

Article

A Moving Window-Based Feature Extraction Method for Gearbox Fault Detection Using Vibration Signals

Ietezaz ul Hassan ^{1,2,*}, Krishna Panduru ^{1,2} , Daniel Riordan ^{1,2}  and Joseph Walsh ^{1,2} 

¹ IMaR Research Centre, Munster Technological University, V92 CX88 Tralee, Ireland; krishna.panduru@mtu.ie (K.P.); daniel.riordan@mtu.ie (D.R.); joseph.walsh@mtu.ie (J.W.)

² School of Science, Technology, Engineering and Mathematics, Munster Technological University, V92 CX88 Tralee, Ireland

* Correspondence: ietezaz.ul.hassan@mymtu.ie

Abstract

Early gearbox defect detection is imperative for reducing unplanned downtime, ensuring reliability and efficiency, and minimizing maintenance expenses. In recent years, with the rise of Artificial Intelligence (AI) and digital transformation, gearbox defect detection using AI has gained popularity. Machine learning (ML) classifiers are very popular and transform gearbox condition monitoring from manual to automatic monitoring systems. This work proposes a moving window-based method for extracting statistical features from recorded vibration signals from the gearbox. The extracted features were used to train traditional ML classifiers. Moving window sizes of 300, 400, 500, 600, 700, and 800 were applied to extract statistical features from the publicly available benchmark dataset. The six different moving window sizes caused six types of datasets, each one corresponding to the moving window size. The generated datasets were partitioned using the K-fold cross-validation method to train and test ML models. This study explored and evaluated seven prominent ML classifiers: Decision Tree, Random Forest, Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbor (KNN), Gradient Boosting Classifier (GBC), and Logistic Regression. The experimental results demonstrated that SVM, Logistic Regression, and GBC can outperform other ML classifiers. The experimental results in terms of accuracy, precision, and recall revealed that the ML classifier's performance improves as the size of the moving window used for feature extraction increases.

Keywords: condition monitoring; vibration-based condition monitoring; gearbox fault prediction; ML; statistical features; moving window



Academic Editor: Ke Feng

Received: 18 December 2025

Revised: 28 January 2026

Accepted: 31 January 2026

Published: 4 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Gearboxes are widely used in automobiles, industrial machinery, wind turbines, and electric motors to transfer power from one rotating component to another [1]. A gearbox is made from a set of gears that are used to control torque and rotational speed based on the gear ratio. A gearbox can be manual, automatic, or semi-automatic [2]. Gearbox failure can result in a number of issues, such as loss of power transfer, mechanical damage, overheating, sudden stoppage, lower efficiency, safety hazards, increased vibration [3,4], and many others. A number of variables, such as inadequate lubrication, contamination, excessive stress or overloading, improper gear alignment, poor components, and wear and tear (aging), can cause the gearbox to fail [5]. A gearbox can be prevented from failing by employing several preventative measures, such as ensuring the gears are properly

lubricated, checking the lubricant frequently, avoiding overloading, preventing overheating, performing routine inspections, and so on. These methods fall under the umbrella of preventive maintenance, which involves inspecting and replacing components on a regular basis [6]. Some of these methods are time-consuming, while others are not cost-effective.

Predictive maintenance is a method that uses condition monitoring, data analytics, and ML to predict equipment or a machine's Remaining Useful Life (RUL) [7]. The RUL can be used to estimate when a component needs to be repaired or replaced. This method offers several advantages, including reduced downtime, cost savings, increased equipment useful life, high safety, and data-driven decision-making ability [8]. A number of steps, such as data collection, data analysis and modelling, condition evaluation, and maintenance decision-making, can be employed to perform predictive maintenance for gearbox condition monitoring. A gearbox's predictive maintenance can be efficiently performed by collecting vibration, temperature, lubricant condition, acoustic, load, and torque data [9]. An appropriate sensor can be used to record the data for each data type. The collected data can be processed using several methods for extracting useful insights. The processing methods can be classified as signal processing-based, thresholding, or AI-based methods [10]. For efficient gearbox defect prediction, AI-based techniques generally require huge datasets; however, they often rely less on explicit domain knowledge than physics-based methods. On the other hand, the thresholding-based method uses thresholding logic on a signal parameter, such as amplitude, to monitor the condition of the gearbox [11]. However, signal processing-based methods analyze the signals in a more in-depth way, especially through visualization, such as frequency response, fluctuations, etc. [12].

A problem with constantly monitored machines is that the collected signals have a long interval. When data is collected through continuous monitoring, it remains in that form for months, years, etc. Because of its high dimensions, analyzing such long-interval data with ML is challenging. The goal of this research is to analyze such collected data using a moving window approach, in which a subset of the signals are chosen for analysis. The objective here is to evaluate ML classifiers and determine whether increasing the moving window size used for feature extraction could improve their performance.

1.1. Motivation

In manufacturing and industrial applications, the gearbox is responsible for transmitting power from the motor to other machine components [13]. The rate of power transfer from the motor to other components is controlled by controlling the rotational speed and torque of the gears inside the gearbox [14]. A defective gearbox could cause issues such as power loss, more energy utilization, and inefficient machine performance [15]. A fault-free gearbox can increase the rate of production in manufacturing and industrial operations by transmitting power more effectively and consuming energy more efficiently. Therefore, maintaining a gearbox in the best possible condition could result in increased efficiency, reduced downtime, improved product quality, more safety, and cost savings. Following a safety inspection, a gearbox can be preserved to remain in good working condition [16].

1.2. Problem Statement

A gearbox's condition can be monitored using gearbox operational data. When a gearbox is operating, a number of parameters, including vibration, temperature, speed, and torque, can be monitored [17]. An appropriate method can be selected to process the collected data for extracting useful insights from the collected operational data [18]. In most cases, the data is collected as signals, which can be processed either in the time domain, frequency domain, or time-frequency domain. These domain-based methods require domain knowledge, or a domain specialist, to be able to differentiate between faulty

and healthy data samples. Meanwhile, gearboxes in different environments have different patterns for healthy and defective samples that require domain specialties.

These domain-based techniques do not allow predictive monitoring, because they require an analyst to continuously keep an eye on signals for detecting faults. These methods also failed to predict the RUL. In the era of digital transformation, Industry 4.0, Industry 5.0, and other technological innovations—where processes are moving from manual to automatic processing—humans are prone to errors; this makes existing techniques time-consuming, costly, and insufficient [19].

1.3. Our Contribution

This research analyzed a publicly available benchmark dataset [20] to enable an AI-based gearbox fault detection system. The dataset was collected by recording vibration data from a drivetrain using various load sizes in both healthy and faulty scenarios. This vibration data was recorded using four sensors placed on a gearbox at four different positions. These vibration signals were further processed for feature extraction, which was used to train ML classifiers for gearbox fault detection. The following are the research contributions (RCs) of this paper.

- RC-1: Moving window: In this research, the dataset was processed using a moving window approach for feature extraction. During feature extraction, the moving window size was set to six different values: 300, 400, 500, 600, 700, and 800, that correspond to raw vibration samples. Window sizes of 300, 400, 500, 600, 700, and 800 were chosen to look into how increasing temporal context affects the stability of statistical feature extraction in varied load conditions. This resulted in six datasets, with each one corresponding to the moving window size.
- RC-2: Statistical feature extraction: During the feature extraction, ten statistical features were extracted from each moving window (moving window sizes 300, 400, 500, 600, 700, and 800). These features included the mean, standard deviation, variance, skewness, kurtosis, peak-to-peak (P2P), crest factor, impulse factor, and root mean square (RMS).
- RC-3: Data partitioning: This research methodology is validated using five-fold cross-validation, which allows for reliable performance evaluation. This approach improves generalization, decreases bias from a single train–test split, and allows a more reliable evaluation of model performance for real-world applications.
- RC-4: ML model performance: This study compares the performance of seven ML classifiers based on accuracy, precision, and recall for gearbox fault detection. The ML classifiers were Decision Tree, GBC, KNN, Logistic Regression, Naïve Bayes, Random Forest, and SVM.

The experimental results revealed that the SVM, Logistic Regression, and GBC outperformed the other classifiers. The experimental results also revealed that using a larger moving window size for feature extraction improved the performance of ML classifiers.

The rest of the paper is organized as follows: Section 2 describes the background and literature review. The proposed methodology, which includes the data description, experimental setup, preprocessing, ML classifiers, and evaluation metrics, is discussed in Section 3. The results are presented in Section 4, and the discussion is given in Section 5. Finally, the paper is concluded in Section 6, followed by recommendations for future work.

2. Background and Literature Review

This section covers the gearbox background and literature review regarding gearbox fault detection methods.

2.1. Background

A gearbox is a mechanical device composed of multiple distinct gear types. It works by moving power from the source, such as an engine or motor, to the output device, which could be a vehicle wheel or a rotating component of machinery [21]. A gearbox can have a wide range of gear types, such as helical, spur, worm, miter, bevel, screw, and rack gears. The different gears in a gearbox can be adjusted according to the load or application requirements. It works by controlling the power transfer torque and gear speed [22]. Gearboxes are used in helicopters, bucket wheel excavators, wind turbines, milling machines, and tracked loaders [1], as shown in Figure 1. A gearbox's functionality is dependent on its application. For example, in vehicles, the gears in the gearboxes are used for acceleration, deceleration, balancing, climbing, and so on. In an industrial setting, gearboxes can be used for construction, manufacturing, or energy generation, which allows machinery to operate at varying speeds and loads. A gearbox could experience certain defects due to mechanical wear, environmental factors, operational stresses, and maintenance issues. Some popular gear faults are gear tooth pitting, cracking, chipping, and misalignment.

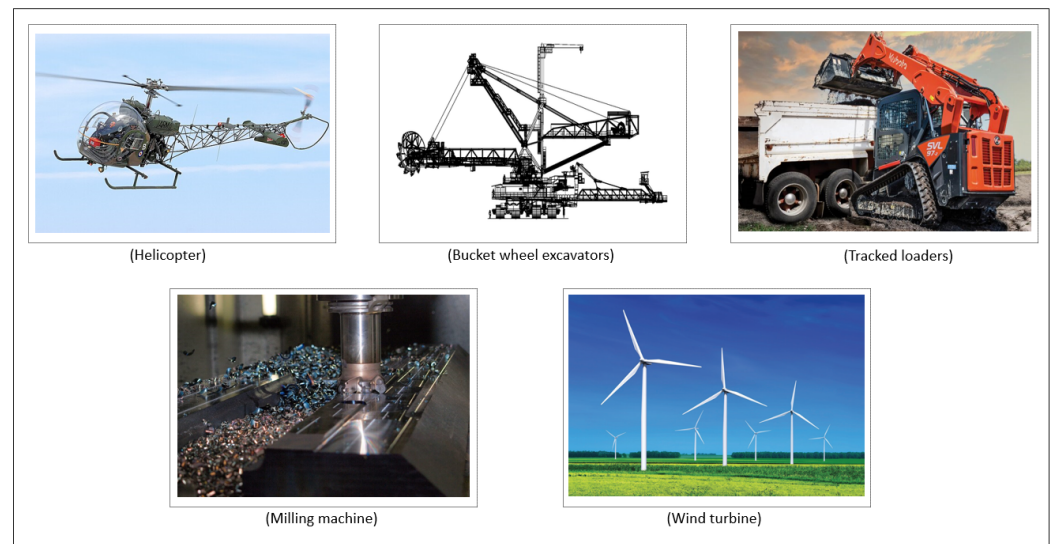


Figure 1. Applications of a gearbox.

2.2. Literature Review

Feng et al. [23] proposed a model for diagnosing planetary gearbox faults in time-varying speed conditions. They evaluated the reciprocal series of the time interval (between consecutive encoder signal pulses) instead of the instantaneous angular speed or acceleration. By using order spectrum analysis to detect a change in the repetitive intervals between the adjacent encoder pulses, they were able to detect the gearbox faults. The gearbox faults were detected by using peak analysis of the order spectrum. They evaluated their method under constant speed and time-varying speed conditions in the lab. Sharma et al. [24] designed a method in which they used Variational Mode Decomposition (VMD) to demodulate the signal and then detected gear tooth defects in varying speed conditions. In their case study, they used an accelerometer to record vibration signals from the gearbox related to different faults under various speed conditions (such as run-up, random variation, and coast-down). The collected vibration signals were decomposed using VMD (which demodulated the raw vibration signal) and were then analyzed statistically, as well as through signal processing methods. The RMS and kurtosis features were extracted from the demodulated signal and classified into two categories: fault and no-fault. In addition, they proved how well VMD performs in comparison to Flexible Analytic Wavelet

Transform (FAWT) and wavelet-based Empirical Wavelet Transform (EWT). The VMD outperformed EWT and FAWT in both simulation and experimentation. VMD is widely used due to its high adaptability, effectiveness in attenuating mode mixing problems, and low computational time requirements.

An adaptive condition monitoring technique was developed by Inturi et al. [25] for wind turbine gearbox fault prediction, with the goal of predicting the location, type, and severity level of a fault. They used a piezoelectric accelerometer to collect vibration signals from the gearbox, which were then decomposed using the Haar wavelet and the fourth level of decomposition to obtain wavelet coefficients. From the obtained wavelet coefficients, wavelet and statistical features—including RMS, skewness, crest factor, median, etc.—were extracted for training the ML-based model. They compared two feature classification methods, including ML-based multilevel classification and Adaptive Neuro-Fuzzy Inference System (ANFIS)-based multilevel classification. When ML was combined with ANFIS, it performed better, yielding an accuracy score of 92%. The four-level classification included speed stage, component, defect type, and defect severity level. They simulated four types of faults: 25%, 50%, 75%, and 100%. Zhang et al. [26] proposed a new method for detecting fault-related frequencies in vibration signals. In their proposed method, they first used the short-frequency Fourier transform to convert the signal from the time domain to the frequency domain. Initially, the overall frequency information of the vibration signal was collected using a series of frequency window functions. In the end, the signal's fault-related harmonic components were improved using the multi-scale sparse frequency distribution method.

Yu et al. [27] improved the gearbox fault detection process by combining physical models and data-driven modelling techniques. Because simulations cannot accurately predict gearbox defects, they carried out their case study in three stages: physical model construction, data-driven modelling, and virtual real fusion. The signals were initially created through simulation and were also collected using sensors from the gearbox. These generated and collected signals were cross-verified to ensure consistency. If they were consistent, they were passed to the data-driven model. Noise reduction, preprocessing, spectrum analysis, feature extraction, and fault identification were some of the techniques employed during the data-driven modeling phase. They extracted the features, including standard deviation, mean squared error, kurtosis, skewness, RMSF, and RVE, from the signals. After that, both signals were fused together, and feature matching was performed to verify the high-matching correlation coefficients. The matched (cross-verified) signals were then decomposed using VMD based on the signal's matching component. Phase correction was then performed on the matched components, and the envelope spectrum was used to diagnose the defect. The data was classified into four classes: healthy, broken teeth, pitted teeth, and cracked teeth. Yao et al. [28] used signal processing, ML, and optimization techniques to remove noise in vibration signals. The collected data was initially decomposed using the VMD and GWO-WMD methods. The signal was reconstructed using the correlation. Fault features were identified, and the DR-KELM approach was used for classification. Their proposed method produced outstanding results when using GWO-VMD, VMD, DR-KELM, and GWO-VMD with KELM.

Vrba et al. [29] proposed a method for diagnosing gearbox faults based on the Normalized Least Mean Square (NLMS) of adaptive filters in predictive settings. They used the SVM model, which was trained on the features extracted using the envelope approach to predict faults. They used the 10-fold cross-validation technique to verify the performance of their method. The experiments were carried out using a publicly available benchmark dataset collected from both healthy and faulty gearboxes with varied loads and at a constant rotation speed using a gearbox fault diagnosis simulator from SpectraQuest. From the

results, it was proven that the SVM model achieved an accuracy of 90%. Hou et al. [30] designed a unique feature selection method and proposed a multiclass Support Vector Data Description (SVDD) model for diagnosing planetary gearbox defects. The fault-sensitive features were identified by measuring cosine similarity in the kernel space of the Gaussian Radial Basis Function (GRBF). The selected features were then passed to the proposed SVDD model, which classified them into one of five classes: normal, missing tooth, cracked tooth, chipped tooth, and surface tooth. The data used in the experiments was obtained from the Wind Turbine Drivetrain Diagnostics Simulation (WTDDS) and achieved a 100% accuracy score.

Hao et al. [31] addressed some of the issues associated with gearbox defect detection, such as complex signal processing, manual feature extraction, and low accuracy. The experimental setup was made up of a planetary gearbox, motor, hydraulic station, primary control platform, and accelerometer sensor for recording vibration data. The data was collected for four types of faults: sun gears with 31, 30, 18, and 15 broken teeth, respectively. The collected vibration signals were analyzed in both time and frequency domains and then classified using the Deep Belief Network (DBN) model, which yielded an accuracy score of 97%. Meanwhile, Azamfar et al. [32] proposed a method for diagnosing gearbox faults using motor current signature analysis and a two-dimensional Convolutional Neural Network (CNN). They performed experiments using motor current data acquired from an industrial gearbox test rig in varied health conditions and at varying speeds. The data was collected using multiple sensors, and each sensor's frequency spectrum was also obtained. The resulting frequency spectrum from each sensor was then fused together by stacking raw frequency data and passed to the CNN model for fault detection. The data was classified into seven classes: healthy, pitting, eccentricity, missing tooth, chipped tooth, low-severity abrasion, and high-severity abrasion. Their proposed method performed better than other traditional ML classifiers, such as discriminant analysis, KNN, Naïve Bayes, SVM, and Decision Trees.

Shi et al. [33] proposed a Bidirectional Convolutional LSTM (BiConvLSTM) model to diagnose the type, location, and direction of the defect. In their model, they used spatial and temporal features extracted from vibrational and rotational speed measurements. They combined CNN and LSTM because CNN is well known for detecting spatial correlation between two data points with the same timestamp, and LSTM is well known for predicting the temporal dependencies between two nearby timestamps. The data used in their study were collected using three accelerometers and one tachometer. They evaluated the performance of their proposed BiConvLSTM model with the performance of CNN, LSTM, and CNN-LSTM. They evaluated their performance using three types of experiments. Initially, they used a 70:30 ratio, then an 80:20 ratio, and finally a 90:10 ratio to split the data into training and testing subsets. Their proposed model outperformed other models and achieved an accuracy score of 70.83% in the 70:30 ratio split, 79.16% in the 80:20 ratio split, and 84.72% in the 90:10 ratio split. Transfer learning was employed by Jamil et al. [34] to detect gearbox faults in wind turbines from limited datasets with acceptable performance results. In their proposed method, they focused on allowing relevant information to be transferred from the source machine to the target domain while preventing negative transfer. They addressed the problem of negative transfer, because in the changing working environment and conditions, the models are more vulnerable to a negative transfer. Their proposed method outperformed traditional deep learning and deep transfer learning models on two publicly available datasets: the Case Western Reserve University Bearing and Real Field Farm Datasets.

Table 1 summarizes the key findings from the reviewed literature.

Table 1. Key features of the past work.

Refs	Method	No of Classes	Classes	Evaluation Criteria
[23]	Peak analysis of order spectrum	2	Healthy Faulty	Not reported
[24]	RMS and kurtosis-based thresholding	2	Faulty No Fault	Not reported
[25]	Combined ML with ANFIS	4	25% fault 50% fault 75% fault 100% fault	Accuracy = 92%
[27]	Envelope spectrum	4	Healthy Broken teeth Pitted teeth Cracked teeth	Not reported
[28]	SVM	2	Healthy Broken	Accuracy = 90%
[30]	SVDD	5	Normal Missing Crack Chipped Worn	Accuracy = 100%
[31]	Deep belief network (DBN)	4	15 broken teeth 18 broken teeth 30 broken teeth 31 broken teeth	Accuracy = 97%
[33]	Bidirectional convolutional LSTM (BiConvLSTM)	12	Combination of faults- 2 fault types 3 fault location 2 fault direction	Accuracy-1 = 70.83% Accuracy-2 = 79.16% Accuracy-3 = 84.72%

2.3. Comparison Between Deep Learning and Traditional ML

Along with traditional ML techniques, recently employed methodologies have increasingly focused on the usage of deep learning-based methods for gearbox defect detection. Deep learning models, including CNNs and Recurrent Neural Networks (RNNs), can learn hierarchical feature representations from raw vibration signals or time-frequency transformations. This results in high classification performance in complex fault scenarios. CNN-based frameworks are capable of extracting local patterns related to fault signatures without requiring hand-crafted feature design. On the other hand, RNN and LSTM networks can detect temporal dependencies in sequential vibration data. Autoencoders and hybrid deep learning architectures can also be used in unsupervised settings to learn features and detect anomalies. Despite these advantages, deep learning algorithms often require huge amounts of labeled data to avoid overfitting. While deep learning models are computationally intensive, the argument regarding interpretability is being addressed by Explainable AI (XAI) methods. Recent studies have successfully employed SHAP and Grad-CAM to interpret complex models in manufacturing [35], aligning data-driven predictions with physical process knowledge [36].

2.4. Closely Related Past Works

Some of the above-mentioned literature reviews are closely related to our work in terms of dataset, feature extraction, training testing data distribution, and classification. All of the studies discussed above focused on gearbox fault detection by using vibration monitoring.

- The dataset used in our experiments is identical to the one utilized in [28] for their research.
- The feature extraction approach in this research aligns with that of [24,25] because they extracted statistical features for their analysis. Similarly, we also utilized ten statistical features in our study.
- We trained traditional ML classifiers on the training data to predict gearbox faults and evaluated their performance on testing subsets. Similarly, refs. [25,28] employed traditional ML classifiers for gearbox fault prediction.

3. Proposed Framework

This section presents the proposed framework, where Figure 2 shows the overall methodology diagram. Initially, the vibration data from the gearbox was collected using four sensors and was obtained in a dataset from Kaggle [20]. The features were extracted from these raw signals using a moving window-based method, with moving window sizes of 300, 400, 500, 600, 700, and 800 that correspond to raw vibration signal samples. The data in this dataset was collected at a sampling frequency of 20 KHz. This led to window durations of 15 ms for 300 samples, 20 ms for 400 samples, 25 ms for 500 samples, 30 ms for 600 samples, 35 ms for 700 samples, and 40 ms for 800 samples. By setting the moving window size to a specific value, one dataset, which included features extracted from the vibration signals, was created in line with the moving window size. The k-fold cross-validation method was used to partition the resulting datasets, with “k” selected as 5, so that each fold was used only once as a test set, while the remaining folds were used for training, and performance was averaged across all folds.

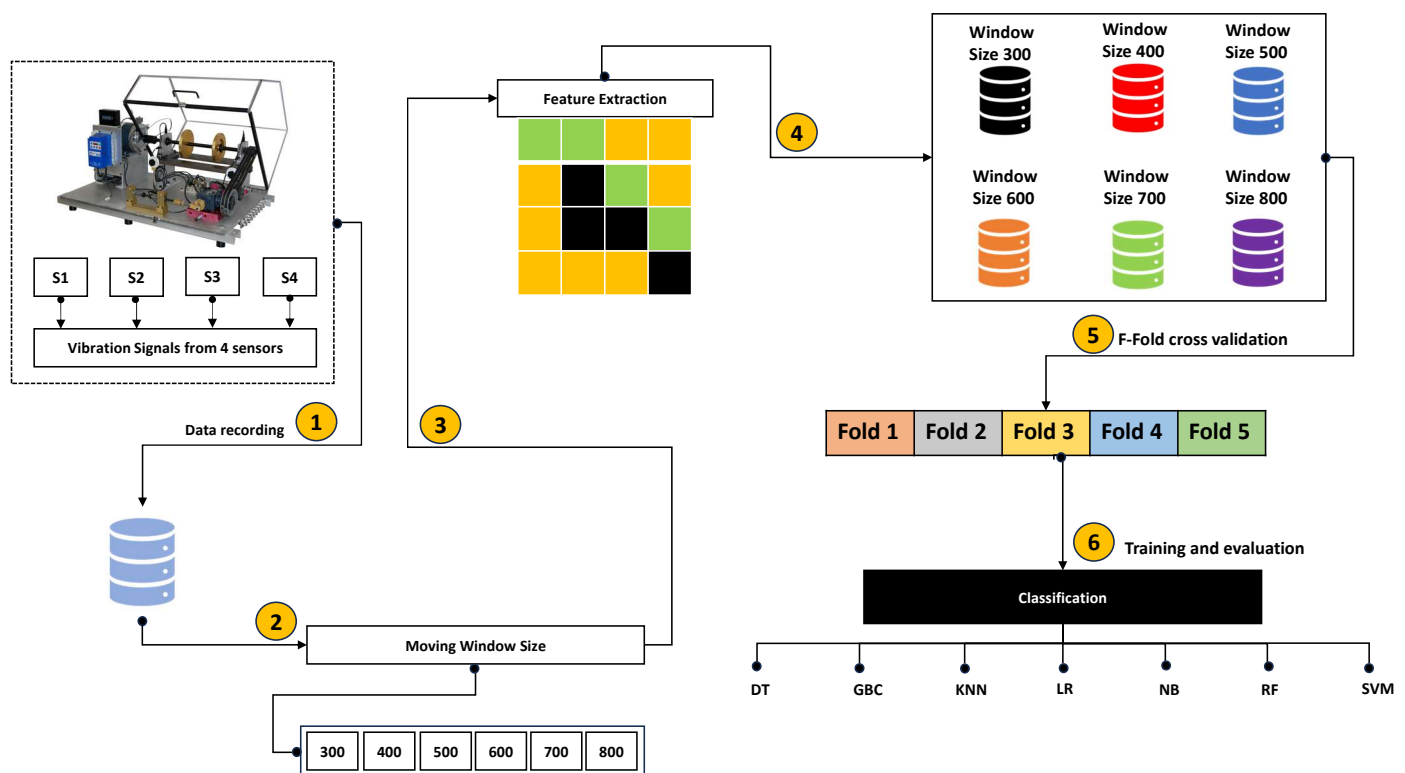


Figure 2. Proposed framework.

3.1. Data Description

The gearbox fault diagnosis dataset [20] was used in this study, which was acquired from Kaggle. The data in the dataset was collected by placing vibration sensors on SpectraQuest's Gearbox Fault Diagnostics simulator with a sampling frequency of 20 KHz, as shown in Figure 3. For measuring vibration signals along four different directions, four vibration sensors were placed in different positions. The data was collected in two different scenarios: healthy and broken tooth conditions (faulty). During the data collection, the load was varied from 0% to 90% with a 10% step size, i.e., 0%, 10%, 20%, and so on. The collected data in terms of file name and each file dimension are summarized in Table 2, where the first number is the number of samples in the file and the second number represents the attributes (in this case, the sensor, and there are a total of four sensors).

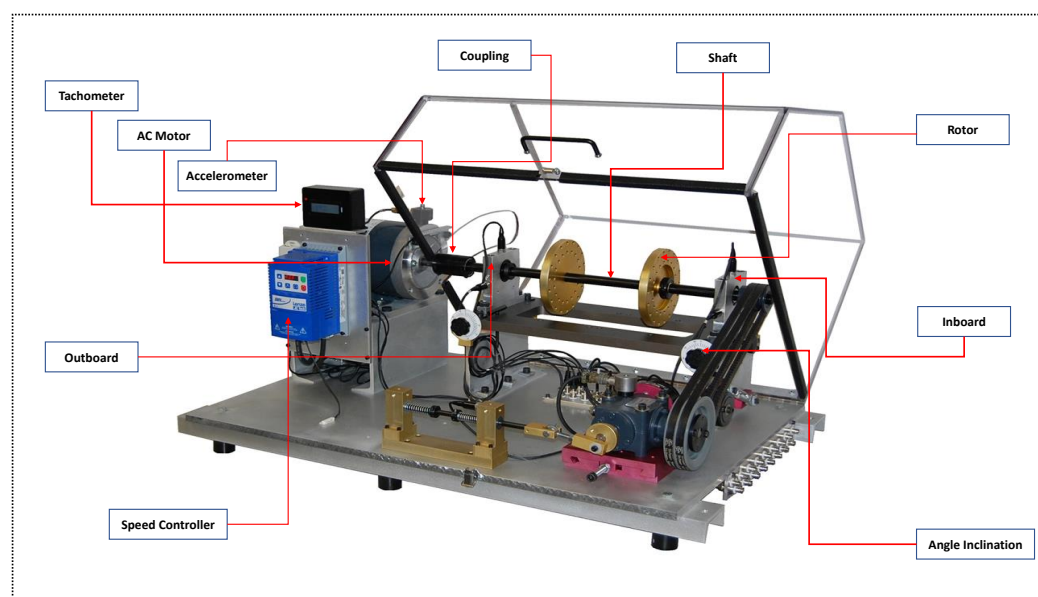


Figure 3. Gearbox prognostic simulator test rig.

Table 2. Dataset summary: file names and dimension for healthy and faulty data.

Healthy		Faulty	
File Name	File Dimension	File Name	File Dimension
h30hz0	(88832,4)	b30hz0	(88320,4)
h30hz10	(92928,4)	b30hz0	(111616,4)
h30hz20	(108544,4)	b30hz0	(114432,4)
h30hz30	(106240,4)	b30hz0	(89856,4)
h30hz40	(100608,4)	b30hz0	(94464,4)
h30hz50	(110848,4)	b30hz0	(94208,4)
h30hz60	(99840,4)	b30hz0	(95488,4)
h30hz70	(101376,4)	b30hz0	(100864,4)
h30hz80	(99840,4)	b30hz0	(110335,4)
h30hz90	(106752,4)	b30hz0	(105728,4)

3.2. Experimental Environment

The experiments in this study were performed on a powerful laptop, the Alienware Core i9 12th generation [37]. The laptop's clock cycle rate was 2.50 GHz, meaning that a single cycle of the processor allowed for 2.50 billion operations per second. The laptop was equipped with an SSD and a 64-bit operating system, as well as Windows 11 version 23H2. The laptop also included an NVIDIA GeForce RTX 3080 Ti Graphic Processing Unit (GPU), which allows faster training of ML models. Python version 3.10.9 was used for

experimentation, which involved everything from preprocessing to training ML models. All programming implementation was carried out in a Jupyter notebook.

3.3. Preprocessing

The preprocessing pipeline employed in this case study is shown in Figure 4. Initially, a single file was picked from the dataset for feature extraction. Features are used to train ML models, which is very helpful when the data is high-dimensional. As mentioned in Section 3.1, the vibration data from the gearbox was acquired using four sensors; therefore, each file includes readings from these four sensors. For feature extraction, we used a moving window-based technique with window sizes of 300, 400, 500, 600, 700, and 800 that correspond to raw vibration signal samples. For feature extraction, the moving window-based approach is helpful because it reduces noise, improves temporal resolution, captures local fluctuations, and allows for short-term patterns that may be missed if the whole signal is considered. The moving window size was varied to capture gradually larger sizes of the vibration signal for feature extraction. In particular, window sizes of 300, 400, 500, 600, 700, and 800 samples were chosen, which correspond to 15 ms, 20 ms, 25 ms, 30 ms, 35 ms, and 40 ms at a sampling frequency of 20 kHz. These window sizes ensure that each window covers a significant portion or several cycles of the shaft rotation, considering that the standard frequency of a rotating shaft in a gearbox is usually between 10 Hz and 50 Hz (periods of 20 ms to 100 ms). Longer windows improve the stability and physical relevance of the extracted statistical features, whereas the shortest window (300 samples, 15 ms) was employed to evaluate model sensitivity with low temporal aggregation. The moving window size was initially set to 300; that allowed for 300 raw vibration signal samples to be read from each sensor, where each window was considered for feature extraction. A total of 40 features were generated by extracting 10 statistical features from each sensor window recording, with 10 features corresponding to each sensor. The extracted features included mean (μ), root mean square (RMS), standard deviation (σ), variance (σ^2), skewness (γ_1), kurtosis (γ_2), peak-to-peak value (P2P), crest factor (CF), shape factor (SF), and impulse factor (IF). As each file has a load value and a class label as well, the extracted 40 features along with the load value and class label were all considered to be a single record in the dataset, corresponding to a moving window size of 300 raw vibration signal samples. This feature extraction process was repeated by taking 300 more raw vibration signal samples until the entire file and dataset were processed. At the end, a dataset corresponding to the moving window size of 300 was formed by combining the extracted features from healthy and faulty vibration signals. For the next iteration of experiments, the moving window's size was set to 400, in which 400 raw vibration signal samples of each sensor were considered for feature extraction at the same time. After that, the moving window's size was set to 500, 600, 700, and 800, and the same experiments were repeated for feature extraction. In all of the experiments, ten features were extracted from each sensor's data, resulting in a total of 40 features, four of which were the same, with each feature corresponding to a particular sensor. The extracted features are defined in the below equations from (1) to (10). These statistical features were retrieved because they are most suited to capturing changes in signal properties caused by mechanical issues such as tooth cracking, gear wear, pitting, and misalignment [38]. The gearbox vibration signal's variability, amplitude, and impulsive characteristics could be captured by such features.

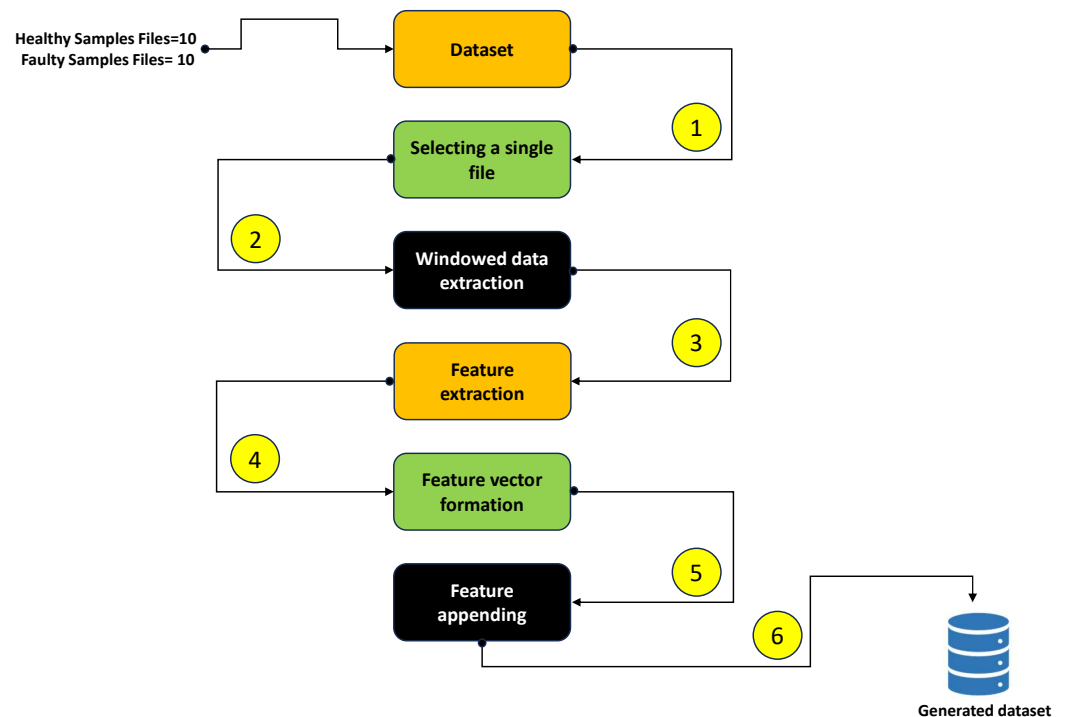


Figure 4. The preprocessing pipeline.

1. **Mean:** The mean is an important statistical feature because of the central tendency, feature normalization, and data imputation [39]. The average value of the signal is represented by the mean, and it provides a measure of central tendency. Apart from that, normalizing or scaling the features to have a mean of zero and a standard deviation of one may improve the performance of ML models, especially those that are sensitive to feature scale, while sometimes, a dataset has possibilities for the missing values for a number of reasons. In this case, the mean can be used to fill in the missing values. In our case, we computed the moving window's mean value and used it as a feature to train ML models. This is because a signal's mean value could indicate the signal's average vibration intensity over time, which is a helpful indicator for steady-state misalignment and load imbalance. Since we have four sensor readings and are processing four moving windows at the same time, we obtained a total of four mean values, one for each sensor moving window.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

2. **RMS:** The RMS value is an important metric that weights the magnitude of an error or fluctuation, giving a higher weight to the higher deviations [40]. As indicated in Equation (2), it is measured by first squaring the signal values, then averaging these squared values, and finally obtaining the square root of the average. The RMS value represents the total vibration energy, which could indicate gearbox fault severity and overall mechanical degradation.

$$\text{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \quad (2)$$

3. **Standard deviation:** The standard deviation is an important measure because it indicates how far a data point deviates from the mean value [41]. A low standard deviation value indicates the data points are closer to the mean, whereas a high

standard deviation value indicates that the data points are spread out across the mean. Understanding the data distribution, consistency, and variability helps for better model development and evaluation. It is computed by taking the squared deviations of the data points from the mean, averaging the squared deviations, and then taking a square root, as indicated by Equation (3). This is because the standard deviation of a signal could indicate amplitude dispersion, and in the case of a gearbox fault, the standard deviation is sensitive to surface degradation and wear.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \quad (3)$$

4. Variance: The variance is another metric that measures the degree to which the data points in a dataset deviate from the mean [42]. The variance is calculated as the average of the squared deviations between each data point and the mean, as indicated in Equation (4). A higher variance value indicates the data points are more widely distributed in the dataset. This is because the variance of a signal shows its energy fluctuation, and in the case of a gearbox fault, it can increase with wear and irregular gear contact.

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \quad (4)$$

5. Skewness: Skewness is another metric for measuring the asymmetry of a data distribution [40]. It can determine whether data points are positively or negatively skewed. Equation (5) can be used to calculate skewness. A positive value of skewness indicates that the tail is longer on the right side; hence, it is right-skewed (positively skewed). The negative value of skewness can indicate negative skewness, which denotes that the tail is longer on the left and thus left-skewed. A balanced distribution has a skewness value of zero, indicating normal distribution. Skewness can be used to demonstrate the asymmetry of signal distribution, which can be used to detect localized faults causing uneven impacts in gearbox vibration signals.

$$\gamma_1 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \right)^{3/2}} \quad (5)$$

6. Kurtosis: The kurtosis measures the sharpness of a probability distribution's peak [43]. It is useful in determining the variation caused by the extreme values (outliers) in the dataset. A high kurtosis value indicates that the distribution is highly tailed, which means there will be more outliers. On the other hand, a low value of kurtosis could indicate that the distribution has light tails and, as a result, will have fewer outliers. Similarly, a normal kurtosis value may indicate that the distribution is normal (for example, Gaussian distribution). The kurtosis value can be computed via Equation (6). It describes impulsiveness and sharp transients in signals, which can detect highly sensitive indications of early-stage problems in gearbox vibration signals.

$$\gamma_2 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \right)^2} \quad (6)$$

7. P2P: The P2P value measures the difference between the maximum and minimum values in the signal [44]. In our case, we measured the P2P value of each moving window. It is useful in capturing the variability, detecting outliers, and performing

normalization. Equation (7) is used to measure the P2P value. In this research, it can be used to indicate the range of vibration extremes, which is useful for detecting abnormal gear meshing and backlash in gearbox vibration signals.

$$P2P = x_{\max} - x_{\min} \quad (7)$$

8. CF: The ratio of a signal's peak value to its RMS value is known as the CF [45]. The CF measures the correlation between the peaks and the signal's total energy. Equation (8) can be used to calculate the CF. The CF in gearbox vibration signals highlights the impact severity regardless of signal energy.

$$CF = \frac{x_{\max}}{\text{RMS}} \quad (8)$$

9. SF: The SF can be used to characterize signal distribution. It is defined as the ratio of the RMS value to the mean absolute value [46] and can be calculated with Equation (9).

$$SF = \frac{\text{RMS}}{\text{mean}} \quad (9)$$

10. IF: The IF can be used to detect sharp peaks in a signal. It is defined as the ratio of the maximum absolute value of the signal to its mean absolute value [45]. It is highly useful for fault detection, signal processing, and vibration analysis. The IF can be calculated via Equation (10).

$$IF = \frac{x_{\max}}{\text{mean}} \quad (10)$$

3.4. Data Distribution

The number of samples generated in each dataset is dependent on the window size, because different sizes of moving windows are used for feature extraction. The dataset corresponding to moving window size 300 has 6676 samples, with 3328 healthy and 3348 faulty samples. The dataset with a moving window size of 400 includes 5006 samples, 2496 of which are healthy and 2510 of which are faulty. Similarly, the dataset with a moving window size of 500 comprises 3998 samples, with 1994 healthy samples and 2004 faulty samples. As seen in Figure 5, increasing the size of the moving window reduces the number of samples in the corresponding dataset. This is because increasing the window size for feature extraction considers more raw data points for feature extraction. Therefore, fewer moves are required to traverse the sensor's complete readings, where features are extracted at each move.

An imbalanced dataset can also affect the ML model's performance. An imbalanced dataset is a dataset that has fewer samples from a few classes and more samples from other classes, causing the resulting model to fail to generalize well. The imbalanced dataset-based trained model could face several challenges, such as bias toward the majority class, poor generalization, and misleading accuracy [47]. Several data-balancing techniques, including undersampling, oversampling, and SMOTE, can be used to address the problem of an imbalanced dataset [48]. Apart from data balancing, evaluation metrics other than accuracy, such as precision, recall, F1 score, and the AUC-ROC curve, can be used to evaluate the performance of a model trained on an imbalanced dataset. Therefore, we checked whether the generated datasets corresponding to the moving window sizes of 300, 400, 500, 600, 700, and 800 were balanced or not. Figure 5 indicates that these datasets were balanced.

To ensure a strong and unbiased evaluation of the models, five-fold cross-validation ($K = 5$) was used instead of traditional predefined train–test splits (e.g., 90:10, 80:20, or 70:30 ratio splits). Fixed train–test splits were unfeasible due to the small test sets that can result

from making use of larger window sizes (15–40 ms), which substantially reduced the total number of samples in each dataset. In five-fold cross-validation, each dataset was divided into five folds; in each iteration, one fold acted as the test set, while the remaining four folds were used for training. This technique was carried out until each fold had been tested once, after which performance metrics were averaged across all folds. This method ensured that all samples were analyzed, avoided temporal data leakage, and provided more reliable estimates of model generalization than traditional ratio-based splits.

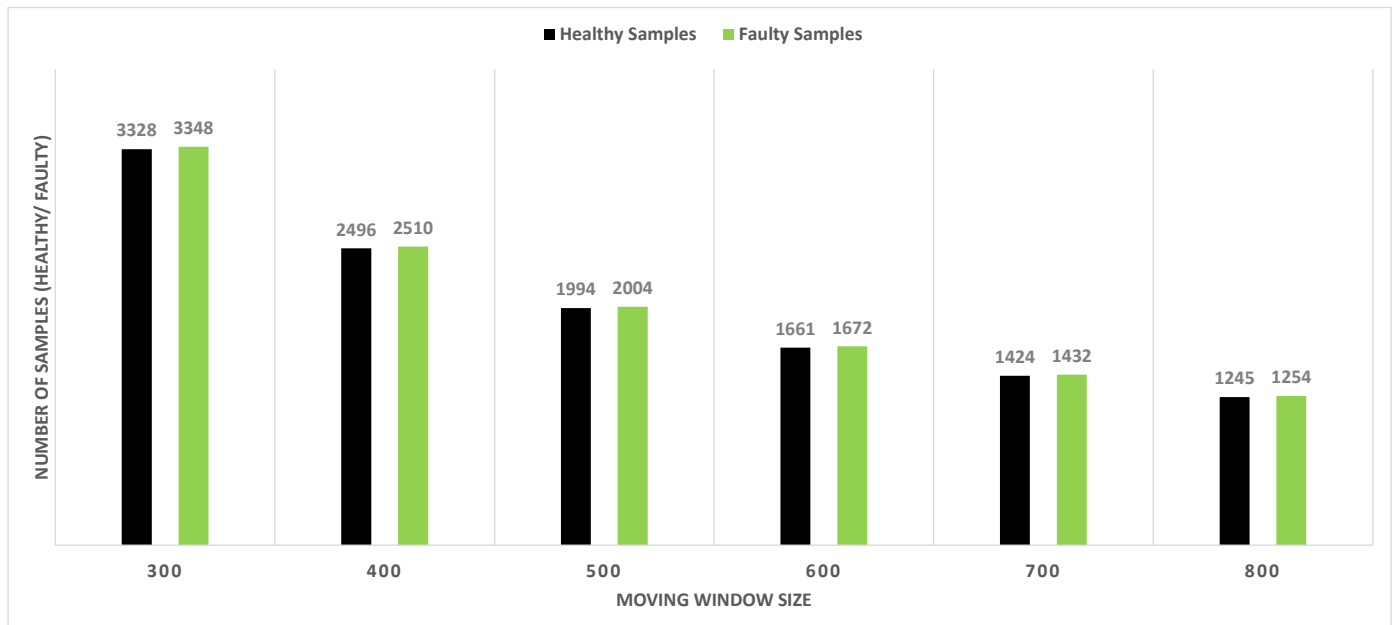


Figure 5. Distribution of healthy and faulty samples in the generated datasets corresponding to the moving window size.

3.5. Classification Models

3.5.1. Decision Tree

A Decision Tree is a supervised ML classifier used for solving classification problems. It has a flowchart-like design that includes concepts such as root node, internal nodes, leaf nodes, splitting criteria, maximum depth, and pruning [49]. The dataset is divided at each node according to the attribute values that best split the dataset. The root node (topmost node) in the tree splits the entire dataset based on the attribute that best splits the dataset on the selected criterion. Variance reduction, entropy, and Gini impurity are possible splitting criteria [50]. The level of impurity used in classification tasks is measured by the Gini impurity. A dataset is considered more pure if its Gini impurity score is low, showing that it mostly consists of a single class. On the other hand, entropy is used as a splitting criterion in information gain-based Decision Trees (such as ID3). Entropy measures the degree of uncertainty of a dataset and decreases when the classes are separated. Variance reduction can be used with Decision Trees while solving a regression problem. It minimizes the variance between nodes by selecting splits, which result in more homogeneous groups. The max depth specifies the maximum number of splits or levels in a tree. A tree can be overfitted with maximum depth if it is complicated and only memorizes the training data. Pruning is one approach to dealing with overfitting in Decision Trees, which removes some of the nodes or subtrees in a tree.

3.5.2. SVM

SVM is a supervised ML classifier that is commonly used for solving classification problems. It is efficient and robust for high-dimensional data as well as for linear and non-

linear classification problems [51]. The hyperplane, support vectors, margin, and kernel are some of the key concepts of SVM [52]. SVM works by creating a hyperplane between data points that best separates them into the classes that they belong to. In 2D, the hyperplane is a line, whereas in 3D, it is a plane and a high-dimensional boundary for larger feature spaces. The best hyperplane is the one that maximizes the distance between the closest data point in each class and itself. The support vectors are the data points nearest to the hyperplane. These points are very helpful when defining the margin as well as the hyperplane, which is why they are so important. The margin is the distance between the hyperplane and the nearest data points of any class. SVM tries to improve this margin, thereby increasing the distance between the class data points and the hyperplane, which improves the classification accuracy. When the data is non-linearly separable, the SVM maps the input data into a higher-dimensional space using a kernel function [53], allowing for a linear separation.

3.5.3. Random Forest

Random Forest is a supervised ML classifier that can solve classification problems. It is an ensemble learning technique, as it combines multiple Decision Trees to improve results and creates a model that performs better than a single Decision Tree model [54]. Random Forest uses a subset of the training data to train the Decision Tree model, which is selected with replacement. This ensures that each tree is trained on a slightly different dataset, thus reducing overfitting. The final classification is determined by majority voting. It combines the predictions of each tree in the forest and then uses majority voting to obtain the final outcome [55].

3.5.4. KNN

KNN is a non-parametric approach that can be applied to both regression and classification problems. KNN predicts the class of a new data point by counting the labels of the k closest data points in the training dataset. The key parameter in the KNN is ' k ', which defines the number of nearest neighbors to be considered. A smaller value of ' k ' may make the model more vulnerable to noise. However, a high value of ' k ' can smooth the predictions, possibly blurring them. It assigns the class label based on a majority vote of the ' k ' nearest neighbors [56].

3.5.5. Logistic Regression

Logistic Regression is a popular supervised learning algorithm that works as a classifier and is best suited to binary classification tasks. Apart from its name, it predicts categorical outputs rather than continuous values. Logistic Regression uses the sigmoid function for mapping predictions to a probability score between 0 and 1 [56]. It uses the binary cross-entropy or log loss function, which is refined throughout training. The goal is to find the weights that minimize the average log loss across all instances.

3.5.6. Naïve Bayes

Naïve Bayes is a probabilistic algorithm used to solve problems in supervised learning settings. It makes naïve assumptions and applies Bayes' theorem to each pair of features with respect to the target class. Naïve Bayes states that all the features are conditionally independent of one another [57]. Naïve Bayes uses Bayes' theorem to find the probability of the target class based on prior knowledge of conditions, which can be useful when related to that class.

3.5.7. GBC

GBC combines many weak classifiers (mostly Decision Trees) to create a strong classifier through ensemble learning [58]. In GBC, each classifier focuses on the previous classifier's errors and updates its weights accordingly.

3.6. Evaluation Metrics

The proposed solution was evaluated based on accuracy, precision, and recall. The F1 score was not considered, because it is the harmonic mean of precision and recall, and we already considered precision and recall for our comparison. In Equations (11)–(13), the True Positives (TPs) are samples that the model correctly predicts as positive samples, whereas True Negatives (TNs) are samples that are correctly predicted as negative samples by the model. On the other hand, False Positives (FPs) are samples incorrectly predicted by the model as positive samples that are actually negative samples. Similarly, False Negatives (FNs) are samples that the model incorrectly predicted as negative but are actually positive samples.

3.6.1. Accuracy

Accuracy is a metric used for evaluating the performance of an ML classifier. It calculates the percentage of correctly classified samples with respect to the total samples and can be calculated using the equation given below (11).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (11)$$

3.6.2. Precision

Precision measures the ratio of actually positive samples to the positive predictions made by the model, as computed using Equation (12):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (12)$$

3.6.3. Recall

Recall is a measure of how many actual positive samples the model accurately predicts as positive. Recall can be calculated using the equation given below (13).

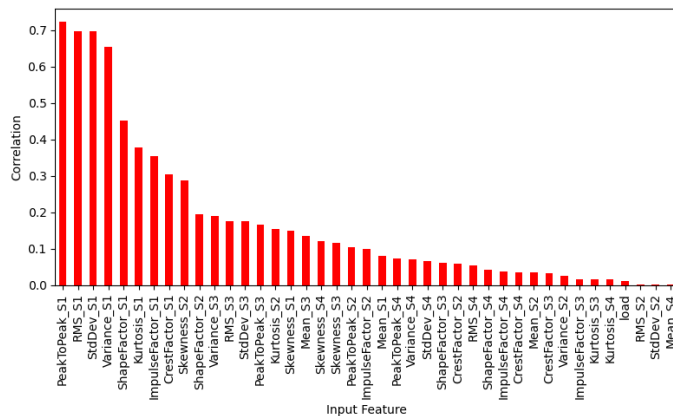
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (13)$$

4. Results

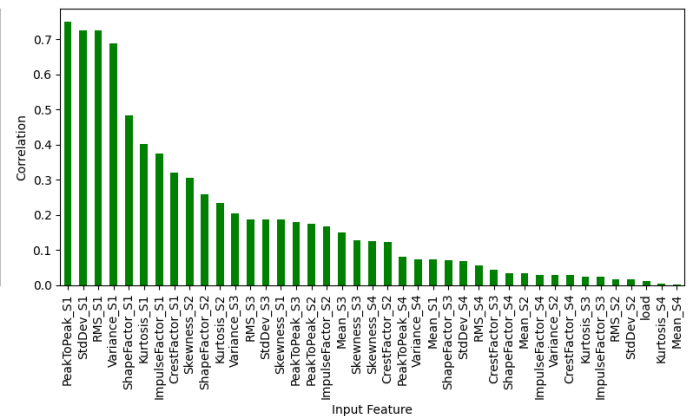
4.1. Exploratory Data Analysis

The exploratory data analysis of the datasets obtained using moving window sizes of 300, 400, 500, 600, 700, and 800 from the gearbox fault dataset showed a few interesting facts about the data. The correlation technique was employed to identify the relationships among the attributes (or features) in the attribute (feature) set that can influence whether this feature vector is healthy or faulty. Figure 6 presents the correlation values of features sorted in descending order in each dataset corresponding to the moving window size. The bar height represents the correlation value of that feature that contributes to the class label. The correlation values of the dataset features, calculated using moving window sizes from 300 to 800 (15 ms to 40 ms), are presented in Figure 6a–f, with each bar representing a feature's correlation value. The features peak-to-peak, RMS, standard deviation, variance, shape factor, kurtosis, and impulse factor—all computed from sensor 1 (designated as

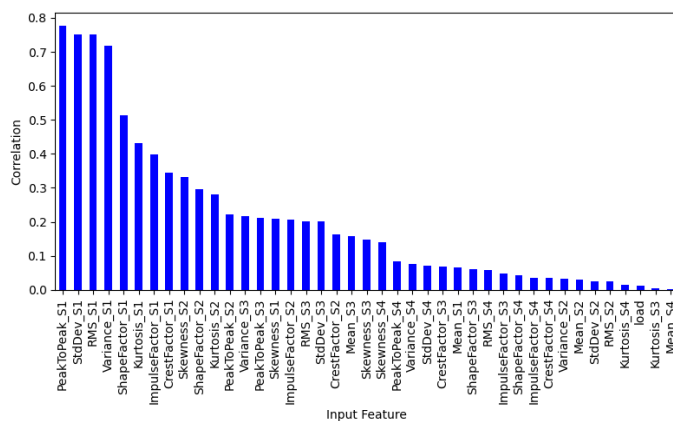
_S1)—have consistently high correlations across all datasets, whereas the correlation values of the remaining features can be observed from Figure 6a–f.



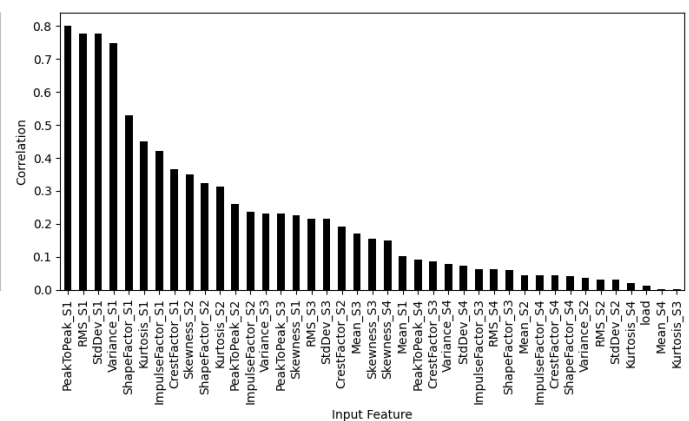
(a) Window size 300



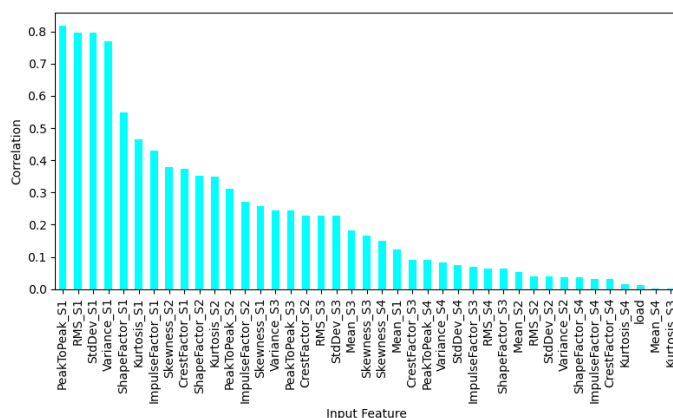
(b) Window size 400



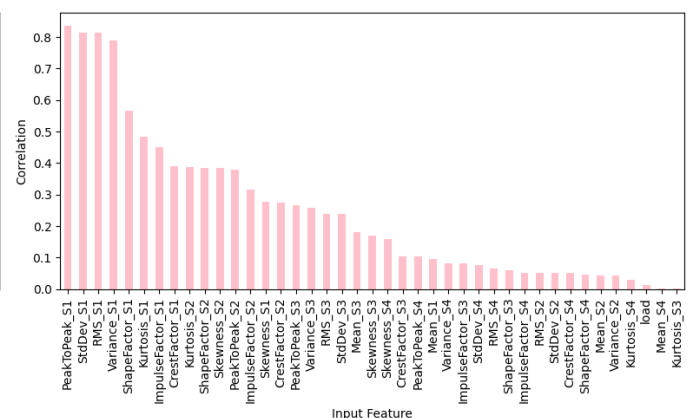
(c) Window size 500



(d) Window size 600



(e) Window size 700



(f) Window size 800

Figure 6. A plot showing the correlation histogram of several input features and the target “label” variable.

4.2. Confusion Matrix

The number of True Positives (TPs), True Negatives (TNs), False Positives (FPs), and False Negatives (FNs) is a reliable indicator of how well a method performs across different evaluation metrics. Although the percentage is presented via accuracy, precision,

and recall, the raw numbers are as interesting to examine. Table 3 presents the number of TPs, TNs, FPs, and FNs.

Table 3. The table displays the number of True Positives (TPs—healthy samples identified as healthy samples), False Positives (FPs—faulty samples identified as healthy samples), False Negatives (FNs—healthy samples identified as faulty), and True Negatives (TNs—faulty samples identified as faulty samples).

Classifier	Window 300				Window 400			
	TPs	FPs	FNs	TNs	TPs	FPs	FNs	TNs
Random Forest	3305	39	23	3309	2485	21	11	2489
Decision Tree	3255	113	73	3235	2433	68	63	2442
Naïve Bayes	2776	311	552	3037	2191	183	305	2327
SVM	3309	14	19	3334	2491	5	5	2505
GBC	3295	22	33	3326	2482	11	14	2499
KNN	3242	49	86	3299	2447	13	49	2497
Logistic Regression	3305	35	23	3313	2489	10	7	2500

Classifier	Window 500				Window 600			
	TP	FP	FN	TN	TP	FP	FN	TN
Random Forest	1985	9	9	1995	1657	4	4	1668
Decision tree	1952	44	42	1960	1628	31	33	1641
Naïve Bayes	1800	111	194	1893	1548	65	113	1607
SVM	1990	4	4	2000	1661	0	0	1672
GBC	1982	5	12	1999	1655	4	6	1668
KNN	1965	10	29	1994	1638	2	23	1670
Logistic Regression	1992	2	2	2002	1660	1	1	1671

Classifier	Window 700				Window 800			
	TP	FP	FN	TN	TP	FP	FN	TN
Random Forest	1421	4	3	1428	1242	6	3	1248
Decision Tree	1403	16	21	1416	1230	25	15	1229
Naïve Bayes	1361	45	63	1387	1215	27	30	1227
SVM	1423	0	1	1432	1245	1	0	1253
GBC	1419	4	5	1428	1238	6	7	1248
KNN	1407	4	17	1428	1239	0	6	1254
Logistic Regression	1424	0	0	1432	1245	0	0	1254

4.3. Accuracy

Table 4 presents the accuracy scores of the classifiers that were trained using five-fold cross validation on datasets corresponding to moving window sizes of 300, 400, 500, 600, 700, and 800. At a moving window size of 300 (equivalent to a 15 ms window), the SVM outperformed all other classifiers with an accuracy score of 99.51%. At the same time, the GBC, Logistic Regression, and Random Forest achieved accuracy scores of 99.18%, 99.13%, and 99.07%, respectively, outperforming the Decision Tree, Naïve Bayes, and KNN. The Naïve Bayes classifier performed poorly as compared to other classifiers, with an accuracy score of 87.07%. When the moving window size of 400 (equal to 20 ms) dataset was used, all of the classifier's accuracy scores improved by only a slight margin. The SVM outperformed all other classifiers with an accuracy score of 99.80%, which was improved from an earlier score of 99.51%. The Logistic Regression, GBC, and Random Forest classifiers also improved their accuracy scores to 99.66%, 99.50%, and 99.36%, up from 99.13%, 99.18%, and 99.07%, respectively. On the other hand, classifiers including KNN, Decision Tree, and Naïve Bayes, which scored poorly compared to all other classifiers,

improved their accuracy scores to 98.76%, 97.38%, and 90.25% from 97.98%, 91.21%, and 87.07%, respectively.

Table 4. ML classifiers' accuracy results on a varied-size moving window dataset.

Window Size	300	400	500	600	700	800
Classifier						
Random Forest	99.07%	99.36%	99.55%	99.76%	99.75%	99.64%
Decision Tree	97.21%	97.38%	97.85%	98.08%	98.70%	98.40%
Naïve Bayes	87.07%	90.25%	92.37%	94.66%	96.22%	97.72%
SVM	99.51%	99.80%	99.80%	100%	99.96%	99.96%
GBC	99.18%	99.50%	99.57%	99.70%	99.68%	99.48%
KNN	97.98%	98.76%	99.02%	99.25%	99.26%	99.76%
Logistic Regression	99.13%	99.66%	99.90%	99.94%	100%	100%

The same trend of improved classifier accuracy was observed when the corresponding dataset with a moving window size of 500 (equivalent to 25 ms) was employed. The highest accuracy score of 99.90% was achieved by Logistic Regression, which is higher than the prior score of 99.66%. This time, the SVM, which previously achieved the highest accuracy score of 99.80%, continued to perform similarly. The GBC, Random Forest, and KNN also improved their accuracy scores from 99.50%, 99.36%, and 99.66% to 99.57%, 99.55%, and 99.02%, respectively, surpassing Decision Tree and Naïve Bayes. On the other hand, the poorly performing Naïve Bayes and Decision Tree algorithms also improved their accuracy scores from 90.25% and 97.38% to 92.37% and 97.85%, respectively. Following that, when the moving window size of 600 (which is equal to 30 ms) dataset was used, the SVM achieved a higher accuracy score of 100%, an improvement over its previous score of 99.80%, whereas Logistic Regression, Random Forest, and GBC improved their accuracy scores to 99.94%, 99.76%, and 99.70%, respectively, from their previous accuracy scores of 99.90%, 99.50%, and 99.57%. Similar to this, KNN and Decision Tree improved their results from 99.02% and 97.85% to 99.25% and 98.08%, respectively, outperforming Naïve Bayes. The Naïve Bayes classifier, which scored poorly with an accuracy score of 94.66%, improved from its previous accuracy score of 92.37% upon increasing the size of the moving window from 500 to 600 (25 ms to 30 ms).

However, when the moving window size was extended from 600 (30 ms) to 700 (35 ms) and 800 (40 ms), the accuracy scores of some classifiers improved, while those of others marginally reduced. Out of all the classifiers, Logistic Regression improved its accuracy score from 99.94% to 100% on moving window sizes of 700 and 800. On moving window sizes of 700 (35 ms) and 800 (40 ms), the Naïve Bayes accuracy score increased from 94.66% to 96.22% and 97.72%, respectively, despite its poor performance. On window sizes of 700 and 800, the SVM's accuracy score dropped from 100% to 99.96%. Upon increasing the window size to 700 and 800, Decision Tree, Random Forest, and GBC also showed slightly reduced accuracy scores.

4.4. Precision

Table 5 presents the precision scores resulting from Random Forest, Decision Tree, Naïve Bayes, SVM, Gradient Boosting Classifier, KNN, and Logistic Regression for moving window sizes of 300 to 800. This table demonstrates that increasing the moving window size used for feature extraction could improve the precision score of different classifiers. Initially, when the classifiers were trained on the moving window size of 300 (corresponding to 15 ms), SVM achieved a precision score of 99.51%, outperforming all other classifiers at the time. At that time, GBC, Logistic Regression, and Random Forest outperformed Decision

Trees, Naïve Bayes, and KNN, achieving precision scores of 99.18%, 99.13%, and 99.07%, respectively. With precision scores of 97.99% and 97.22%, respectively, KNN and Decision Tree performed better than the Naïve Bayes score of 87.29%. However, all of the classifiers' precision scores increased when the moving window size increased from 300 (15 ms) to 400 (ms). The SVM achieved the highest precision score of 99.80% on the moving window size of 400 (corresponding to 20 ms), which was an improvement over the prior precision score of 99.51%. However, at the same time, the precision scores of Logistic Regression, GBC, and Random Forest also increased from 99.13%, 99.18%, and 99.07% to 99.66%, 99.50%, and 99.36%, respectively. The precision scores of Decision Tree and KNN similarly increased from 97.99% and 97.22% to 98.77% and 97.39%, respectively, when the moving window size increased from 300 to 400. From 87.29% to 90.35%, the lowest-performing classifier, Naïve Bayes, also showed an increased precision score.

Table 5. ML classifiers' precision results on a varied-size moving window dataset.

Window Size	300	400	500	600	700	800
Classifier						
Random Forest	99.07%	99.36%	99.55%	99.76%	99.76%	99.64%
Decision Tree	97.22%	97.39%	97.85%	98.09%	98.71%	98.41%
Naïve Bayes	87.29%	90.35%	92.47%	94.71%	96.23%	97.72%
SVM	99.51%	99.80%	99.80%	100%	99.97%	99.96%
GBC	99.18%	99.50%	99.58%	99.70%	99.69%	99.48%
KNN	97.99%	98.77%	99.03%	99.26%	99.27%	99.76%
Logistic Regression	99.13%	99.66%	99.90%	99.94%	100%	100%

Following that, all of the classifiers showed an improvement in their precision scores when the moving window size of 500 (which is equivalent to 25 ms) was taken into consideration. With precision scores of 99.90% and 99.80%, respectively, Logistic Regression and SVM outperformed all other classifiers. As compared to the previous precision scores of 99.50%, 98.77%, and 99.36% obtained on moving window of size 400, GBC, Random Forest, and KNN achieved precision scores of 99.58%, 99.55%, and 99.03%, respectively. Although the Naïve Bayes and Decision Tree classifiers had the lowest precision scores, they also improved as the moving window size increased, going from 97.39% and 90.35% to 97.85% and 92.47%, respectively. Following that, all of the classifiers showed the same pattern of slightly increasing their precision scores when the moving window size 600 (corresponding to 30 ms) dataset was considered. SVM and Logistic Regression improved their precision scores slightly from 99.80% and 99.90% to 100% and 99.94%, respectively, on the moving window size of 600. With these scores, SVM and Logistic Regression performed better than other classifiers. However, Random Forest, GBC, and KNN outperformed Decision Trees and Naïve Bayes, improving their precision scores to 99.76%, 99.70%, and 99.26% from 99.55%, 99.58%, and 99.03%, respectively. On the other hand, Decision Tree and Naïve Bayes, which had the lowest precision scores of 98.09% and 94.71%, also improved their precision from 97.85% and 92.47%, respectively.

Furthermore, it was found that some classifiers had improved precision scores, while others had worse scores when the moving window size increased to 700 (equivalent to 35 ms) and 800 (equivalent to 40 ms). Logistic Regression improved from its prior score of 99.94% on window size 600 to a precision score of 100% on moving window sizes 700 and 800. Meanwhile, for window sizes of 700 and 800, SVM's precision scores slightly decreased to 99.97% and 99.96%, respectively, from its previous 100% precision scores. KNN's score improved from 99.26% to 99.27% and 99.76%, whereas Naïve Bayes' scores improved from

94.71% to 96.23% and 97.72%, respectively. The GBC precision score dropped from 99.70% to 99.69% and 99.48% on moving window sizes 700 and 800, respectively.

4.5. Recall

The recall scores of classifiers trained on datasets corresponding to moving window sizes 300, 400, 500, 600, 700, and 800 partitioned using k-fold cross-validation are presented in Table 6 below. With a recall score of 99.51%, the SVM outperformed all other classifiers, whereas at the same time, GBC, Logistic Regression, and Random Forest outperformed Decision Tree, Naïve Bayes, and KNN by recall scores of 99.18%, 99.13%, and 99.70%. KNN and Decision Tree surpassed Naïve Bayes, which has a recall score of 87.07%, with recall scores of 97.98% and 97.21%, respectively. Following that, all of the classifiers' recall scores improved when the corresponding dataset with a moving window size of 400 was used. With a recall score of 99.80%, SVM outperformed all other classifiers. This score was an improvement on the prior recall score of 99.51% on the dataset corresponding to a moving window size of 300. On the other hand, with recall scores of 99.66%, 99.50%, and 99.36%, respectively, Logistic Regression, GBC, and Random Forest performed better than Decision Tree, KNN, and Naïve Bayes. These scores also improved from prior recall scores obtained on the moving window size 300 dataset, with Logistic Regression improving from 99.13% to 99.66%, GBC improving from 99.18% to 99.50%, and Random Forest improving from 99.07% to 99.36%. Compared to this, KNN and Decision Trees outperformed Naïve Bayes, which has a recall score of 90.25%, with recall scores of 98.76% and 97.38%, respectively. In addition, these classifiers outscored previous recall scores, with KNN improving from 97.98% to 98.76%, Decision Tree improving from 97.21% to 97.38%, and Naïve Bayes improving from 87.07% to 90.25%.

Table 6. ML classifiers' recall results on a varied-size moving window dataset.

Window Size	300	400	500	600	700	800
Classifier						
Random Forest	99.07%	99.36%	99.55%	99.76%	99.75%	99.64%
Decision Tree	97.21%	97.38%	97.85%	98.08%	98.70%	98.40%
Naïve Bayes	87.07%	90.25%	92.37%	94.66%	96.22%	97.72%
SVM	99.51%	99.80%	99.80%	100%	99.96%	99.96%
GBC	99.18%	99.50%	99.57%	99.70%	99.68%	99.48%
KNN	97.98%	98.76%	99.02%	99.25%	99.26%	99.76%
Logistic Regression	99.13%	99.66%	99.90%	99.94%	100%	100%

The same pattern of improving recall score with moving window size was observed when considering moving window sizes of 500 and 600. At the moving window size of 500, Logistic Regression outperformed other classifiers, achieving a recall score of 99.90%, which was also higher than the prior recall score of 99.66% obtained on the dataset corresponding to a moving window size of 400. On the other hand, Naïve Bayes, which had the lowest recall score of 92.37% on the moving window size 500-associated dataset, improved its recall score from 90.25% on the moving window size 400-corresponding dataset. On the other hand, SVM outperformed all other classifiers and achieved a 100% recall score on the dataset corresponding to a moving window size of 600, whereas SVM previously achieved a recall score of 99.80% on the dataset corresponding to moving window size 500, which improved in that case. However, with a recall score of 94.66%, Naïve Bayes performed poorly compared to other classifiers on the dataset regarding a moving window size of 600. However, Naïve Bayes also improved its recall score from 92.37% on the dataset corresponding to a moving window size of 500, similar to other classifiers.

On the other hand, some classifiers showed a slight decrease in recall scores on the moving window sizes of 700 and 800, while others showed improvements. On the corresponding datasets with moving window sizes of 700 and 800, the Logistic Regression classifier achieved 100% recall scores. Compared to the prior recall score of 99.94% on the dataset corresponding to moving window size 600, this recall score was improved, whereas the SVM's recall score dropped to 99.96% on moving window sizes of 700 and 800 from 100% on the dataset corresponding to a moving window size of 600. Similarly, the GBC recall score decreased from 99.70% on moving window sizes of 600 to 99.68% and then 99.48% on moving window sizes of 700 and 800, respectively. Similarly, the Random Forest recall score dropped from 99.76% on moving window sizes of 600 to 99.75% then 99.64% on moving window sizes of 700 and 800, respectively. A pattern of increasing recall scores with increasing moving window size has been shown by KNN and Naïve Bayes.

5. Discussion

Signals obtained from gearboxes in real-world settings can often be polluted by operations and noise outside from surrounding machinery. The statistical features extracted from moving windows could provide robustness to high-frequency noise by averaging temporal variations. Another important feature of industrial environments is load variation, because changes in load can have an effect on vibration amplitude and frequency.

This study focuses on time-domain statistical features extracted using window lengths corresponding to sub-rotational, single-rotational, and multi-cycle rotational intervals, despite the fact that frequency-domain and time-frequency domain features are frequently used in vibration-based defect detection. In particular, the window sizes of 400 samples (one full rotation), 500–800 samples (covering prolonged rotational intervals, including up to two full cycles), and 300 samples (a little less than one full rotation) are taken under consideration. Prior comparative analyses have shown that across such window sizes, time-domain statistical features perform similarly to or better than frequency-domain and fused feature representations for several machine learning classifiers. In addition, frequency-domain features do not always improve performance and, in some cases, they result in reduced classification performance. As a result, time-domain features were chosen in this study to ensure methodological clarity, computing efficiency, and reliable fault discrimination.

In this research, the dataset was initially processed for feature extraction, which involved extracting statistical features from the dataset using a moving window-based approach. The moving window-based approach uses a fixed-size window to traverse the signal at each step of the analysis [59]. The dataset employed for this study was collected under a variety of load conditions ranging from 0% to 100%, with a step size of 10%, which corresponds to realistic operational scenarios commonly observed in industrial systems. The size of the moving window used for feature extraction was determined by the gearbox's physical properties and the shaft's rotational frequency. In an average gearbox system, the shaft rotational frequency ranges between 10 and 50 Hz, corresponding to rotational periods of 20 and 100 ms. To ensure that the extracted features could capture both partial and complete rotational information, multiple window sizes were attempted. Window sizes of 300, 400, 500, 600, 700, and 800 were evaluated, with time durations ranging from about 15 to 40 ms. A window size of 300 could indicate a duration shorter than one complete shaft rotation, but it was deliberately included to look into the behavior of the feature extraction method during partial cycles. This allows for a sensitivity analysis of the proposed framework to sub-cycle information, which could provide information regarding the effects of incomplete rotational coverage on feature stability and fault discriminability. A window size of 400 samples and above could span at least one fundamental rotating

phase. A window size of 400 samples could include at least one rotational cycle, whereas a window size of 800 could capture two rotational cycles, which enhances feature robustness at the cost of temporal resolution; however, a window size of 400 to 800 could capture a rotational cycle ranging from single to double. A balanced trade-off between rotational completeness and temporal resolution can be achieved by evaluating window lengths of one, two, and three cycles.

This manuscript employs several ML classifiers evaluated using five-fold cross validation to systematically analyze the effect of the window size used for feature extraction on classifier performance. The results, which are presented in terms of accuracy, precision, and recall, show that the temporal length of the input window significantly impacts classifier performance. When the moving window size was increased from 300 samples (equivalent to 15 ms) to 600 samples (equivalent to 30 ms), performance for a significant number of classifiers progressively improved. This pattern was especially prominent for linear and probabilistic models such as Logistic Regression and Naïve Bayes. Due to incomplete rotational information, Naïve Bayes achieved an accuracy, precision, and recall of about 87% at a 300 moving window, demonstrating poor class separability. As the window size increased to 600–800 samples, performance improved significantly, reaching 97.7%, indicating that larger windows generate more stable statistical feature distributions, which are required for probabilistic classifiers that assume conditional independence. This behavior demonstrates that sub-cycle windows fail to capture enough rotational dynamics for accurate probability calculation.

With accuracies over 97% at 300 samples, tree-based models such as Decision Trees, Random Forests, and GBC performed well even at smaller window sizes. Their ability to accommodate noisy or partially informative inputs and model non-linear interactions between features is responsible for their stability. However, as the window size increased, their performance continued to improve, reaching its highest point between moving window sizes 600 and 700. The marginal gains in performance saturated or slightly declined beyond this range, especially for GBC, indicating that excessively large windows could create duplicate information and reduce sensitivity to localized fault signals. With 100% accuracy, precision, and recall at a window size of 600, SVM consistently obtained the best performance over all window sizes. When the window size corresponded with at least one full shaft rotation, the feature space became significantly separable, showing the robust margin maximization capability of SVM. Beyond the a moving window size of 600, performance stabilization suggests that this temporal range already contains the discriminative data required for the most accurate classification. With increasing window size, the KNN classifier showed an average improvement in performance, achieving almost 98% accuracy at a moving window size of 300 and approximately 99.8% at a moving window size of 800. Larger window sizes can generate more smooth and consistent feature representations, thus improving neighborhood consistency. This pattern shows KNN's sensitivity to feature-space geometry. However, the relatively small improvements after a moving window size of 600 indicate a decline as a result of more redundant features.

Logistic Regression performs efficiently with all window sizes, surpassing 99% accuracy from window size 400 onwards and obtaining perfect classification (100%) at window sizes of 700 and 800. This indicates that when enough rotational information is captured, the features obtained are highly linearly separable. The rapid increase in Logistic Regression's performance also shows how well the feature extraction method captures the features of gearbox faults. Although it is less than one full shaft rotation, the moving window size of 300 provides useful information about classifier sensitivity in a partial-cycle configuration. Several classifiers still achieve useful accuracy, indicating the presence of relevant transient patterns, even if performance frequently drops at this window size, particularly for Naïve

Bayes. This sensitivity analysis shows that while smaller cycle windows could include discriminative information, optimal and stable performance happens when the window size is at least one full rotating cycle. The reported patterns of performance have been shown to be reliable and independent of a specific train–test split due to the use of five-fold cross-validation. The evaluation minimizes overfitting and provides an accurate estimation of generalization performance for each window size by averaging the results over several folds. The credibility of the experimental results is further confirmed by the continued improvement patterns observed across classifiers.

In general, window sizes between 400 and 600 samples represent an ideal balance between feature robustness and temporal resolution, approximating the gearbox’s core rotational dynamics while maintaining sensitivity to fault-related features. These results show that window size, feature stability, and model features act together to significantly affect classifier performance and demonstrate the significance of physically informed window selection. The purpose of this research was to evaluate a general and computationally lightweight statistical feature-based framework. This framework can be used without requiring extensive knowledge of gearbox design, rotational speed, or accurate synchronization information. That is why Time-Synchronous Averaging (TSA) and Envelope Analysis, which are effective tools for isolating periodic fault signals and noise suppression, are not considered. This research focused on simplicity, interpretability, and ease of deployment, which are important considerations in a number of industrial situations where access to exact shaft speed measurement or consistent operating conditions could be limited.

6. Conclusions and Future Work

In this research, the performance of several traditional ML classifiers was evaluated for gearbox fault detection using vibration signals. The vibration signals were processed using moving window sizes of 300, 400, 500, 600, 700, and 800 to generate features that were then used to train the ML classifiers. In conclusion, SVM, Logistic Regression, and GBC could outperform other classifiers for predicting gearbox defects using vibration signals. The results showed that increasing the moving window size used to extract features from the gearbox vibration signal could improve the ML classifier’s performance.

The proposed framework was designed with practical industrial deployment in mind. Signals obtained from gearboxes in real-world settings can often be polluted by operations and noise outside from surrounding machinery. The statistical features extracted from moving windows could provide robustness to high-frequency noise by averaging temporal variations.

Another important feature of industrial environments is load variation, because changes in load can have an effect on vibration amplitude and frequency. The dataset employed for this study was collected under a variety of load conditions ranging from 0% to 100%, with a step size of 10%, which corresponds to realistic operational scenarios commonly observed in industrial systems.

In the future, the moving window segmentation method will be employed to generate data segments for training unsupervised or one-class learning models for fault detection. During training, only healthy samples will be provided that will allow the model to gather information on the gearbox’s normal working patterns. Based on deviations from the learned healthy behavior, the trained model will then identify defective or unusual samples. This method will allow for early identification of defects without having to look for labeled faulty data, and it could potentially be extended to improve detection robustness and sensitivity using traditional ML or some lightweight deep learning-based approaches.

Author Contributions: Conceptualization, I.u.H.; methodology, I.u.H.; formal analysis, I.u.H.; investigation, I.u.H.; writing—original draft preparation, I.u.H.; writing—review and editing, I.u.H. and

K.P.; supervision, K.P., D.R., and J.W.; project administration, K.P. and J.W.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Lero, the Research Ireland Centre for Software grant 13/RC/2094_P2 and co-funded under the European Regional Development Fund through the Southern and Eastern Regional Operational Programme to Lero www.lero.ie.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and code presented in this study are available in a GitHub repository at <https://github.com/IetezazHassan/Moving-window-based-feature-extraction-method-for-vibration-based-condition-monitoring>, reference number [60].

Acknowledgments: This paper and its research would not be possible without the help of the IMaR team at Munster Technological University. During the preparation of this manuscript/study, the author(s) used ChatGPT (version GPT-4) for the purposes of formatting the structure of this article. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANFIS	Adaptive Neuro Fuzzy Inference Dystem
BiConvLSTM	Bidirectional Convolutional LSTM
CNN	Convolutinal Neural Network
DBN	Deep Belief Network
EWT	Empirical Wavelet Transform
FAWT	Flexible Analytic Wavelet Transform
GBC	Gradient Boosting Classifier
GRBF	Gaussian Radial Basis Function
KNN	K-Nearest Neighbor
ML	ML
NLMS	Normalized Least Mean Square
P2P	Peak to Peak
RC	Research Contribution
RMS	Root Mean Square
SVDD	Support Vector Data Description
SVM	Support Vector Machine
VMD	Variational Mode Decomposition
WTDDS	Wind Turbine Drivetrain Diagnostic Simulation

References

1. Gangsar, P.; Bajpei, A.R.; Porwal, R. A review on deep learning based condition monitoring and fault diagnosis of rotating machinery. *Noise Vib. Worldw.* **2022**, *53*, 550–578.
2. Tamada, S.; Bhattacharjee, D.; Dan, P.K. Review on automatic transmission control in electric and non-electric automotive powertrain. *Int. J. Veh. Perform.* **2020**, *6*, 98–128.
3. Goswami, P.; Rai, R.N. A systematic review on failure modes and proposed methodology to artificially seed faults for promoting PHM studies in laboratory environment for an industrial gearbox. *Eng. Fail. Anal.* **2023**, *146*, 107076.
4. Ayaz, H.; Waqas, M.; Abbas, G.; Abbas, Z.H.; Bilal, M. Multiple re-configurable intelligent surfaces based physical layer eavesdropper detection for V2I communications. *Phys. Commun.* **2023**, *58*, 102074.
5. Hussain, M. The brief review on the gearbox failure identification. *ACADEMICIA Int. Multidiscip. Res. J.* **2021**, *11*, 816–822.

6. Zhang, X.; Yuan, S.; Wang, L. Research on failure and prevention of wind turbine gearbox. In *Proceedings of the 5th International Conference on Information Science, Electrical, and Automation Engineering (ISEAE 2023), Wuhan, China, 24–26 March 2023*; SPIE: Bellingham, WA, USA, 2023; Volume 12748, pp. 1005–1010.
7. ul Hassan, I.; Panduru, K.; Walsh, J. Multi-sensor Fusion Based Recommended Framework for Predictive Maintenance in Next Generation Machines. *Sensors Electron. Instrum. Adv.* **2024**, *183*, 184–186.
8. Wen, Y.; Rahman, M.F.; Xu, H.; Tseng, T.L.B. Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective. *Measurement* **2022**, *187*, 110276.
9. Karabacak, Y.E.; Özmen, N.G.; Gümüsel, L. Intelligent worm gearbox fault diagnosis under various working conditions using vibration, sound and thermal features. *Appl. Acoust.* **2022**, *186*, 108463.
10. Hassan, I.U.; Panduru, K.; Walsh, J. Review of Data Processing Methods Used in Predictive Maintenance for Next Generation Heavy Machinery. *Data* **2024**, *9*, 69.
11. Zhang, Q.; Song, C.; Yuan, Y. Fault Diagnosis of Vehicle Gearboxes Based on Adaptive Wavelet Threshold and LT-PCA-NGO-SVM. *Appl. Sci.* **2024**, *14*, 1212.
12. Gangsar, P.; Tiwari, R. Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review. *Mech. Syst. Signal Process.* **2020**, *144*, 106908.
13. Tong, W. *Mechanical Design and Manufacturing of Electric Motors*; CRC Press: Boca Raton, FL, USA 2022.
14. Dindar, A.; Chaudhury, K.; Hong, I.; Kahraman, A.; Wink, C. An experimental methodology to determine components of power losses of a gearbox. *J. Tribol.* **2021**, *143*, 111203.
15. Kumar, A.; Gandhi, C.; Zhou, Y.; Kumar, R.; Xiang, J. Latest developments in gear defect diagnosis and prognosis: A review. *Measurement* **2020**, *158*, 107735.
16. Murthy, R.N.; Sagar, N.; Prashanth, M.; Srinidhi Kumar, G.; Sachin, B. Predictive Maintenance of Gearbox: A Cost-Effective IoT Approach for Remaining Useful Life Estimation. In *Intelligent Control, Robotics, and Industrial Automation*; Springer: Singapore, 2023; pp. 55–67.
17. Mauricio, A.; Sheng, S.; Gryllias, K. Condition monitoring of wind turbine planetary gearboxes under different operating conditions. *J. Eng. Gas Turbines Power* **2020**, *142*, 031003.
18. Badihi, H.; Zhang, Y.; Jiang, B.; Pillay, P.; Rakheja, S. A comprehensive review on signal-based and model-based condition monitoring of wind turbines: Fault diagnosis and lifetime prognosis. *Proc. IEEE* **2022**, *110*, 754–806.
19. Kumar, A.; Gandhi, C.; Tang, H.; Sun, W.; Xiang, J. Latest innovations in the field of condition-based maintenance of rotatory machinery: A review. *Meas. Sci. Technol.* **2023**, *35*, 022003.
20. OpenEI, Open Energy Data Initiative (OEDI). 2018. Available online: <https://data.openei.org/submissions/623> (accessed on 10 January 2025).
21. Singh, I. *Gearboxes and Harmonic Drives in Electric Drive Systems*; Pencil: Mumbai, India, 2024.
22. Li, G.; Zhu, W. A Review on Up-to-Date Gearbox Technologies and Maintenance of Tidal Current Energy Converters. *Energies* **2022**, *15*, 9236.
23. Feng, Z.; Gao, A.; Li, K.; Ma, H. Planetary gearbox fault diagnosis via rotary encoder signal analysis. *Mech. Syst. Signal Process.* **2021**, *149*, 107325.
24. Sharma, V.; Parey, A. Extraction of weak fault transients using variational mode decomposition for fault diagnosis of gearbox under varying speed. *Eng. Fail. Anal.* **2020**, *107*, 104204.
25. Inturi, V.; Shreyas, N.; Chetti, K.; Sabareesh, G. Comprehensive fault diagnostics of wind turbine gearbox through adaptive condition monitoring scheme. *Appl. Acoust.* **2021**, *174*, 107738.
26. Zhang, L.; Li, Y.; Dong, L.; Yang, X.; Ding, X.; Zeng, Q.; Wang, L.; Shao, Y. Gearbox fault diagnosis using multiscale sparse frequency-frequency distributions. *IEEE Access* **2021**, *9*, 113089–113099.
27. Yu, J.; Wang, S.; Wang, L.; Sun, Y. Gearbox fault diagnosis based on a fusion model of virtual physical model and data-driven method. *Mech. Syst. Signal Process.* **2023**, *188*, 109980.
28. Yao, G.; Wang, Y.; Benbouzid, M.; Ait-Ahmed, M. A hybrid gearbox fault diagnosis method based on GWO-VMD and DE-KELM. *Appl. Sci.* **2021**, *11*, 4996.
29. Vrba, J.; Cejnek, M.; Steinbach, J.; Krbcova, Z. A machine learning approach for gearbox system fault diagnosis. *Entropy* **2021**, *23*, 1130.
30. Hou, H.; Ji, H. Improved multiclass support vector data description for planetary gearbox fault diagnosis. *Control Eng. Pract.* **2021**, *114*, 104867.
31. Hao, H.; Fuzhou, F.; Feng, J.; Xun, Z.; Junzhen, Z.; Jun, X.; Pengcheng, J.; Yazhi, L.; Yongchan, Q.; Guanghui, S.; et al. Gear fault detection in a planetary gearbox using deep belief network. *Math. Probl. Eng.* **2022**, *2022*, 9908074.
32. Azamfar, M.; Singh, J.; Bravo-Imaz, I.; Lee, J. Multisensor data fusion for gearbox fault diagnosis using 2-D convolutional neural network and motor current signature analysis. *Mech. Syst. Signal Process.* **2020**, *144*, 106861.

33. Shi, J.; Peng, D.; Peng, Z.; Zhang, Z.; Goebel, K.; Wu, D. Planetary gearbox fault diagnosis using bidirectional-convolutional LSTM networks. *Mech. Syst. Signal Process.* **2022**, *162*, 107996.
34. Jamil, F.; Verstraeten, T.; Nowé, A.; Peeters, C.; Helsen, J. A deep boosted transfer learning method for wind turbine gearbox fault detection. *Renew. Energy* **2022**, *197*, 331–341.
35. Wu, M.; Shukla, S.; Vrancken, B.; Verbeke, M.; Karsmakers, P. Data-Driven Approach to Identify Acoustic Emission Source Motion and Positioning Effects in Laser Powder Bed Fusion with Frequency Analysis. *Procedia CIRP* **2025**, *133*, 531–536.
36. Wu, M.; Yao, Z.; Verbeke, M.; Karsmakers, P.; Gorissen, B.; Reynaerts, D. Data-driven models with physical interpretability for real-time cavity profile prediction in electrochemical machining processes. *Eng. Appl. Artif. Intell.* **2025**, *160*, 111807.
37. Available online: <https://tinyurl.com/52hpvsan> (accessed on 28 January 2026).
38. Shukla, P.K.; Roy, V.; Chandanan, A.K.; Sarathe, V.K.; Mishra, P.K. A Wavelet Features and Machine Learning Founded Error Analysis of Sound and Trembling Signal. *SN Comput. Sci.* **2023**, *4*, 717.
39. Iniesta, R.; Stahl, D.; McGuffin, P. Machine learning, statistical learning and the future of biological research in psychiatry. *Psychol. Med.* **2016**, *46*, 2455–2465.
40. Altaf, M.; Akram, T.; Khan, M.A.; Iqbal, M.; Ch, M.M.I.; Hsu, C.H. A new statistical features based approach for bearing fault diagnosis using vibration signals. *Sensors* **2022**, *22*, 2012.
41. Bobade, P.; Vani, M. Stress detection with machine learning and deep learning using multimodal physiological data. In Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020; pp. 51–57.
42. Dastile, X.; Celik, T.; Potsane, M. Statistical and machine learning models in credit scoring: A systematic literature survey. *Appl. Soft Comput.* **2020**, *91*, 106263.
43. Tayyab, S.M.; Asghar, E.; Pennacchi, P.; Chatterton, S. Intelligent fault diagnosis of rotating machine elements using machine learning through optimal features extraction and selection. *Procedia Manuf.* **2020**, *51*, 266–273.
44. Sheshkal, S.A.; Riegler, M.A.; Hammer, H.L. ML-Peaks: Chip-seq peak detection pipeline using machine learning techniques. In Proceedings of the 2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS), L'Aquila, Italy, 22–24 June 2023; pp. 335–340.
45. Javeed, M.; Jalal, A.; Kim, K. Wearable sensors based exertion recognition using statistical features and random forest for physical healthcare monitoring. In Proceedings of the 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST), Islamabad, Pakistan, 12–16 January 2021; pp. 512–517.
46. Koklu, M.; Ozkan, I.A. Multiclass classification of dry beans using computer vision and machine learning techniques. *Comput. Electron. Agric.* **2020**, *174*, 105507.
47. Kumar, P.; Bhatnagar, R.; Gaur, K.; Bhatnagar, A. Classification of imbalanced data: Review of methods and applications. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1099*, 012077.
48. Mooijman, P.; Catal, C.; Tekinerdogan, B.; Lommen, A.; Blokland, M. The effects of data balancing approaches: A case study. *Appl. Soft Comput.* **2023**, *132*, 109853.
49. Priyanka; Kumar, D. Decision tree classifier: A detailed survey. *Int. J. Inf. Decis. Sci.* **2020**, *12*, 246–269.
50. Costa, V.G.; Pedreira, C.E. Recent advances in decision trees: An updated survey. *Artif. Intell. Rev.* **2023**, *56*, 4765–4800.
51. Sheykhmousa, M.; Mahdianpari, M.; Ghanbari, H.; Mohammadimanesh, F.; Ghamisi, P.; Homayouni, S. Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 6308–6325.
52. Pisner, D.A.; Schnyer, D.M. Support vector machine. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 101–121.
53. Karal, Ö. Performance comparison of different kernel functions in SVM for different k value in k-fold cross-validation. In Proceedings of the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 15–17 October 2020; pp. 1–5.
54. Genuer, R.; Poggi, J.M. *Random Forests with R*; Springer: Cham, Switzerland, 2020.
55. Boateng, E.Y.; Otoo, J.; Abaye, D.A. Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: A review. *J. Data Anal. Inf. Process.* **2020**, *8*, 341–357.
56. Lubis, A.R.; Lubis, M.; Al-Khowarizmi. Optimization of distance formula in K-Nearest Neighbor method. *Bull. Electr. Eng. Informatics* **2020**, *9*, 326–338.
57. Ige, T.; Kiekintveld, C.; Piplai, A.; Waggler, A.; Kolade, O.; Matti, B.H. An investigation into the performances of the Current state-of-the-art Naive Bayes, Non-Bayesian and Deep Learning Based Classifier for Phishing Detection: A Survey. *arXiv* **2024**, arXiv:2411.16751.
58. Al-Haddad, L.A.; Jaber, A.A.; Hamzah, M.N.; Fayad, M.A. Vibration-current data fusion and gradient boosting classifier for enhanced stator fault diagnosis in three-phase permanent magnet synchronous motors. *Electr. Eng.* **2024**, *106*, 3253–3268.

59. Ma, X.; Lin, Y.; Nie, Z.; Ma, H. Structural damage identification based on unsupervised feature-extraction via variational auto-encoder. *Measurement* **2020**, *160*, 107811.
60. Hassan, I. Moving Window Based Feature Extraction Method for Vibration-Based Condition Monitoring. 2025. Available online: <https://github.com/IetezazHassan/Moving-window-based-feature-extraction-method-for-vibration-based-condition-monitoring> (accessed on 28 January 2026).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.