VOL. 21, NO. 2, 2022, 54-63
www.fke.utm.my/elektrika
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

Intelligent Vibration Analysis of Industrial Cooling Fans

Labib Sharrar* and **Kumeresan Danapalasingam**

School of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia.

*Corresponding author: sharrar@graduate.utm.my

Abstract: Industrial cooling fans are responsible for maintaining stable temperatures for delicate components. Therefore, a cooling system failure can certainly lead to machine downtime. Fault Condition Monitoring (FCM) is a predictive maintenance method that can be applied to cooling fans for fault prediction. As the components of a cooling fan wear off, its vibration pattern tends to alter or become more erratic. Thus, this paper uniquely elaborates on three intelligent vibration analysis techniques that are applicable in the FCM of cooling fans. In this research, 1) image encoding with convolutional neural network (CNN), 2) moving average, and 3) fuzzy logic techniques are designed, employed, and their potentials as FCM tools are compared. The vibration data is collected from an experimental test bench that consists of a fan, an accelerometer, and a microcontroller, among others. Once sufficient training data is obtained (11000 data points for each of the fan's conditions), the three vibration analysis models are trained on that data using Python and MATLAB. The results reported in this paper illustrate the accuracy of these intelligent vibration analysis techniques in detecting faults in cooling fans. The novelty of the research revolves around the fan fault detection techniques that are being compared. The image-encoding technique described in this paper has yet to be applied for fault classification and detection. Additionally, while fuzzy logic and moving average are popular methods, this is the first time that they are being used for vibration analysis of cooling fans. Furthermore, this is also a novel comparative study of different vibration analysis techniques.

Keywords: Convolutional Neural Network, Moving Average, Fuzzy Logic, Predictive Maintenance, Fault Condition Monitoring.

Article History: received 24 March 2022; accepted 20 July 2022; published 25 August 2022.

1. INTRODUCTION

Industrial machines consist of various parts that must be kept at stable temperatures. To prevent these components from overheating, a good cooling system is a necessity. Thus, industrial cooling fans are required to operate at optimum levels to maintain proper temperatures for industrial machines. However, fans can be victims of sudden failures [1]. Should a cooling system failure occur, it may result in machine instability or downtime, as well as a reduction of manufacturing yield. Fortunately, with the rise of the Fourth Industrial Revolution, fault analysis and predictive maintenance techniques have begun to play an increasingly important part in industries. Fault condition monitoring or FCM systems for short can enable technicians to identify structural, mechanical, or electrical faults at an earlier stage before the issue spirals further out of control. As such, techniques to forecast possible system defects are garnering more and more attention every day [2]. The FCMs can be especially valuable for predictive maintenance of rotating mechanical machinery, such as cooling fans, which play a significant role in industries [3]. Predictive maintenance (PdM) systems can allow us to avoid manual checking procedures that are time-consuming and prone to errors. As reported by the work of [4], when it comes to PdM, analyzing or monitoring vibration signals is one of the most effective methods to diagnose faults in rotational equipment, since it varies along with the change in the state of mechanical components. Therefore, this paper presents a novel comparative study on vibration analysis of industrial cooling fans to identify faults. The study compares the performance of three vibration analysis techniques: (1) image encoding and convolutional neural network (CNN), (2) moving average and (3) fuzzy logic.

Through image encoding techniques, CNNs have recently gained prominence in vibration analysis due to their ability to extract temporal information from data as shown by the work of Zhao et al. [5]. Image encoding can convert portions of sequential data to two-dimensional colored images and then the texture features can be extracted from these converted images to gain insight into the state of the mechanical part [6]. A work done by Sanchez et al. [7] which involves the use of vibration analysis in bearings for failure prevention, displayed the use of CNN and image encoding in classifying vibration signals from bearings. A paper by Chen et al. [8], used an encoding method called gramian angular field (GAF) and CNN to classify patterns from candlestick graphs. Another encoding method suggested by Dekhane et al. [6], converted the incoming sequential data to grey-scale

images that could be classified by CNN models to identify trends in data. Due to the potential of CNN to classify sequential data, it was selected as a vibration analysis technique for this paper. It is used in conjunction with an image encoding technique proposed by Hur et al. [9]. The method was created for classifying human activities through the accelerometer data. Therefore, the originality of this specific vibration analysis technique lies in the fact that it is being applied in this research for classifying faults.

As for the other two methods, moving average [10] and fuzzy logic [11], they are already considered traditional fault diagnosis techniques, along with k-nearest neighbor (KNN) [12] and support vector machine (SVM) [13]. An example of a fuzzy logic-based vibration analysis technique can be found in the work of Mukane et al. [14], which proposed a LabVIEW-based implementation of a fuzzy logic system to identify machine defects by carrying out vibration analysis. Although moving average and fuzzy logic have been used for rotational equipment, such as motors, they have not yet been applied for fault diagnosis of cooling fans. In this paper, these two techniques along with CNN are for analyzing vibration data and classifying faults in a cooling fan, which further contributes to the novelty of this study. This paper is organized as follows: the Methodology section describes the experiment that was conducted to apply the three techniques, while the Results and Analysis section illustrates the outcome of each technique.

2. METHODOLOGY

As the main objective of this paper is to present a system that detects faults through vibration analysis, an experimental setup with an accelerometer was carried out. The accelerometer was wired to an Arduino Nano (ATmega328) microcontroller board. The overall steps for the experiments described in this section are (i) data collection from the experimental setup, (ii) using a filter or algorithm to sort and clean the acquired vibration data, (iii) using the cleaned data to train the machine learning or deep learning model, validating the trained model, evaluating the trained model and deploying the model in real-time on the testing data to detect faults.

First and foremost, to create a machine learning or deep learning model for this system, the most important step would be to collect data. Thus, an electronic sensor circuit was attached to a fan and the data from the circuit was serially logged into a single board before being saved in .csv format. The next few subsections give a detailed explanation of the system overview and the experiments that were carried out.

2.1 System Overview

The working sequence of the system has been explained in this subsection. Once the machine learning model has been trained, it is deployed on the system. Based on the model's output, the system will decide whether there is a fault in the cooling fan or not. The system overview has been shown in Figure 1.

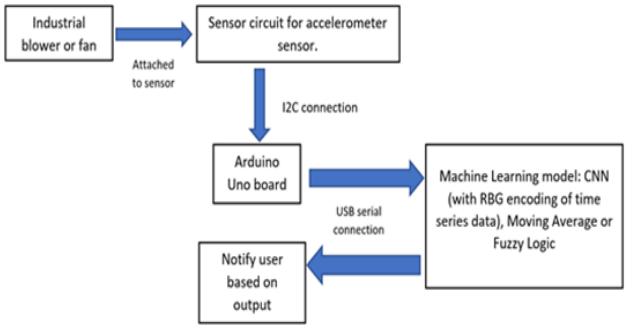

Figure 1. The overview of the entire system.

As mentioned earlier, the performance of three machine learning techniques (CNN, MA, and fuzzy logic) has been investigated in this paper. The sensor circuit is interfaced with the Arduino board through I2C, which in turn would be serially connected to a single board computer. The CNN, moving average, and fuzzy logic models would be running on the computer. Thus, the models analyze the data, and should any faults be detected, the user will be notified.

2.2 Experimental Setup

While there are a plethora of microcontroller boards in the market, an Arduino was chosen due to its user-friendly nature and ease of programming. Among the accelerometers compatible with Arduino, the MPU6050 happened to show the highest level of sensitivity in comparison to the accelerometers ADXL335 and ADXL337, which is why it was selected for this project. A circuit board was made to sample data at a frequency of 50Hz. The accelerometer sensor is connected to the Arduino board through the I2C interface. The circuit diagram has been shown in Figure 2.

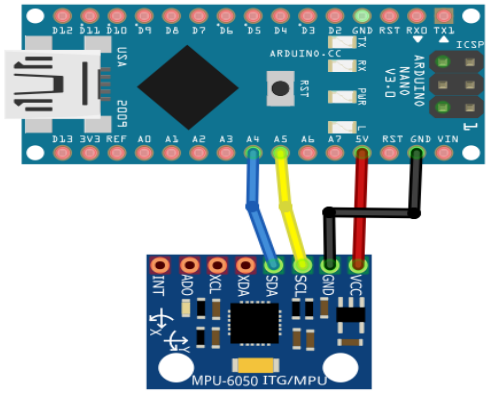

Figure 2. Schematics of the electronic circuit to collect vibration data from the fan.

Data acquisition requires a storage device. For simplicity, the microcontroller was interfaced with a single computer to store the data. As the circuit (shown in Figure 3) samples the vibration data, the sampled data would be serially sent to a single board computer to be saved in .csv format.

While the purpose of this paper is to compare vibration analysis techniques of industrial fans, experiments for this comparative study had to be done using a regular table fan. It is important to remember that when a cooling fan suffers from a defect, its vibration will change [1]. With the intent to experiment, we manually changed the fan speed to simulate the failure conditions. The table fan has three different modes of speed: 0, 1, and 2. Just for experimenting, speed modes 0 and 1 were considered normal speeds, while speed mode 2 was selected as the faulty mode. To begin data acquisition, the circuit board was attached to the outer metal casing of the fan, as shown in Figure 3.

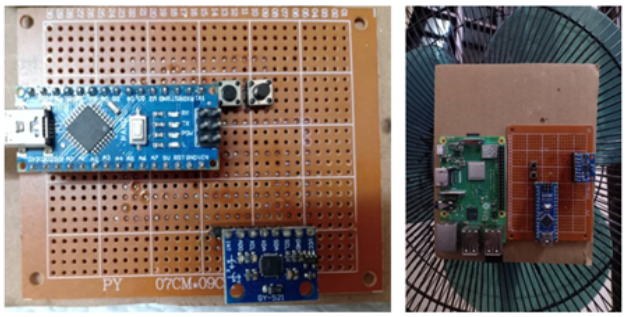

Figure 3. The experimental setup was carried out on a regular fan as shown in this image.

The Arduino Nano board and the accelerometer were soldered together as shown in the above figure. In the experiment, the board was serially interfaced with the single board computer known as Raspberry Pi for data logging. Since the size of a Raspberry Pi is small, it was possible to make the experimental setup more compact. The data for each fan mode were categorized into folders. In Table 1, the acceleration data arrangement has been shown with units for each axis in m/s2.

Table 1. Structure of the collected data.

X-Axis (m/s^2)	Y-Axis (m/s^2)	Z-Axis (m/s^2)
-6.538618	15.561279	1.795532
-7.182129	17.356812	-0.598511
-5.386597	18.553833	2.394043

2.3 Fault Classification using Image Encoding and CNN

Convolutional neural networks (CNNs), which are modeled after the biological vision system, are a family of neural networks explicitly used for image recognition and classification [15]. In this paper, an image encoding technique proposed by Hur et al. [9] has been applied to obtain the colored images from the data set. While the formula proposed by the paper was originally developed to classify human activities, it is being used in this paper to classify vibration trends of the cooling fan. Thus, this application itself is a noble contribution. Furthermore, while the applied formulas were indeed taken from a journal, a three-dimensional array of (900, 450, 3) (900 rows, 450 columns, and 3 depth dimensions) was used to host the pixel values calculated from the formula. The NumPy Python library was used to create the array. The pixel values were iterated over the array to produce the RGB image. Thus, this step adds an algorithmic novelty to the presented CNN-based vibration analysis method.

Initially, to go through the image encoding phase, the vibration data from the fan are normalized. The specific formulas (equations (1), (2), (3), and (4)) were used to encode the data in three colored channels (red, green, and blue). For a clearer explanation of how the time series data was converted to colored images, the steps have been illustrated below. It should be noted that all the formulas mentioned in this section were referenced by Hur et al. [9].

Step 1 - Normalize all the accelerometer signals and scales to 255 using the formulas shown in Eq. (1), (2), (3), and (4).

$$D = \begin{bmatrix} x1 & y1 & z1 \\ \vdots & \vdots & \vdots \\ xN & yN & zN \end{bmatrix} \quad (1)$$

$$\bar{x} = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (2)$$

$$\bar{y} = \frac{y - \min(Y)}{\max(Y) - \min(Y)} \quad (3)$$

$$\bar{z} = \frac{z - \min(Z)}{\max(Z) - \min(Z)} \quad (4)$$

In Eq. (2), D represents the array to hold the vibration readings from the axes x, y, and z for every instant until a certain instant value of N (for example 450) is reached. In the latter three equations of *Step 1*, the symbols $\bar{x}$, $\bar{y}$ and $\bar{z}$ represent the normalized values of the three axes.

Step 2 - Convert the normalized data into three integers that correspond to pixel values in red, green, and blue colored channels. As such, for each sample (each sample of x, y, and z), three different pixel types would be produced as shown by Eq. (5), (6), and (7).

$$R\bar{x} = |\bar{x}| \quad \text{Eq. (5)}$$

$$G\bar{x} = |(\bar{x} - |\bar{x}|) \times 10^2| \quad \text{Eq. (6)}$$

$$B\bar{x} = |(\bar{x} \times 10^2 - |\bar{x} \times 10^2|) \times 10^2| \quad \text{Eq. (7)}$$

The variables $R\bar{x}$, $G\bar{x}$ and $B\bar{x}$ in the three equations of Step 2, represent the colors red, blue, and green for the x axis values. The same operation would be carried out for the corresponding values of the y and z axes.

Step 3 - The colored pixels generated over a certain interval would be stored in matrices. Storage of the generated pixels into three different matrices of red, blue, and green as displayed in Eq. (8), (9), and (10).

$$R = \begin{bmatrix} Rx1 & Ry1 & Rz1 \\ \vdots & \vdots & \vdots \\ RxN & RyN & RzN \end{bmatrix} \quad (8)$$

$$G = \begin{bmatrix} Gx1 & Gy1 & Gz1 \\ \vdots & \vdots & \vdots \\ GxN & GyN & GzN \end{bmatrix} \quad (9)$$

$$B = \begin{bmatrix} Bx1 & By1 & Bz1 \\ \vdots & \vdots & \vdots \\ BxN & ByN & BzN \end{bmatrix} \quad (10)$$

Step 4 – Finally the three colored matrices are concatenated into a single image. The overall process of image encoding has been explained in Figure 4.

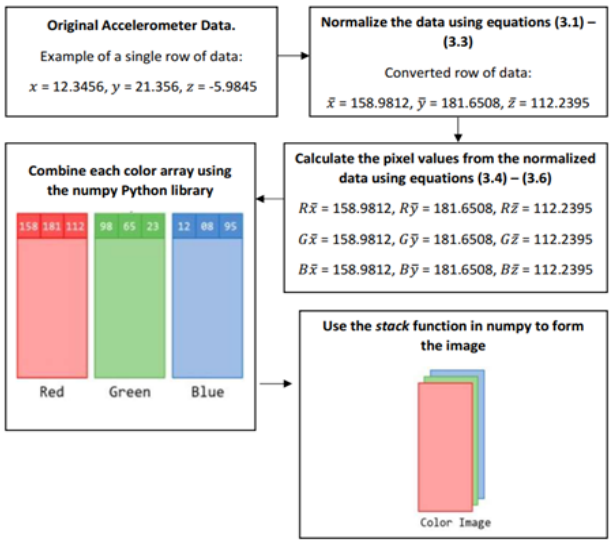

Figure 4. Block diagram representing the image encoding process.

While the formulas used for obtaining the RGB images were the same as proposed by *Hur et al.* [9], the pixel array was arranged differently. A Python library called NumPy was used to make an array of dimensions (900,450,3) (meaning 900 rows, 450 columns, and 3 channels) to host the RGB pixels. During the real-time implementation, the accelerometer is to be attached to the cooling fan and the vibration data can be serially compiled on a computer. After sufficient data are collected, they are categorized to go through the image encoding phase. The images obtained are classified by a trained CNN model. The convolutional neural network in this project was created using the PyTorch deep learning framework. Just as there are many categories among machine learning techniques or different types of neural networks, there are several CNN architectures. Among them, the Resnet and the VGG series are the most powerful algorithms for image classification. Therefore, the compiled RGB images were trained on the Resnet50 and the VGG16 architecture. Under the *Results and Analysis* section, their accuracy is compared.

2.4 Fault Classification Using Moving Average and Euclidean Norm

Moving average (MA) is a type of statistical method that calculates the averages of different subsets in a data set. An average of each point within a certain interval is calculated. Although it is the simplest of the three techniques described in this paper, MA is considered a powerful filtering technique for time series or sequential data. It is effective for removing noise or fluctuations from the data [16]. Furthermore, it is less computationally expensive in comparison to CNN and fuzzy logic. While MA is a fairly old method, it has yet to be applied for fault classification in cooling fans as presented in this paper. Additionally, MA is rarely used for fault classification. The fact that this method has been used for finding faults in a cooling fan presents a contribution to this paper. The formula for MA has been shown in Eq. (11), where A_n represents the average of a set of values at a certain interval and n represents the total number of intervals:

$$MA = \frac{A_1 + A_2 + \cdots + A_n}{n} \tag{11}$$

The MA is used in conjunction with another method called Euclidean norm, which means the straight-line distance between two points in Euclidean space. Since, in this paper, Python is the primary language for data analysis, the NumPy library was used for calculating the Euclidean Norm between points. The equation for Euclidean Norm is displayed in Eq. (12).

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{12}$$

The variables p and q represent two points in Euclidean n-space. As for the function $d()$, it represents the function for Euclidean distance. Finally, the q_i and p_i are cartesian coordinates, whereby the i stands for a particular instance of the coordinate (for example p_1, p_2, p_3 or q_1, q_2, q_3). By using the combination of MA and Euclidean norm methods, it may be less onerous to observe the changes in vibration under faulty conditions, since averaging can be a powerful filtering technique. In the results section, the output of this method is compared with the other two methods in terms of accuracy and practicality. The MA method was applied separately to all three different fan speeds. From the graph outputs, it is possible to visualize the level of vibration at each axis as the speed changes. If this technique is efficient enough, it will be possible to identify the maximum and minimum levels on the MA graphs and use simple if-else operations to classify trends in vibration data. Hence, MA represents a simple method for identifying faults in cooling fans.

2.5 Application of Fuzzy Logic

Technically, fuzzy logic refers to variables that may be any real number between 0 and 1 inclusive. It is based on the concept of partial truth, where the true value of a variable may range from being either true or false [17]. The fuzzy logic technique employs a form of reasoning like humans since its decision-making process involves intermediaries between yes and no [18]. As one of the oldest machine learning methods, it has been used in a variety of electronics ranging from rice cookers to washing machines [19, 20]. In a fuzzy logic system, the input data is first put through fuzzification, whereby it is converted to a value between 0 and 1. After that, the membership functions of the input are constructed, which make up the rule base and the inference engine. The inference and the rule bases are the ones that classify the data. The classification result goes through defuzzification before being sent as an

output. A typical block diagram of a fuzzy logic system has been displayed in Figure 5.

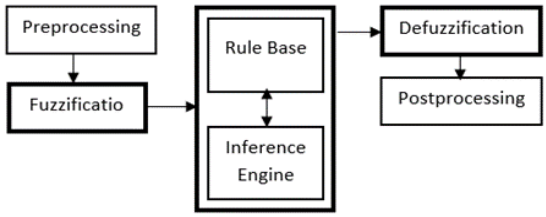

Figure 5. Block diagram of the fuzzy logic method (the highlighted blocks make up the fuzzy logic controller).

To apply fuzzy logic to predict faults in industrial fans would be to construct membership functions to classify the vibration data. As mentioned earlier, the data from the experimental setup are categorized into three files based on the fan speed. The accelerometer measures the vibrations along the x, y, and z axes. Thus, membership functions must be constructed for each axis. For each axis, the membership functions will classify the fan's overall acceleration (Low, Medium, and High) based on the input data. To construct the membership functions, it was necessary to see the distribution of data along each axis at each speed. For that, histograms were plotted in each case. Once it was possible to observe the distribution of data, boundaries were formed for the fuzzy logic membership functions. Through trial and error, triangular pulses were plotted over each distribution to contain most of the data. The data distribution of the axes x, y, and z are displayed in Figures 6, 7, and 8, respectively.

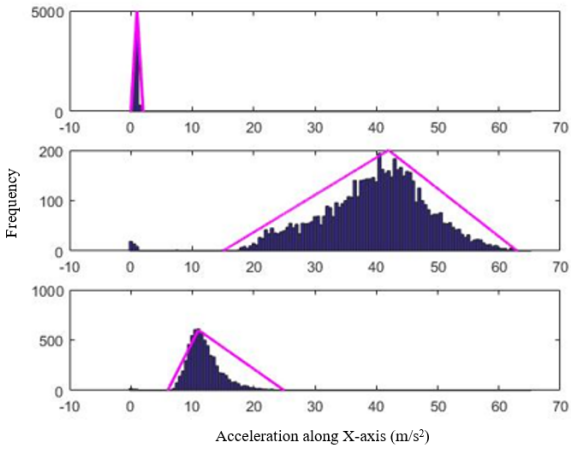

Figure 6. Data distribution along the x-axis at the three different speeds with the estimated triangular pulse.

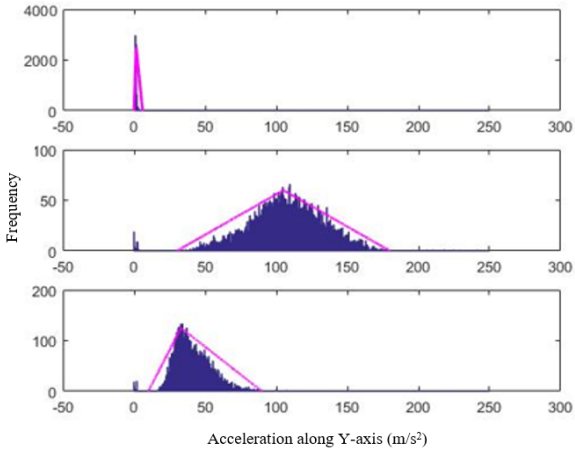

Figure 7. Data distribution along the y-axis at the three different speeds with the estimated triangular pulse.

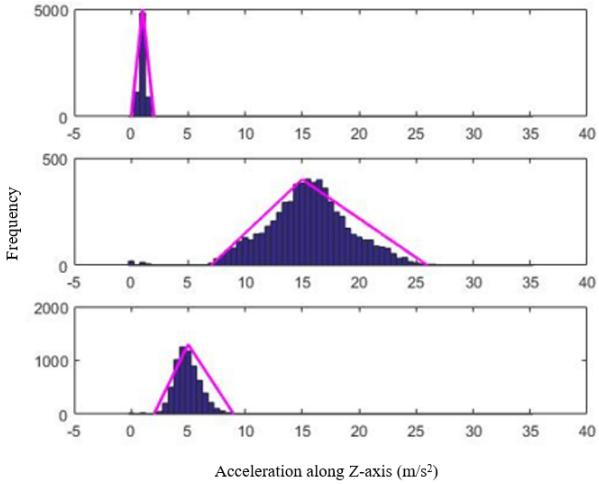

Figure 8. Data distribution along the z-axis at the three different speeds with the estimated triangular pulse.

With a proper understanding of the data distribution complete, the membership functions are plotted. The horizontal range of the values in the distribution remained the same, but the range along the vertical axis was changed from 0 to 1. The membership functions derived from the histograms and triangular plots are displayed in Figures 9, 10, and 11.

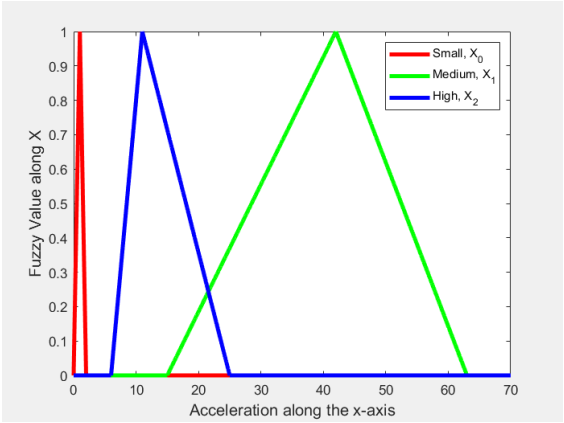

Figure 9. Membership Functions for the X-axis acceleration.

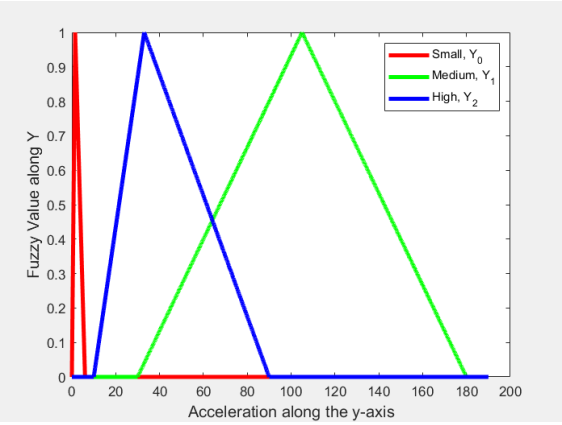

Figure 10. Membership Functions for the Y-axis acceleration.

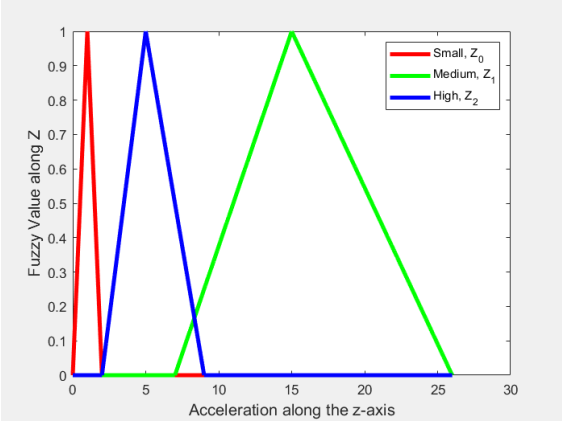

Figure 11. Membership Functions for the Z-axis acceleration.

The membership functions represent the fuzzy conditions that are used to classify the status of the cooling fan motor. In Figures 9, 10, and 11 the membership functions are labeled as *small*, *medium,* and *high*, which represent the speed modes 0, 1, and 2 of the cooling fan respectively. However, before that, we need to know what conditions are used the classify the acceleration along the three axes. For example, if the acceleration value at the x-axis is between 0.11 m/s^2 and 2.43 m/s^2, the state of the x-axis acceleration would be small (the state of the fan at speed mode 0). The conditions to classify the acceleration for each axis are shown below in Table 2.

Table 2. Conditions to classify each variable.

Accelerometer Axis	**Range for acceleration value: A (m/s^2)**	**Acceleration Status**
X-axis	$0.1 < A \leq 2.4$	Small
	$14.9 < A \leq 62.7$	Medium
	$6.1 < A \leq 25.4$	High
Y-axis	$0 < A \leq 6.0$	Small
	$29.8 < A \leq 180.2$	Medium
	$10.2 < A \leq 90.6$	High
Z-axis	$0.0 < A \leq 2.1$	Small
	$7.1 < A \leq 26.0$	Medium
	$2.1 < A \leq 9.0$	High

For the fuzzy logic system to be able to classify the input data, fuzzy rules must be established to classify the cooling fan motor's conditions. Since different fan speed modes were used to simulate the fan conditions, the fuzzy logic system classifies the speed modes through the rules displayed in Table 3.

Table 3. Fuzzy rules to classify the fan's condition.

No.	**X Accel.**	**Y Accel.**	**Z Accel.**	**Accel.**	**Condition**
1	Small	Small	Small	Small	OK
2	Small	Small	Medium	Medium	OK
3	Small	Medium	Small	Medium	OK
4	Small	Medium	Medium	Medium	OK
5	Medium	Small	Small	Medium	OK
6	Medium	Small	Medium	Medium	OK
7	Medium	Medium	Small	Medium	OK
8	Medium	Medium	Medium	Medium	OK
9	Small	Small	High	High	FAULTY
10	Small	Medium	High	High	FAULTY
11	Small	High	Small	High	FAULTY
12	Small	High	Medium	High	FAULTY
13	Small	High	High	High	FAULTY
14	Medium	Small	High	High	FAULTY
15	Medium	High	Small	High	FAULTY
16	Medium	High	Medium	High	FAULTY
17	Medium	High	High	High	FAULTY
18	High	Small	Small	High	FAULTY
19	High	Small	Medium	High	FAULTY
20	High	Medium	Small	High	FAULTY
21	High	Medium	Medium	High	FAULTY
22	High	Small	High	High	FAULTY
23	High	High	Small	High	FAULTY
24	High	Medium	High	High	FAULTY
25	High	High	Medium	High	FAULTY
26	High	High	High	High	FAULTY

The input data from the accelerometer sensor is classified based on the range of conditions shown in Table 2 and the fuzzy rules displayed in Table 3. If the overall status of the fan's acceleration is in **Small** or **Medium** mode, the status of the fan is **OK**. Should the acceleration of the fan be **High**, the fan's motor is considered **FAULTY**. For clarity, a block diagram representing the explanation is shown in Figure 12.

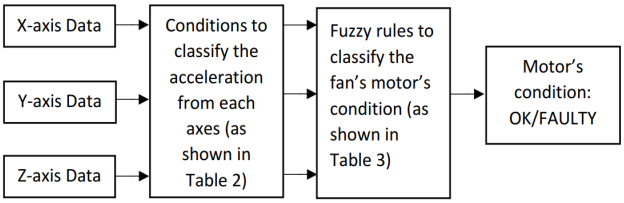

Figure 12. Process of classification using the fuzzy logic technique.

3. RESULTS

3.1 Results of CNN and Image Encoding

After data collection from the accelerometer had been complete, the data were categorized into three separate files to better analyze the vibrations at three different fan speeds. Firstly, as mentioned in the methodology section, the vibration readings were converted into colored images using formulas from the Iss2Image image encoding by

Hur et al. [9]. Results of the conversion are shown in Figure 13. After all the raw accelerometer data had been converted to colored images, the output results were prepared for training.

Figure 13. Samples of the colored images were produced for speeds 0 (left), 1 (center), and 2 (right).

Finally, after all the raw accelerometer data had been converted to colored images, the output results were prepared for training. The image data was split into three directories as always necessary for training a machine learning model*: train*, *validation,* and *test*. Once that was done, the data was trained on the two CNN architectures, Resnet50 and VGG16. For each architecture, the data was trained for 100 epochs with a batch size of 8. Graphs were plotted to illustrate the performance of each model during the training and the testing phase, as shown in Figures 14, 15, 16, 17, 18, and 19. The overall performance of each architecture has been tabulated in Table 4.

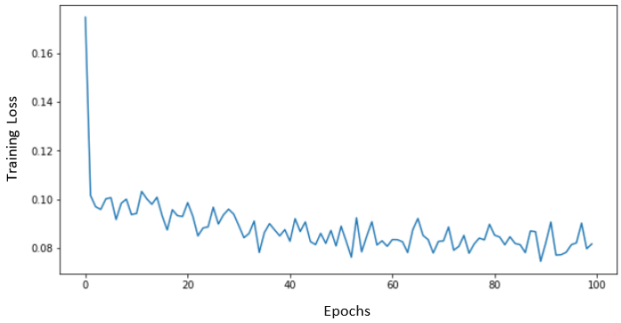

Figure 14. Performance of Resnet50 during training.

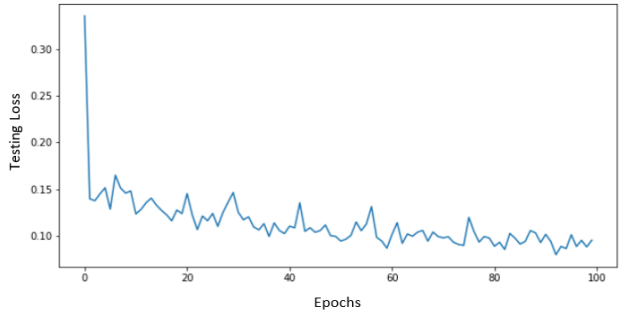

Figure 15. Performance of Resnet50 during testing.

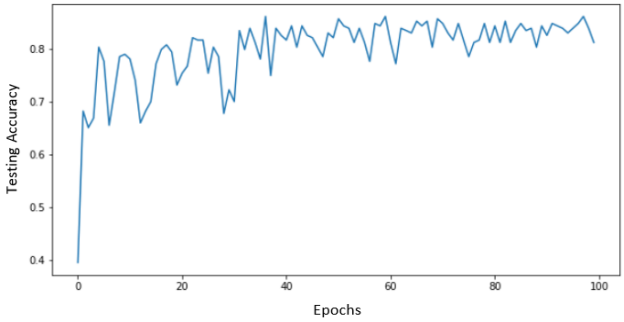

Figure 16. Testing accuracy of the Resnet50 model.

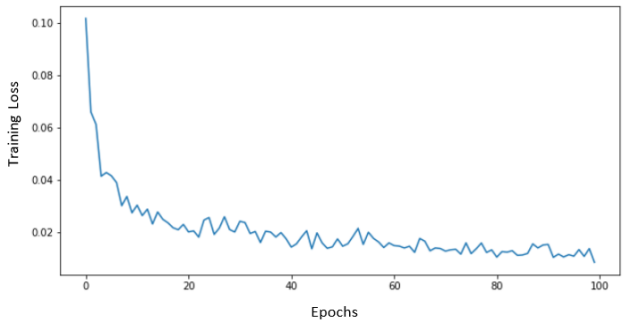

Figure 17. Performance of VGG16 during training.

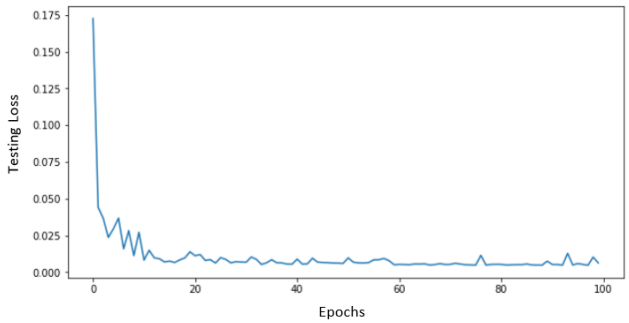

Figure 18. Performance of VGG16 during testing.

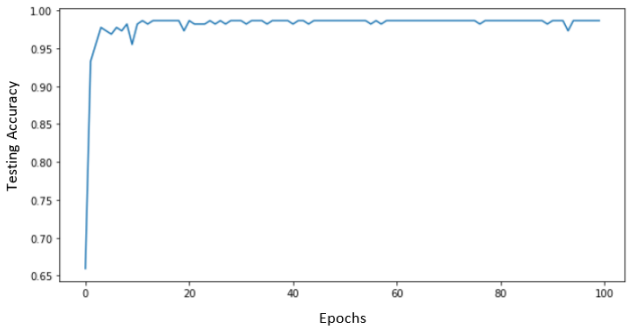

Figure 19. Testing accuracy of the VGG16 model.

Table 4. Results of the two CNN architectures.

CNN Model	Training Loss	Testing Loss	Testing Accuracy
Resnet50	8.2 %	9.5 %	81.2 %
VGG16	0.9 %	0.6 %	98.7 %

Based on Table 4, the VGG16 model performed better than the Resnet50 model, as it has a higher testing accuracy. If we compare the graphs of testing accuracy for both models, we can see that the testing accuracy mostly remained steady for the VGG16 model, unlike for Resnet50 where the testing accuracy fluctuated tremendously over the 100 epochs. However, training VGG16 took longer since it is much deeper. Nevertheless, both models could have been slightly overfitted. Improvements to training should be made in the future. Overall, when using the trained model to make predictions on testing data, VGG16 displayed much higher accuracy. The testing was done in batches of 4 images at a time. The output displays for each model are tabulated in Figure 20.

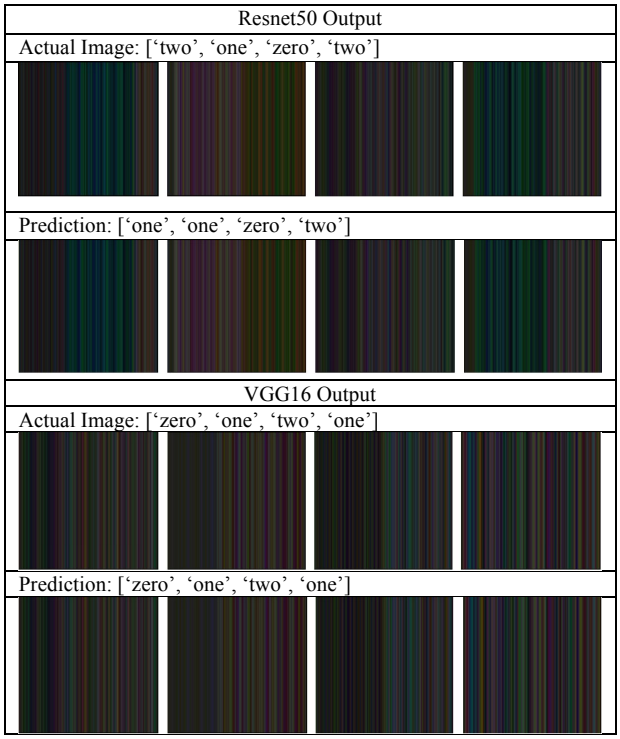

Figure 20. Comparison of output results.

As seen from the display, for this batch the VGG16 predicted all the samples correctly while the Resnet50 made a few mistakes. Although VGG16 did display high accuracy, it should be noted that neural networks work best when trained with large datasets [15]. The dataset for this project was relatively small. Therefore, results for both CNN architectures may be better when trained with more data.

3.2 Results of the Moving Average Method

The application for the MA was simple. The collected data were analyzed using the Python libraries pandas, NumPy and Seaborn. Raw data from the accelerometer were taken through the Euclidean norm and MA. In the graphs, for a clearer understanding, the MA line was highlighted in black. Results from this method were comparatively easier to understand than the other methods. The graphs indicated the threshold as shown in Figures 21, 22, and 23. Should this method be sufficiently accurate, it can be possible to identify faults using the basic if-else statements rather than using complex machine learning or statistical methods.

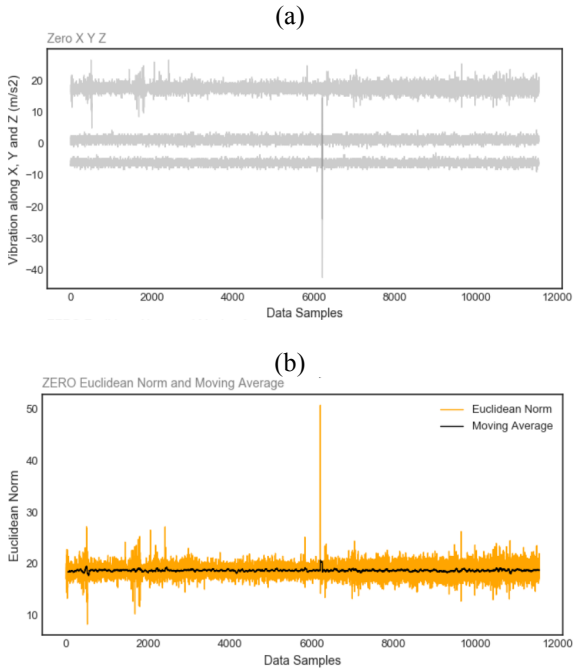

Figure 21. Euclidean (a) and Moving Average Graph (b) for Fan Speed 0.

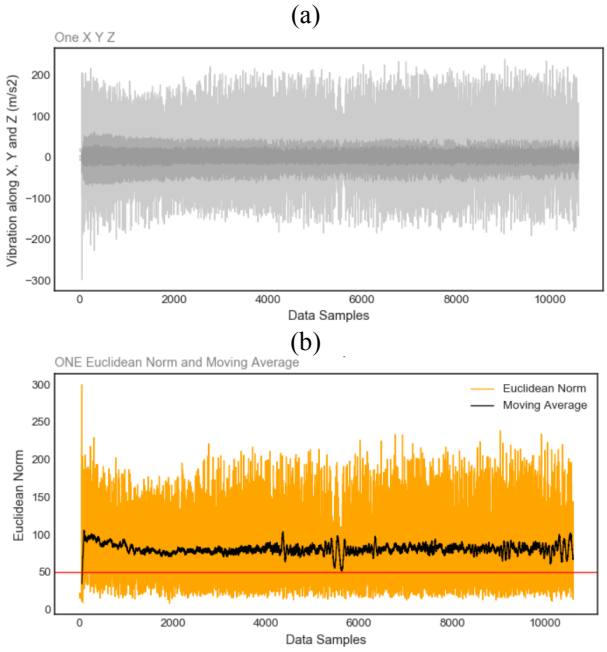

Figure 22. Euclidean (a) and Moving Average Graph (b) for Fan Speed 1.

The red lines in figure 23 indicate the levels above or below which the samples from the compiled data will be categorized into different speed modes. Thus, by looking at Figures 23, 24, and 25, we can estimate that if the MA is above 30 and below 50, the fan speed is at level 2, which is indicative of a faulty system. The indicator levels were applied to the vibration dataset to estimate the validation and evaluation accuracy. Firstly, the dataset was split into

training and testing data per the rules. The result for both parts displayed high accuracy as shown in Table 5.

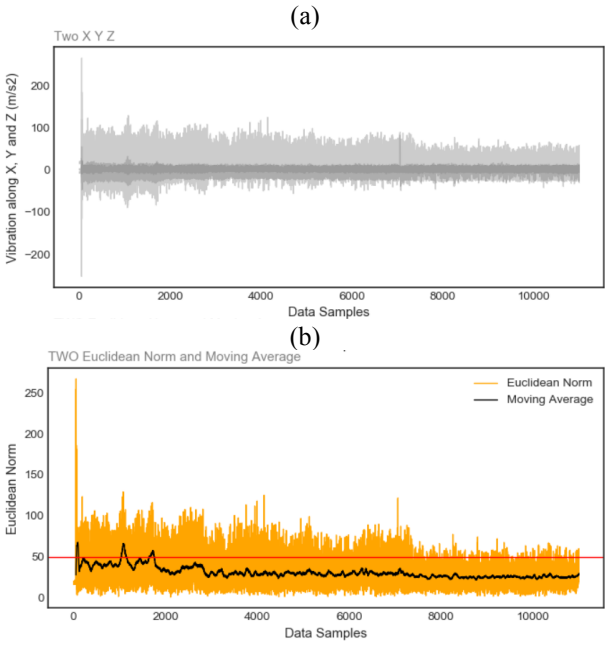

Figure 23. Euclidean (a) and Moving Average Graph (b) for Fan Speed 2.

Table 5. Validation and Evaluation Accuracy of MA.

Validation Accuracy	99.2%
Evaluation Accuracy	99.3%

Based on the results of Table 5, it is clear that although the MA technique is simple, the results have proven to be highly accurate for this experiment.

3.3 Results of the Fuzzy Logic Method

Membership functions for each of the accelerometer axes were used to classify the testing data of the cooling fan. Classifications were done using the conditions shown in Table 2 and the fuzzy logic rules mentioned in Table 3. The validation and evaluation accuracies were recorded. Overall, the results were favorable. The validation (training) and evaluation (testing) accuracies of the fuzzy logic technique are displayed in Table 6.

Table 6. Validation and Evaluation Accuracy.

Validation Accuracy	96.1%
Evaluation Accuracy	98.1%

3.4 Comparison of the Different Methods

Now we have finally come to the part where we compare the three different methods. Among the three techniques, classification using Image Encoding and CNN was the most sophisticated, while MA was the simplest. All the methods displayed high accuracy. Evaluation accuracies have been tabulated below in Table 7.

Table 7. The accuracy of each technique.

CNN (VGG16)	Moving Average (MA)	Fuzzy Logic
98.7%	99.3%	98.1%

According to the results, despite being the simplest method, MA displays the highest accuracy in fault prediction. The VGG16 model from the CNN technique has the second-highest accuracy and Fuzzy Logic has the lowest number in this regard. Nevertheless, all three methods showed exceptionally high accuracies and can easily be applied to identifying faults in the experimental setup.

When selecting the best method among the three, the logical choice would be to select the one with the highest accuracy, which is the MA method. However, some considerations must be made before deciding on the best method. It should be noted that the data collected for in project was only from an experimental setup rather than an industrial one. It may be possible that vibration readings collected from industrial sites may suffer from interferences from the surroundings. Thus, the vibration readings may not be as consistent as the ones collected from the hardware setup in this experiment. As a family of neural networks, CNNs can deal with inconsistencies as well as non-linearity. However, it is at the moment unknown how the MA algorithm will perform when it encounters these issues. Thus, more studies should be carried out in the future regarding these vibration analysis techniques to determine which is truly the most accurate.

4. CONCLUSION

The primary purpose of this study was to compare different techniques of vibration analysis for industrial cooling fans. Proper vibration analysis is required to predict faults that may occur in machinery and carry out earlier maintenance before the issue gets out of hand. By having such an intelligent system, companies get avoid downtime, as well as expensive repairs. In this paper, three different vibration analysis techniques were compared based on their potential for predictive maintenance. The fault diagnosis techniques illustrated in this document involve image encoding and CNN, MA, and fuzzy logic.

Since vibration is the necessary variable, an accelerometer was interfaced with an Arduino microcontroller board and attached to a cooling fan. Due to the inaccessibility of an industrial type of fan or blower, an experimental fan was used for this study. The three different speed modes (0,1 and 2) of the fan were used to represent the healthy and faulty conditions (at speed 2 the fan's condition is assumed to be faulty). After the data had been collected and categorized accordingly, they were trained on the three techniques.

As displayed under the Results and Analysis section, the accuracies of the three methods are compared with one another. For the first technique, which involved image encoding and the use of CNN, the VGG16 architecture showed much higher accuracy compared to the Resnet50

architecture. It also showed much lower testing and training loss, which is indicative of its high accuracy.

Regarding the MA technique, it showed validation and evaluation accuracies of 99.2% and 99.6% respectively. As for fuzzy logic, the validation and evaluation accuracies were 96.1% and 98.1%, respectively. Overall, all three techniques showed considerably high evaluation accuracies. Among the three, fuzzy logic was the least accurate (98.1%). At 98.7%, the VGG16 showed the second highest accuracy. Compared to the other two, the CNN-based method was the most complex. Ironically, the MA technique, which was the simplest, displayed the highest evaluation accuracy at 99.3%. Thus, for this particular experiment, the MA vibration analysis technique was more accurate.

However, more comparative studies should be done regarding these vibration analysis techniques. Perhaps in future studies, vibration data can be collected from actual industrial fans at companies and the performances of the three techniques can be compared once more. This can also show which technique is better at dealing with inconsistencies as well as interferences in vibration data at industrial sites.

REFERENCES

[1] H. Taplak, E. Kurt, and M. Parlak, "Fault Diagnosis For Exhaust Fan Using Experimental Predictive Maintenance Method," *Int. J. Acoust. Vib.*, vol. 21, pp. 274–280, 2016, doi: 10.20855/ijav.2016.21.3421.

[2] S. Erkaya and Ş. Ulus, "Investigation of fan fault problems using vibration and noise analysis," in *Recent Innovations in Mechatronics*, 2013, vol. 1, doi: 10.17667/riim.2014.1-2/1.

[3] P. Zhang, T. Yang, J. li, and S. Huang, "Flexible and Smart Online Monitoring and Fault Diagnosis System for Rotating Machinery," 2012, doi: 10.1109/CDCIEM.2012.88.

[4] S. Sp, L. Marulaiah, K. Kumar, and S. K R, "Vibration based Fault Diagnosis Techniques for Rotating Mechanical Components: Review Paper," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 376, p. 12109, 2018, doi: 10.1088/1757-899X/376/1/012109.

[5] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *J. Syst. Eng. Electron.*, vol. 28, no. 1, pp. 162–169, 2017, doi: 10.21629/JSEE.2017.01.18.

[6] A. Dekhane, A. Djellal, F. Boutebbakh, and R. Lakel, *Cooling Fan Combined Fault Vibration Analysis Using Convolutional Neural Network Classifier*. 2020.

[7] L. Pinedo Sánchez, D. Mercado-Ravell, and C. Carballo-Monsivais, "Vibration Analysis in Bearings for Failure Prevention using CNN." 2020.

[8] Y.-C. Tsai, J.-H. Chen, and C.-C. Wang, "Encoding Candlesticks as Images for Patterns Classification Using Convolutional Neural Networks." 2019.

[9] T. Hur, J. Bang, T. Huynh-The, J. Lee, J.-I. Kim, and S. Lee, "Iss2Image: A Novel Signal-Encoding Technique for CNN-Based Human Activity Recognition," *Sensors*, vol. 18, p. 3910, 2018, doi: 10.3390/s18113910.

[10] Y. Yang, W. Wu, and L. Sun, "Prediction of mechanical equipment vibration trend using autoregressive integrated moving average model," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, pp. 1–5, doi: 10.1109/CISP-BMEI.2017.8302110.

[11] H. F. Azgomi and J. Poshtan, "Induction motor stator fault detection via fuzzy logic," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, 2013, pp. 1–5, doi: 10.1109/IranianCEE.2013.6599711.

[12] S. Samanta, J. N. Bera, and G. Sarkar, "KNN based fault diagnosis system for induction motor," in *2016 2nd International Conference on Control, Instrumentation, Energy Communication (CIEC)*, 2016, pp. 304–308, doi: 10.1109/CIEC.2016.7513791.

[13] C. Li, S. Liu, H.-C. Zhang, and Y. Hu, "Machinery condition prediction based on wavelet and support vector machine," in *Journal of Intelligent Manufacturing*, 2013, vol. 28, pp. 1725–1729, doi: 10.1109/QR2MSE.2013.6625909.

[14] R. V Mukane, N. M. Gurav, S. Y. Sondkar, and N. C. Fernandes, "LabVIEW Based Implementation of Fuzzy Logic for Vibration Analysis to Identify Machinery Faults," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 2017, pp. 1–5, doi: 10.1109/ICCUBEA.2017.8463707.

[15] B. Fasel, "Robust face analysis using convolutional neural networks," in *Object recognition supported by user interaction for service robots*, 2002, vol. 2, pp. 40–43 vol.2, doi: 10.1109/ICPR.2002.1048231.

[16] L. Zhang and N. Hu, "Time Domain Synchronous Moving Average and its Application to Gear Fault Detection," *IEEE Access*, vol. 7, pp. 93035–93048, 2019, doi: 10.1109/ACCESS.2019.2927762.

[17] N. K. Alang-Rashid and A. S. Heger, "A general purpose fuzzy logic code," in *[1992 Proceedings] IEEE International Conference on Fuzzy Systems*, 1992, pp. 733–742, doi: 10.1109/FUZZY.1992.258748.

[18] S. H. R. Abbasi and F. Shabaninia, "A research on employing fuzzy composite concepts based on human reasoning through singleton and non-singleton fuzzification," in *2011 IEEE International Conference on Information Reuse Integration*, 2011, pp. 500–501, doi: 10.1109/IRI.2011.6009604.

[19] S. Satpathy and A. P. Sahu, "A Graphical User Interface, Fuzzy Based Intelligent Rice Cooker," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, 2015, pp. 1216–1220, doi: 10.1109/CICN.2015.234.

[20] W. Ai-Zhen and R. Guo-Feng, "The Design of Neural Network Fuzzy Controller in Washing Machine," in *2012 International Conference on Computing, Measurement, Control and Sensor Network*, 2012, pp. 136–139, doi: 10.1109/CMCSN.2012.35.