R. Rusyadi, *Vibration Analysis on Rotating Machines using Fast Fourier Transform (FFT)*, Annual Conference on Management and Information Technology (ACMIT), 2016.

[6] S. Na et al., *Research on k-means Clustering Algorithm An Improved k-means Clustering Algorithm*, IEEE, 2010.

[7] W. Lee et al., *Noise Learning Based Denoising Autoencoder*, IEEE, 2021.

[8] F. Luleci, F.N. Catbas, O. Avci, *Generative Adversarial Networks for Labelled Vibration Data Generation*, Conference Proceedings of the Society for Experimental Mechanics Series, vol. 5, 2023. https://doi.org/10.1007/978-3-031-05405-1_5