energies

MDPI

Article

Classification Analytics for Wind Turbine Blade Faults: Integrated Signal Analysis and Machine Learning Approach

Waqar Ali *, Idriss El-Thalji, Knut Erik Teigen Giljarhus and Andreas Delimitis

Department of Mechanical and Structural Engineering and Materials Science, University of Stavanger, 4021 Stavanger, Norway; idriss.el-thalji@uis.no (I.E.-T.); knut.e.giljarhus@uis.no (K.E.T.G.); andreas.delimitis@uis.no (A.D.)

* Correspondence: ali.waqar@uis.no

Abstract: Wind turbine blades are critical components of wind energy systems, and their structural health is essential for reliable operation and maintenance. Several studies have used time-domain and frequency-domain features alongside machine learning techniques to predict faults in wind turbine blades, such as erosion and cracks. However, a key gap remains in integrating these methods into a unified framework for fault prediction, which could offer a more comprehensive solution for diagnosing faults. This paper presents an approach to classify faults in wind turbine blades by leveraging well-known signals and analysis with machine learning techniques. The methodology involves a detailed feature engineering process that extracts and analyzes features from the time and frequency domains. Open-source vibration data collected from an experimental setup (where a small wind turbine with an artificially eroded and cracked blade was tested) were utilized. The time- and frequency-domain features were extracted and analyzed using various machine learning algorithms. It was found that erosion and crack faults have unique time- and frequency-domain features. The crack fault introduces an amplitude modulation in the vibration time wave, which produces sidebands around the fundamental frequency in the frequency domain. However, erosion fault introduces asymmetricity and flatness to the vibration time wave, which produces harmonics in the frequency-domain plot. The results also highlighted that utilizing both time- and frequency-fault features enhances the performance of the machine learning algorithms. This study further illustrates that even though some machine learning algorithms provide similar high classification accuracy, they might differ in quantifying error Types I, II, and, III, which is extremely important for maintenance engineers, as it might lead to undetected fault events and false alarm events.

Keywords: predictive maintenance; machine learning; diagnostic analytics; erosion fault; crack fault; wind turbine blade

Citation: Ali, W.; El-Thalji, I.; Giljarhus, K.E.T.; Delimitis, A. Classification Analytics for Wind Turbine Blade Faults: Integrated Signal Analysis and Machine Learning Approach. *Energies* **2024**, *17*, 5856. https://doi.org/10.3390/en17235856

Academic Editors: Filipe Magalhães and Sérgio Pereira

Received: 26 October 2024
Revised: 18 November 2024
Accepted: 20 November 2024
Published: 22 November 2024

1. Introduction

Wind energy is a rapidly growing sector within the renewable energy landscape, driven by the global demand for sustainable and clean energy sources [1]. The reliability and efficiency of wind energy systems heavily depend on the structural integrity of wind turbine blades. These blades are exposed to various environmental and operational stresses, which can lead to structural faults such as cracks and erosion [2]. The early detection and classification of such faults are crucial for preventing catastrophic failures, minimizing downtime, and reducing maintenance costs [3]. Traditional inspection methods, such as visual and manual checks, are time-consuming, labor-intensive, and often subjective [4]. Traditional methods for fault detection often rely on manual inspections and basic signal processing techniques, which may not offer the required accuracy or efficiency [5]. Consequently, there is a growing interest in leveraging advanced machine learning techniques to enhance the fault detection capabilities in wind turbine blades [6].

In recent years, however, proactive maintenance approaches have shown promise in addressing these limitations by enabling earlier fault detection and reducing downtime.

For instance, an early-warning system for detecting power cable weaknesses in offshore wind farms aids in the timely supervision of critical components [7]. Additionally, a cost-effective maintenance plan that considers multiple defect types in wind turbine blades has been shown to enhance the reliability of maintenance strategies [8]. An integrated preventive–proactive–reactive approach for offshore wind farms has been proposed to optimize energy production and minimize maintenance costs through a combined strategy [9]. Such approaches highlight the advancements in proactive maintenance, moving beyond reactive methods to enable predictive and condition-based strategies that minimize operational risks.

Wind turbine blades are susceptible to several types of damage that can significantly impact their performance and lifespan. Lightning damage is particularly common, with delamination occurring in 72.4% of cases, and can lead to severe structural failures. Fatigue damage arises from cyclic loading, often resulting in cracks at the blade root, gradually degrading the blade's integrity. Leading edge erosion caused by exposure to airborne particulates can reduce aerodynamic efficiency, leading to a 5% decrease in energy production. Additionally, icing damage can further reduce electricity generation by over 10%, with potential losses exceeding 20% during extended turbine shutdowns. These issues highlight the critical need for effective monitoring and maintenance strategies to ensure the longevity and efficiency of wind turbine blades [10].

Numerous research studies have been conducted to address this critical issue, offering various solutions to improve the detection and differentiation of faults in wind turbine blades. Several studies utilized wavelet analysis [11,12]. Recent studies on wind turbine blade fault detection use vibration signal analysis combined with machine learning and transfer learning techniques. One study achieved over 99.5% accuracy using k-nearest neighbors, while another utilized ResNet-50 to classify faults with 99.00% accuracy. A separate approach using decision trees and meta-classifiers reached 80% accuracy with low computational time, highlighting the efficiency of these methods in fault detection [13–15].

Conventional fault detection methods like vibration analysis, acoustic emissions, and ultrasonic testing are widely used but face criticism for poor early fault detection and accuracy. Vibration techniques struggle with early-stage faults [16], while acoustic emissions and ultrasonic testing are unreliable for complex systems and early detection [17,18]. These methods detect dynamic changes but lack precision in fault localization [19,20]. However, sensor placement on blades has improved fault pinpointing [21].

Despite criticisms, conventional methods can be highly effective when enhanced with modern techniques. Advances in vibration analysis have improved early fault detection in rotating machinery [22] and integrating acoustic emission techniques with new signal processing algorithms have boosted accuracy [23]. Similarly, ultrasonic testing combined with machine learning enhances early fault detection [24]. Hybrid models, such as those combining vibration analysis with machine learning [25], or acoustic emissions with AI [26], have also shown improved fault detection. Studies on wind turbine blades confirm that properly applied conventional methods can effectively detect faults in complex structures [27–29].

While traditional fault detection methods have their limitations, ongoing advancements and hybrid approaches have significantly enhanced their reliability and effectiveness, as demonstrated by both recent literature and leading research institutions [30–34].

This study aims to extract both time-domain and frequency-domain features from vibration data to classify normal, erosion, and crack conditions in wind turbine blades. After selecting the most effective features, machine learning models will be evaluated based on their classification performance. Initially, the models will be assessed using time-domain and frequency-domain features separately. Subsequently, the models will be evaluated using time-domain and frequency-domain features combined to determine the most effective method. This analysis will help assess the benefits of integrating both domains for improved fault classification, ultimately facilitating the early detection of cracks and erosion. The approach will contribute to more effective maintenance strategies, enhancing the reliability and efficiency of wind turbine operations.

The following section outlines the feature extraction process from both the time and frequency domains, followed by a description of the machine learning models used for fault classification. The results section presents a performance analysis of the models, comparing the effectiveness (accuracy and error types) of the time domain, frequency domain, and combined features. The discussion section interprets the findings and evaluates the benefits of integrating features from both domains.

2. Materials and Methods

2.1. State-of-the-Art Fault Analysis

Numerous research studies have been conducted to address this critical issue, offering various solutions to improve the detection and differentiation of faults in wind turbine blades. Below is a review of these studies, highlighting the advancements and methodologies proposed to tackle this problem. Discrete wavelet transform (DWT) combined with fast Fourier transform (FFT) to analyze vibration signals for fault detection, specifically targeting erosion in wind turbine blades. This method allows for multi-resolution analysis, which is particularly effective in detecting subtle frequency shifts that indicate erosion damage, which might be missed by traditional Fourier analysis alone [11]. This study demonstrates the effectiveness of frequency-domain analysis in detecting erosion faults in wind turbine blades. However, it leaves room for further research to extend these techniques to the identification of cracks and other types of faults within the frequency domain. Additionally, there is a need for future work to integrate machine learning approaches to enhance the identification and classification of various fault types, providing a more comprehensive and automated fault detection system.

A novel wavelet package energy transmissibility function (WPETF) method, enhances frequency-domain analysis by increasing high-frequency resolution and reducing noise sensitivity. This method is applied to detect faults in wind turbine blades by analyzing vibration signals without requiring loading or excitation information [12].

Another study focuses on analyzing vibration signals caused by cracks in wind turbine blades across four conditions—healthy, light, intermediate, and severe cracks under varying wind velocities. The methodology involves extracting statistical and harmonic features, followed by feature selection using one-way ANOVA, and classification using the k-nearest neighbors algorithm. The performance of the method is compared with neural networks, decision trees, and support vector machines, achieving an accuracy of over 99.5% [13]. Transfer learning-based study proposed to monitor and detect faults in wind turbine blades. By converting vibration signals from turbine blades into radar plots, these visual data representations were analyzed using pre-trained deep learning models (ResNet-50, AlexNet, VGG-16, and GoogleNet). The paper focuses on classifying four specific types of faults: blade bend fault, hub-blade loose connection, blade erosion fault, and blade pitch angle twist. Among these models, ResNet-50 achieved the highest accuracy (99.00%), demonstrating the effectiveness of transfer learning in enhancing fault detection capabilities [14].

Another study focuses on the classification of wind turbine blade faults using statistical parameters derived from quiver signals. The process involves feature extraction, selection, and categorization using machine learning techniques. Specifically, the decision tree method is employed to select statistical characteristics from vibration signals obtained via a data acquisition system (DAQ). The selected features are then classified using meta-classifiers, including random subspace and random committee classifiers. The research highlights the effectiveness of the random committee classifier, which achieves an accuracy of 80% with a computing time of 0.01 s, offering a quick and efficient approach to fault classification in wind turbine blades [15].

Conventional fault detection methods such as vibration analysis, acoustic emissions, and ultrasonic testing have long been utilized in monitoring the health of mechanical systems. However, these methods have come under criticism due to their limitations in accuracy and early fault detection. For example, vibration-based fault detection methods

have highlighted the challenges in detecting early-stage faults due to the reliance on signal processing techniques that might miss subtle indicators of failure [16] also traditional vibration analysis has limitations in providing accurate early warnings, pointing out the potential for unexpected breakdowns due to undetected faults. The effectiveness of acoustic emission techniques has also been critiqued, arguing that they often fail to provide reliable early fault detection in complex mechanical systems [17]. Ultrasonic testing has also been noted for its limitations, particularly in detecting faults at an early stage [18]. While these methods are useful, they are often not sufficient for early detection in highly dynamic environments [19]. For instance, vibration analysis can detect changes in the dynamic behavior of blades but may not precisely pinpoint the fault location or severity [20]. However, researchers have also shown the effectiveness of vibration-based fault location by placing sensors on the blade structure [21].

Despite these critiques, there is substantial evidence supporting the effectiveness of conventional methods, especially when enhanced with modern techniques. Advances in vibration analysis have significantly improved early fault detection in rotating machinery, making these methods more reliable [22]. Acoustic emission techniques, when integrated with newer signal processing algorithms, can provide accurate early fault detection [23]. Ultrasonic testing, when combined with machine learning techniques, has also been shown to enhance early fault detection accuracy [24]. Moreover, recent studies have provided compelling evidence that conventional methods, when combined with hybrid approaches, can be highly effective. A hybrid model combining vibration analysis with machine learning techniques resulted in improved fault detection accuracy [25]. Similarly, the potential of combining acoustic emissions with machine learning has been highlighted, enhancing the detection of early-stage faults in various mechanical systems [26]. Additionally, studies focused on wind turbine blades have shown that conventional vibration-based and acoustic methods, when applied correctly, can effectively detect faults in these large and complex structures [27,28]. The effectiveness of using advanced ultrasonic testing combined with AI algorithms for early fault detection in complex systems has also been demonstrated [29]. Improved signal processing methods in vibration analysis could overcome traditional limitations and provide early detection capabilities [35].

Recent advancements in fault detection methodologies have demonstrated the growing utility of phase-space analysis for identifying early-stage anomalies in mechanical systems. By visualizing system trajectories in phase space, this approach captures complex, nonlinear dynamics that traditional methods often miss. For instance, one study proposed an intelligent model utilizing phase-space analysis to enhance fault classification accuracy in single-circuit transmission lines [36]. Similarly, another review highlighted phase-space topology methods for vibration-based fault diagnostics, emphasizing their effectiveness in nonlinear systems [37]. Further research explored vibration signal analysis using symbolic dynamics, showcasing improved gearbox fault diagnosis through advanced classification techniques [38]. Additionally, a comprehensive review demonstrated the critical role of phase-space analysis in early fault detection for induction motors, emphasizing its impact on maintenance optimization and system reliability [39]. Collectively, these studies underscore the robustness and versatility of phase-space analysis in advancing fault detection frameworks.

Advanced deep learning approaches are increasingly employed to directly extract meaningful features from raw signals, improving accuracy and efficiency in fault detection and classification. Ref. [40] developed a deep learning-based 1D barcode detector that leverages dilated convolutions to capture intricate patterns in signal data, optimizing feature extraction without extensive preprocessing.

2.2. Methodology

The methodology for fault detection and classification in wind turbine blades using machine learning encompasses several critical steps to improve the accuracy and efficiency of fault identification. The approach in Figure 1 begins with data collection using an open-

source dataset provided by [41], which includes vibration data essential for fault diagnosis. Following data collection, feature engineering is performed in two distinct domains: the time domain and the frequency domain. Time-domain features capture statistical and temporal characteristics of the vibration signals, while frequency-domain features are derived through techniques such as Fast Fourier Transform to analyze spectral properties. After feature extraction, the next step involves feature evaluation and selection to identify and retain the most informative features that effectively distinguish between healthy and faulty blade conditions. This is followed by model training and validation, where various machine learning algorithms are trained on the selected features and validated using techniques such as cross-validation to ensure robust performance. Finally, performance evaluation assesses the accuracy and effectiveness of the trained models using metrics like precision, recall, and F1 score to determine their capability in classifying different fault types and healthy states. This systematic methodology ensures a comprehensive approach to enhancing fault detection and classification in wind turbine blades through well-known machine learning techniques.

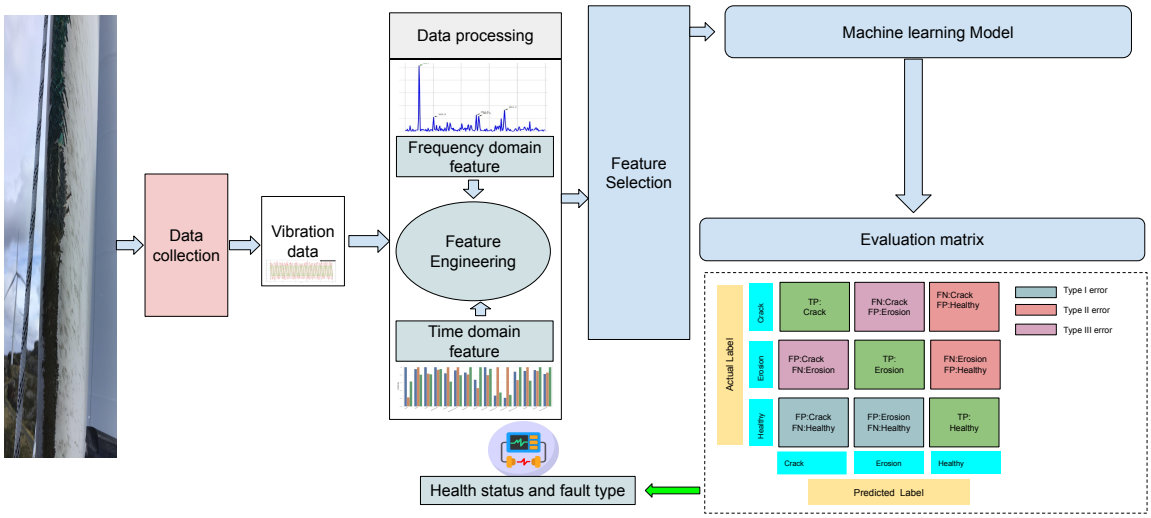


Figure 1. Overview of the methodology used in this study, illustrating key components and workflows for data processing and analysis.

2.3. Data Collection

Data collection is a crucial step in this study, using an open-source dataset from Mendeley provided by Ogaili et al. [41]. This dataset includes vibration data from wind turbine blades under various conditions and wind speeds. For this research, we focused on three specific states: healthy blades, blades with surface erosion, and blades with cracks. High-resolution sensors. were used to capture vibration signals near the turbine hub. The data, recorded at different wind speeds, is stored in CSV format, with each file containing time-domain signals.

2.4. Data Augmentation

Data augmentation techniques were employed to expand the dataset and enhance the model's generalization ability across varied scenarios. Figure 2 illustrates the use of overlapping segments in time series data augmentation, a technique pivotal for enhancing machine learning models. This method effectively increases the dataset's diversity by segmenting the time series signal into overlapping parts. Each segment, depicted in different colors in the image, is extracted with a fixed size of 256 points and a step size of 6 points, which ensures consecutive segments share some common data. This overlap captures a broader range of variations within the signal, thereby improving the model's capacity to generalize and detect subtle patterns. Such augmentation not only enriches the

representation of the signal but also enhances the model's robustness to noise and minor variations. The primary objective of this augmentation was to expand the dataset, which facilitates more comprehensive feature extraction and robust model training. Subsequent steps involve using these augmented segments to compute features, train models, and validate their performance.

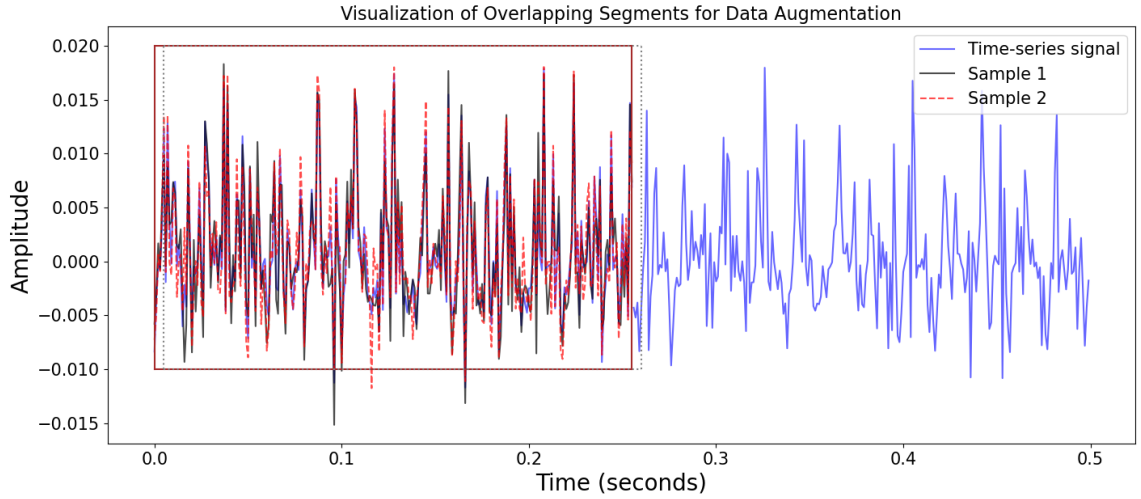


Figure 2. Time series signals with overlapping segments highlighted. The plot shows the entire signal in blue, with the first two overlapping segments highlighted in black and dark red, respectively. Each segment is annotated with its sample number.

2.5. Feature Engineering

Feature engineering is conducted in two primary domains—time domain and frequency domain. Time-domain features capture the statistical and temporal characteristics of the vibration signals and frequency-domain features are derived using techniques such as Fast Fourier Transform to analyze the spectral properties of the signals. These features are then used for feature selection and evaluation to identify the most informative characteristics for fault detection.

2.5.1. Time-Domain Features

Time-domain analysis involves examining the signal's characteristics over time to understand system behavior and identify anomalies indicative of faults. Various statistical features are calculated and compared to effectively distinguish between healthy and faulty conditions (as detailed in Table 1). In Table 1, N represents the total number of data points, and x_i denotes the measured i-th data point in the dataset.

Table 1. Definitions and formulas of statistical measures.

Measure	Definition	Formula
Mean	The average of all data points.	$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$
RMS	The root mean square of the data points.	$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
Peak	The maximum absolute value of the data points.	$\text{Peak} = \max(\lvert x_i \rvert)$
Peak-to-Peak	The difference between the maximum and minimum values.	$\text{P2P} = x_{\max} - x_{\min}$
Variance	The measure of data dispersion from the mean.	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$
Standard Deviation	The square root of the variance.	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$

Table 1. *Cont.*

Measure	Definition	Formula
Skewness	The measure of the asymmetry of the data distribution.	Skewness $= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\mu}{\sigma}\right)^3$
Kurtosis	The measure of the "tailedness" of the data distribution.	Kurtosis $= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\mu}{\sigma}\right)^4 - 3$
Crest Factor	The ratio of the peak value to the RMS value.	Crest Factor $= \frac{\text{Peak}}{\text{RMS}}$
Impulse Factor	The ratio of the peak value to the mean value.	Impulse Factor $= \frac{\text{Peak}}{\mu}$
Shape Factor	The ratio of the RMS value to the mean value.	Shape Factor $= \frac{\text{RMS}}{\mu}$
Clearance Factor	The ratio of the peak value to the square of the RMS value.	Clearance Factor $= \frac{\text{Peak}}{(\text{RMS})^2}$
Energy	The sum of the squares of the data points.	Energy $= \sum_{i=1}^{N} x_i^2$
Entropy	The measure of uncertainty or randomness of the data.	Entropy $= -\sum_{i=1}^{N} p_i \log(p_i)$
Zero Crossings	The number of times the data points change sign.	$Z = \sum_{i=1}^{N-1} \mathbf{1}(x_i x_{i+1} < 0)$

One important time-domain characteristic is the autocorrelation function, which provides a measure of how the signal correlates with a delayed version of itself [42]. Autocorrelation is useful for detecting periodicities, trends, and repetitive patterns within the signal that may be linked to underlying faults [42]. The autocorrelation function is defined as follows:

$$R(\tau) = \frac{1}{N}\sum_{i=1}^{N-\tau} x(i)x(i+\tau) \quad (1)$$

where we have the following:

- $R(\tau)$ is the autocorrelation at lag τ;
- $x(i)$ is the value of the signal at time step i;
- τ is the time lag; and
- N is the total number of points in the signal.

By analyzing the autocorrelation, periodic patterns and trends in the data can be identified, helping to pinpoint the presence of faults in the system. This method is particularly beneficial for time series data where regular fluctuations may indicate underlying mechanical issues [43].

2.5.2. Frequency-Domain Features

Frequency-domain analysis involves transforming time-domain signals into their frequency-domain representations using the fast Fourier transform (FFT). This transformation reveals the underlying frequency components, providing insights into periodicities and identifying dominant frequencies. In vibration analysis for fault detection in blades, frequency-domain analysis is particularly useful. By examining the frequency spectra, specific frequencies associated with faults like cracks or erosion can be identified, enabling more precise diagnoses (as detailed in Table 2).

The process begins by pre-processing the time-domain signal, which includes the following steps:

- Mean removal: The mean of the signal is subtracted to center the data around zero, eliminating any DC component that could distort the frequency analysis. This is expressed as follows:

$$x_{\text{centered}}(t) = x(t) - \mu_x \quad (2)$$

  where μ_x is the mean of the signal $x(t)$.
- Detrending: Detrending is applied to remove any linear trends, ensuring that the frequency components reflect only the oscillatory behavior of the signal.

- Windowing: A Hann window is applied to the signal to minimize spectral leakage, which can occur due to discontinuities at the edges of the signal. The Hann window function is defined as follows:

$$w(t) = 0.5\left(1 - \cos\left(\frac{2\pi t}{N-1}\right)\right) \tag{3}$$

where N is the length of the signal.

Following pre-processing, the FFT is performed on the signal. The FFT converts the time-domain signal into its frequency-domain representation, providing a spectrum that reveals the magnitude of different frequency components.

- FFT calculation: The FFT of the signal is computed as follows:

$$\hat{f} = \text{FFT}(x_{\text{windowed}}(t)) \tag{4}$$

The magnitude of the FFT is then obtained, representing the amplitude of each frequency component.

- Frequency array: A corresponding frequency array is generated to map the FFT magnitudes to their respective frequencies:

$$f = \frac{1}{\text{duration}} \times \text{frequency\_indices} \tag{5}$$

Table 2. Frequency-domain features.

Feature Name	Definition	Formula
Sideband frequency amplitude	Amplitude of sideband frequencies around a Fundamental frequency, indicating modulation effects.	$A_{SB} = \frac{1}{N}\sum_{n=1}^{N}\lvert X(f_n)\rvert$
Fundamental frequency amplitude	Amplitude at the system's natural resonant frequency, typically where resonance occurs.	$A_{NF} = \lvert X(f_{nat})\rvert$
2f Amplitude	Amplitude at twice the fundamental frequency, useful for detecting second-order harmonics.	$A_{2f} = \lvert X(2f)\rvert$
3f Amplitude	Amplitude at three times the fundamental frequency, useful for detecting third-order harmonics.	$A_{3f} = \lvert X(3f)\rvert$
Total harmonic distortion (THD)	Ratio of the sum of powers of all harmonic components to the power of the fundamental frequency.	$\text{THD} = \sqrt{\sum_{n=2}^{\infty}\left(\frac{A_{nf}}{A_f}\right)^2}$
Frequency centroid	The weighted average frequency, where each frequency is weighted by its amplitude.	$f_{centroid} = \frac{\sum_{n=1}^{N} f_n\lvert X(f_n)\rvert}{\sum_{n=1}^{N}\lvert X(f_n)\rvert}$
Spectral flatness	Measures how flat or peaky the spectrum is, used to distinguish between noise and tonal components.	$\text{Flatness} = \frac{\left(\prod_{n=1}^{N}\lvert X(f_n)\rvert\right)^{1/N}}{\frac{1}{N}\sum_{n=1}^{N}\lvert X(f_n)\rvert}$

By combining frequency- and time-domain features, a comprehensive analysis framework is established, providing valuable insights into both the frequency and characteristics of vibration signals. This approach enhances the accuracy and reliability of fault detection in turbine blades.

2.6. Machine Learning Algorithms for Classification

This section provides a theoretical overview of the machine learning algorithms used for classification tasks, including XGBoost, random forest, support vector machine (SVM), logistic regression, and multi-layer perceptron (MLP).

2.6.1. XGBoost Classifier

XGBoost (extreme gradient boosting) is an efficient and scalable implementation of gradient boosting that sequentially builds an ensemble of weak learners (decision trees). At each step, it minimizes the following objective function [44]:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{6}$$

where we have the following:

- l is a differentiable loss function (e.g., squared error for regression).
- $\Omega(f_k)$ is the regularization term to control model complexity.
- f_k represents a decision tree, and K denotes the number of trees.

The first term measures how well the model fits the data, and the second term penalizes complex models to avoid overfitting.

2.6.2. Random Forest Classifier

Random forest is an ensemble learning method that combines multiple decision trees. It works by averaging the predictions of individual trees for regression or by taking the majority vote for classification [45]. The general formula for a random forest classifier is as follows:

$$\hat{y} = \text{MajorityVote}(T_1(x), T_2(x), \ldots, T_n(x)) \tag{7}$$

where we have the following:

- $T_i(x)$ represents the prediction from the i^{th} decision tree.

Random forests reduce overfitting by using random subsets of features and samples, making them robust to noisy data.

2.6.3. Support Vector Machine (SVM) Classifier

SVMs aim to find the hyperplane that maximizes the margin between different classes. The optimization problem for a linear SVM is given by [46]:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i \tag{8}$$

where we have the following:

- $\mathbf{w}$ is the weight vector.
- $\mathbf{x}_i$ are the input data points.
- y_i are the class labels ($+1$ or -1).
- b is the bias term.

For non-linear problems, SVMs use kernel functions to map the input space into higher dimensions, where a linear separation is possible.

2.6.4. Logistic Regression Classifier

Logistic regression is a simple yet effective classifier that models the probability of a binary outcome. The probability of the output class is modeled as follows:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)}} \tag{9}$$

where we have the following:

- $\mathbf{w} \cdot \mathbf{x}$ is the dot product of the weight vector and input features.
- b is the bias term.
- The logistic function (sigmoid) maps the linear output to a probability between 0 and 1.

The model parameters are estimated by maximizing the **log-likelihood** function [47].

2.6.5. MLP (Multi-Layer Perceptron) Classifier

MLP is a feedforward artificial neural network composed of input, hidden, and output layers. The forward propagation for one neuron can be expressed as follows:

$$a_j = f\left(\sum_{i=1}^{n} w_{ij}x_i + b_j\right) \tag{10}$$

where we have the following:

- w_{ij} are the weights between neurons.
- x_i are the inputs from the previous layer.
- b_j is the bias term.
- f is the activation function (e.g., ReLU, sigmoid).

The network is trained using backpropagation, which updates the weights to minimize a loss function [48]. The loss function is typically a form of cross-entropy for classification tasks.

2.7. Model Evaluation

The confusion matrix for this multi-class problem (crack, erosion, healthy) can be represented as follows:

$$\begin{bmatrix} TP_{Crack} & FP_{Crack/Erosion} & FP_{Crack/Healthy} \\ FN_{Erosion/Crack} & TP_{Erosion} & FP_{Erosion/Healthy} \\ FN_{Healthy/Crack} & FN_{Healthy/Erosion} & TP_{Healthy} \end{bmatrix}$$

Each cell in this matrix represents the number of instances predicted for each class compared to the actual class. The terms are defined as follows:

- True positive (TP): Correctly predicted positive cases.
- True negative (TN): Correctly predicted negative cases.
- False positive (FP): Incorrectly predicted positive cases.
- False negative (FN): Incorrectly predicted negative cases.
- Accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

  Accuracy measures the overall correctness of the model.
- Precision:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

  Precision measures the proportion of correctly predicted positive observations to the total predicted positive observations. For example, the confusion matrix is as follows:

$$\begin{bmatrix} 34 & 21 & 7 \\ 19 & 40 & 2 \\ 2 & 3 & 69 \end{bmatrix}$$

  where we have the following:

  - Row 1 corresponds to the crack class.
  - Row 2 corresponds to the erosion class.
  - Row 3 corresponds to the healthy class.

  The precision for each class is calculated as follows:

$$\text{Precision for crack} = \frac{TP_{\text{crack}}}{TP_{\text{crack}} + \sum \text{FPs to crack}} = \frac{34}{34 + 19 + 2} = \frac{34}{55} \approx 0.618$$

$$\text{Precision for erosion} = \frac{TP_{\text{Erosion}}}{TP_{\text{Erosion}} + \sum \text{FPs to erosion}} = \frac{40}{40+21+3} = \frac{40}{64} \approx 0.625$$

$$\text{Precision for healthy} = \frac{TP_{\text{Healthy}}}{TP_{\text{Healthy}} + \sum \text{FPs to healthy}} = \frac{69}{69+7+2} = \frac{69}{78} \approx 0.885$$

Finally, the overall (macro-averaged) precision is the average of the individual precisions:

$$\text{Overall precision} = \frac{0.618 + 0.625 + 0.885}{3} = \frac{2.128}{3} \approx 0.709$$

Thus, the overall precision is approximately 0.709.

- Recall (sensitivity or true positive rate):

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

Recall measures the proportion of correctly predicted positive observations to all actual positives.

- Specificity (true negative rate):

$$Specificity = \frac{TN}{TN + FP} \tag{14}$$

Specificity measures the proportion of correctly predicted negative observations to all actual negatives.

- F1 score:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

The F1 score is the harmonic mean of precision and recall.

- False positive rate (FPR):

$$FPR = \frac{FP}{FP + TN} \tag{16}$$

The false positive rate measures the proportion of incorrect positive predictions out of all actual negatives.

The confusion matrix in Figure 3 illustrates the performance of the classification model by showing the relationship between predicted and actual classes across three categories: crack, erosion, and healthy. The matrix highlights three types of errors. Type I errors (false negative), marked in red, occur when the model incorrectly predicts a healthy when the actual class is (either crack or erosion). These errors are represented in the third column. Type II errors, shaded in gray, occur when the model predicts crack or erosion, but the true class is healthy. These errors are shown in the third row, where the model fails to detect the actual class. Type III errors, marked in purple, represent misclassifications between the fault types themselves. This error happens when the model predicts one type of fault (e.g., crack) as another type of fault (e.g., erosion). These misclassifications occur in the off-diagonal cells between the crack and erosion categories.

This visualization provides an understanding of the model's accuracy in distinguishing between fault types and healthy cases, allowing for a deeper analysis of its strengths and limitations.

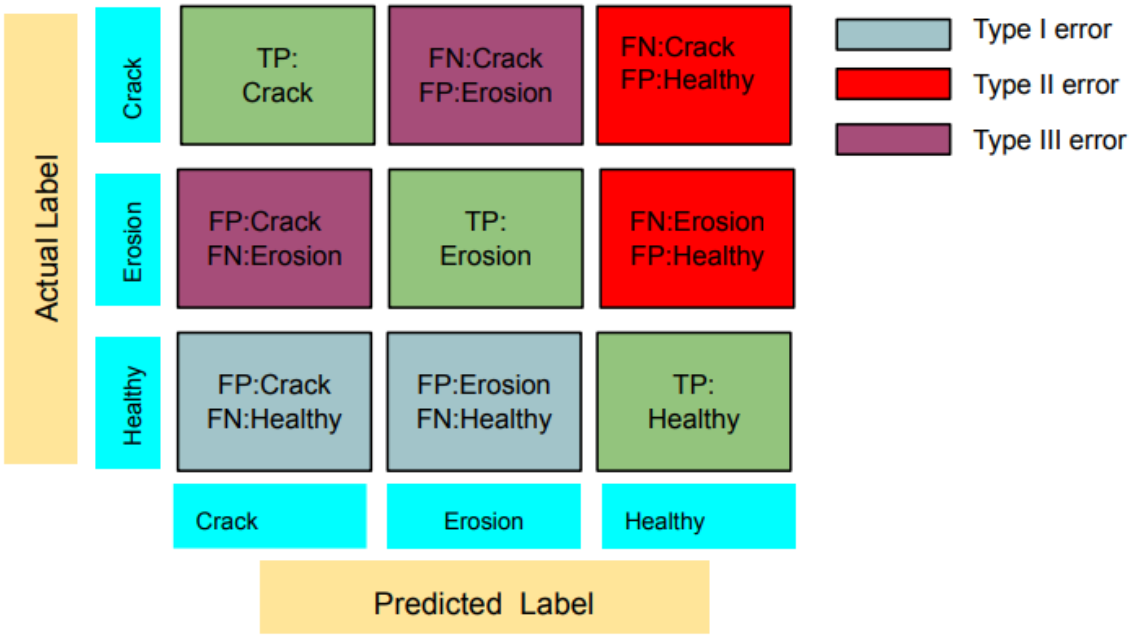


Figure 3. Confusion matrix visualization showing Type I, Type II, and Type III errors for predicted crack, erosion, and healthy label in relation to the true label.

3. Results

In this section, four groups of results are illustrated and described: (1) the time-domain fault features, (2) the frequency-domain fault features, (3) the performance of the applied machine learning algorithms, and (4) the error Type I, II, and III quantifications. Moreover, the performance analytics and error quantification are provided based on three categories: (1) for time-domain fault features only, (2) for frequency-domain fault features only, and (3) combined time- and frequency-domain fault features.

3.1. Results Related to Time-Domain Fault Features

Figure 4 shows the comparison of raw signals and autocorrelation for healthy, cracked, and eroded wind turbine blades.

- Blade in a healthy condition: The raw signal has a lower amplitude with smoother autocorrelation, indicating stable and consistent behavior without significant disturbances, typical of an intact structure.
- Blade with crack fault: The raw signal exhibits higher amplitude fluctuations, approximately 17.42% higher than the healthy condition. The autocorrelation shows distinct periodic peaks, reflecting fault-related periodicity and the presence of amplitude modulation and wave asymmetricity. These peaks highlight the regularity introduced by the crack in the structure.
- Blade with erosion fault: The raw signal displays more irregular variations, with amplitude fluctuations 23.46% higher compared to the healthy condition. The autocorrelation reveals less pronounced periodicity with the presence of wave asymmetricity and shape flatness. This indicates more random, chaotic structural changes due to surface erosion. Additionally, the erosion condition shows a 5.14% increase in amplitude fluctuations compared to the cracked condition, reflecting more severe damage.

The crack fault is characterized by distinct periodic peaks in the autocorrelation, highlighting the regular fault patterns, whereas the erosion condition exhibits more irregular and random patterns, indicating more severe and less structured damage compared to the healthy and crack fault conditions.

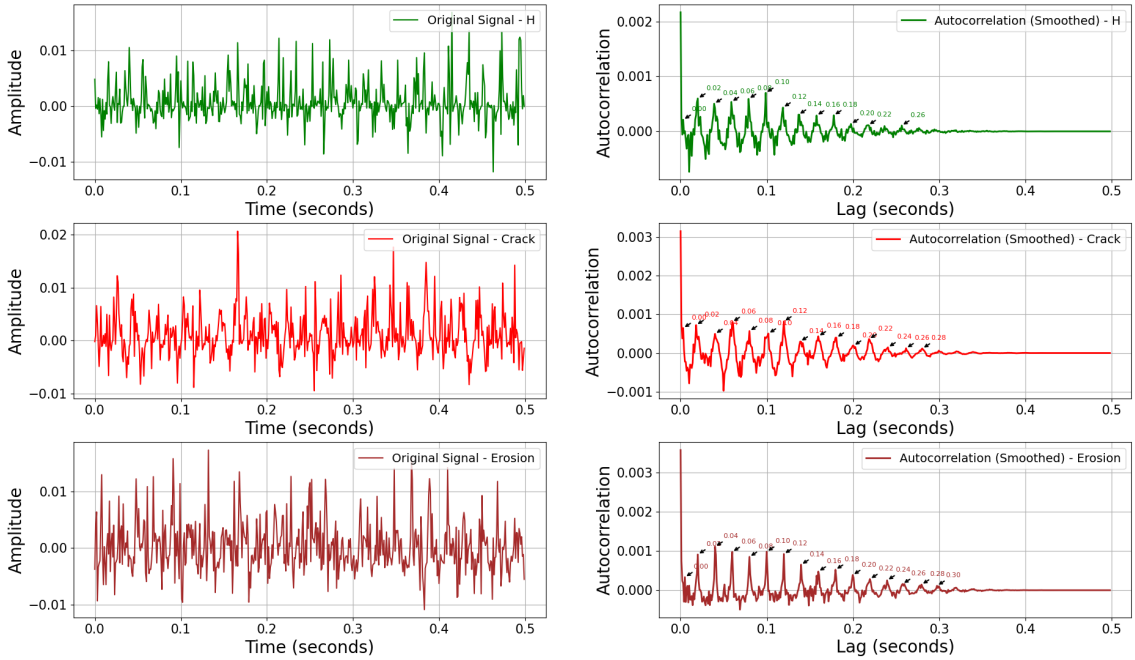


Figure 4. The raw signals and their autocorrelation responses for wind turbine blade faults: healthy, crack, and erosion conditions.

Figure 5 illustrates the comparison of various normalized time-domain features across different fault types: healthy (H), crack (faulty), and erosion (faulty). The bar plot highlights the differences in these features, making it easier to identify which features are more prominent in faulty conditions compared to the healthy state. This visual representation aids in the diagnosis and differentiation of specific faults based on their statistical characteristics.

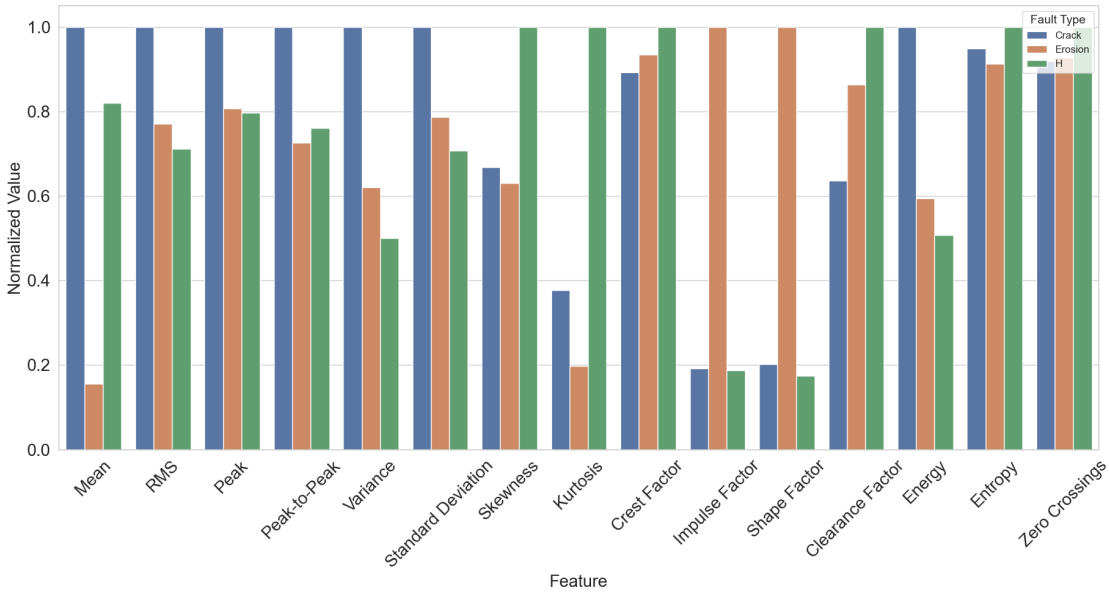


Figure 5. Comparison of normalized time-domain features across fault types.

Based on the Figure 5 features analysis, Table 3 provides a comparison of various statistical features across different fault types: healthy (H), crack (faulty), and erosion (faulty). The table highlights which features have lower or higher values in the healthy state compared to the faulty states, and indicates which features are more important for

distinguishing between specific faults. For instance, features like shape factor and impulse factor are particularly significant for identifying erosion faults due to their higher values.

Table 3. Comparison of features distinguishing healthy and faulty conditions.

Feature	H (Healthy)	Crack (Faulty)	Erosion (Faulty)	Most Important For
Mean	Lower	Higher	Higher	Crack, Erosion
RMS	Lower	Higher	Higher	Crack, Erosion
Peak	Lower	Higher	Higher	Crack, Erosion
Peak-to-Peak	Lower	Higher	Higher	Crack, Erosion
Variance	Lower	Higher	Highest	Erosion
Standard Deviation	Lower	Higher	Highest	Erosion
Skewness	Lower	Higher	Higher	Crack, Erosion
Kurtosis	Lower	Higher	Higher	Crack, Erosion
Crest Factor	Lower	Higher	Higher	Crack, Erosion
Impulse Factor	Lower	Higher	Highest	Erosion
Shape Factor	Lower	Higher	Higher	Crack, Erosion
Clearance Factor	Lower	Higher	Higher	Crack, Erosion
Energy	Lower	Higher	Highest	Erosion
Entropy	Higher	Lower	Lower	Crack, Erosion
Zero Crossings	Higher	Lower	Lower	Crack, Erosion

Time-domain analysis reveals key statistical features crucial for differentiating between healthy and faulty conditions. For erosion faults, important features are the impulse factor, and shape factor, which show higher values compared to healthy and crack conditions, indicating greater variability and peaks.

For crack faults, the significant features are mean, RMS, peak, variance, and standard deviation. These features are higher in crack faults compared to healthy conditions, with mean and variance particularly indicative of distinguishing between crack and erosion.

3.2. Results Related to Frequency-Domain Fault Features

The FFT spectra in Figure 6 illustrate three distinct cases: healthy, crack fault, and erosion fault. It can be observed that each case exhibits unique spectral features.

For the healthy case, the fundamental frequency is 50.1 Hz with an amplitude of around 0.22. For the crack fault case, the amplitude of the fundamental frequency reaches approximately 0.31, higher than both the healthy and erosion fault cases. Notably, sideband frequencies at 32.1 Hz and 68.1 Hz are observed in the cracked case due to the amplitude modulation. These sidebands are absent in both the healthy and erosion fault conditions, making them distinctive indicators of a crack. Furthermore, the amplitude of the fundamental frequency in the cracked case is significantly greater compared to the other two cases.

In the erosion fault case, a distinctive characteristic is the presence of harmonics at 100, 150, 200, and 250 Hz, due to distortion in the wave shape and wave asymmetricity. These harmonics are absent in the healthy case and weaker in the crack fault case. While the amplitude of the fundamental frequency in the erosion case is around 0.27 which is lower than in the cracked case, it is still higher than in the healthy case.

Frequency-domain analysis also provides valuable indicators, including sideband frequency amplitude, fundamental frequency amplitude, and the amplitudes of the 1f, 2f, and 3f components. These features help identify specific fault types by examining the amplitude of various frequency components.

Focusing on these key features from both time and frequency domains enhances the accuracy of fault detection and maintenance strategies. The key features are summarized in Table 4.

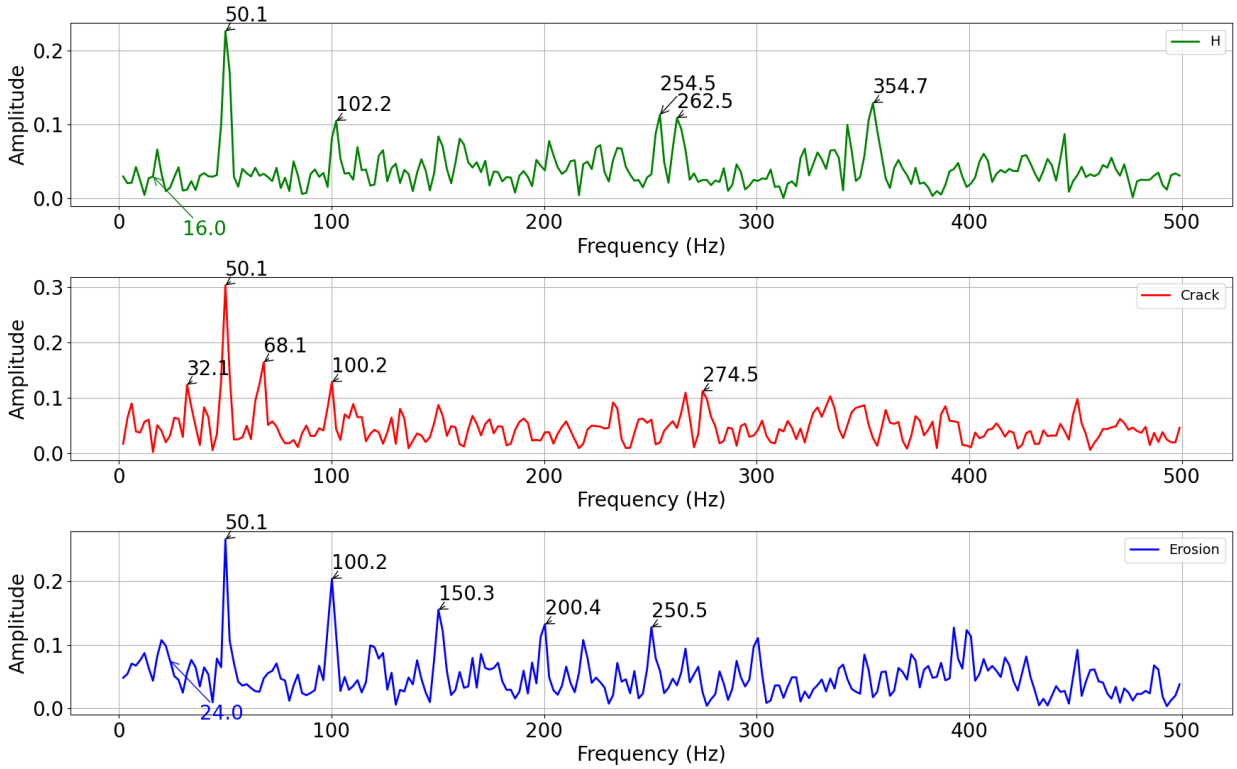


Figure 6. FFT spectrum for healthy, crack fault, and erosion fault blade conditions.

Table 4. Key features for differentiating healthy and faulty conditions.

Feature	Domain	Significance
Variance	Time	Higher in crack faults, indicating greater variability
Standard deviation	Time	Higher in crack faults, indicating greater variability
Impulse factor	Time	Higher in erosion faults, indicating higher peaks
Energy	Time	Higher in crack faults, indicating higher peaks
Shape factor	Time	Higher in erosion faults, indicating higher peaks
Mean	Time	Higher in crack faults compared to healthy and erosion conditions
RMS	Time	Higher in crack faults compared to healthy and erosion conditions
Sideband frequency amplitude	Frequency	symptom of surface crack
fundamental frequency Amplitude	Frequency	Distinguishes between healthy, crack, and erosion
2f Amplitude	Frequency	symptom of surface erosion
3f Amplitude	Frequency	symptom of surface erosion
4f Amplitude	Frequency	symptom of surface erosion

3.3. Results Related to the Performance of the Applied Machine Learning Algorithms

In this section, the performance of five machine learning models on the dataset is evaluated. The models considered include XGBoost, random forest, support vector machine (SVM), logistic regression, and MLP classifier. Each model's performance is assessed based on cross-validation accuracy, test accuracy, precision, recall, and F1 score. Additionally, the confusion matrix for each model provides a detailed breakdown of the classification results. The confusion matrices allow for analysis of the model's ability to correctly classify fault and non-fault instances in wind turbine blades, while also identifying the prevalence of Type I, Type II and, Type III errors. Type I error occurs when the model incorrectly predicts a healthy instance when the actual class is either crack or erosion. These errors result in undetected faults, leading to potential damage and higher costs due to delayed maintenance. These errors are represented in the third column of the confusion matrix, where the actual fault is not recognized, a Type II error, occurs when the model predicts a fault (either crack or erosion) but the actual class is healthy. These errors lead to unnecessary

inspections or maintenance, increasing operational costs. They are shown in the third row of the confusion matrix, where the model mistakenly detects a fault in a healthy instance. Type III errors (misclassification between fault types) represent instances where the model correctly detects a fault but misclassifies its type. For example, the model predicts a crack fault when the actual issue is erosion or vice versa. These misclassifications can result in incorrect maintenance actions, which may not address the actual problem. These errors are located in the off-diagonal cells between the crack and erosion categories in the confusion matrix.

3.3.1. Category 1: Combined Time–Frequency-Domain Fault Features

The performance metrics of the models summarized in Table 5 reveal key differences in their ability to classify faults using combined time–frequency-domain features. The analysis is broken down for each model, including their performance across Type I, Type II, and Type III errors.

Table 5. Model performance comparison based on combined time- and frequency-domain features.

Metric	Logistic Regression	MLP Classifier	XGBoost	Random Forest	SVM
Cross-Validation Accuracy	0.767	0.503	0.976	0.980	0.638
Test Accuracy	0.726	0.365	1.000	1.000	0.599
Precision	0.720	0.701	1.000	1.000	0.671
Recall	0.726	0.365	1.000	1.000	0.599
F1 score	0.722	0.275	1.000	1.000	0.559
Confusion Matrix	[34, 21, 7] [19, 40, 2] [2, 3, 69]	[2, 60, 0] [0, 57, 4] [0, 61, 13]	[62, 0, 0] [0, 61, 0] [0, 0, 74]	[62, 0, 0] [0, 61, 0] [0, 0, 74]	[12, 41, 9] [2, 39, 20] [0, 7, 67]

Logistic regression shows moderate performance with a cross-validation accuracy of 0.767 and a test accuracy of 0.726. The model has a precision of 0.720 and a recall of 0.726, indicating a balance between correctly predicting faults and missing some actual fault instances. The confusion matrix reveals:

- Type I error: A total of five instances, where two healthy cases were misclassified as cracks and three healthy cases were misclassified as erosion.
- Type II error: A total of nine instances, where seven crack faults were misclassified as healthy and two erosion faults were misclassified as healthy.
- Type III error: A total of 40 instances, with 21 crack faults misclassified as erosion and 19 erosion faults misclassified as cracks.

This model struggles with both over-predicting faults (Type I errors) and failing to detect faults (Type II errors), with a significant number of misclassifications between fault types (Type III errors).

MLP classifier performs poorly, reflected by its cross-validation accuracy of 0.503 and a test accuracy of 0.365. The model achieves a precision of 0.701, but the low recall of 0.365 and F1 score of 0.275 indicate a high rate of misclassification. The confusion matrix shows:

- Type I error: A total of 61 instances, where 61 healthy cases were misclassified as erosion.
- Type II error: A total of four instances, where four erosion faults were misclassified as healthy.
- Type III error: A total of 60 instances, with 61 crack faults misclassified as erosion.

The large numbers of Type III errors and Type I errors show that MLP struggles to distinguish between crack and erosion faults and distinguish between faulty and healthy conditions.

XGBoost demonstrates outstanding performance, achieving perfect metrics with a cross-validation accuracy of 0.976 and a test accuracy of 1.000. The model has a precision, recall, and F1 score of 1.000. The confusion matrix confirms the following:

- Type I error: Zero instances of errors.
- Type II error: Zero instances of errors.
- Type III error: Zero instances of misclassification between faults.

XGBoost achieves perfect classification, with no misclassifications of any type.

Random forest, similar to XGBoost, achieves perfect performance, with a cross-validation accuracy of 0.980 and a test accuracy of 1.000. The model achieves perfect scores across precision, recall, and F1 score. The confusion matrix reveals the following:

- Type I errors: Zero instances of errors.
- Type II errors: Zero instances of errors.
- Type III errors: Zero instances of misclassification between faults.

Like XGBoost, random forest shows no errors across any error type, making it highly effective for fault classification.

SVM (support vector machine) shows moderate performance with a cross-validation accuracy of 0.638 and a test accuracy of 0.599. The model achieves a precision of 0.671 and recall of 0.599, reflecting its ability to correctly identify faults, though there are still some misclassifications. The confusion matrix reveals:

- Type I error: A total of seven instances, where seven healthy cases were misclassified as erosion.
- Type II error: A total of 29 instances, where 9 crack faults were misclassified as healthy and 20 erosion faults were misclassified as healthy.
- Type III error: A total of 43 instances, with 41 crack faults misclassified as erosion and 2 erosion faults as cracks.

SVM shows reasonable performance, but with notable errors in misclassifying faults, particularly when distinguishing between fault types (Type III errors) and also have significant Type II errors.

3.3.2. Category 2: Time-Domain Fault Features Only

The performance metrics in Table 6 reveal how various models perform when using only time-domain features for fault classification. The evaluation includes an analysis of precision, recall, and F1 score, alongside a confusion matrix that reflects the occurrence of Type I, Type II, and Type III errors.

Table 6. Model performance comparison based on time-domain features only.

Metric	Logistic Regression	MLP Classifier	XGBoost	Random Forest	SVM
Cross-Validation Accuracy	0.767	0.471	0.983	0.985	0.638
Test Accuracy	0.766	0.482	0.985	0.975	0.599
Precision	0.761	0.442	0.985	0.975	0.671
Recall	0.766	0.482	0.985	0.975	0.599
F1 score	0.763	0.411	0.985	0.975	0.559
Confusion Matrix	[41, 15, 6] [20, 39, 2] [0, 3, 71]	[56, 0, 6] [61, 0, 0] [35, 0, 39]	[61, 1, 0] [2, 59, 0] [0, 0, 74]	[59, 3, 0] [2, 59, 0] [0, 0, 74]	[12, 41, 9] [2, 39, 20] [0, 7, 67]

Logistic regression performs moderately well with a cross-validation accuracy of 0.767 and a test accuracy of 0.766. The model achieves a precision of 0.761 and a recall of 0.766, indicating a balance between correctly predicting positive cases and missing some actual fault cases. The F1 score of 0.763 reflects this balance. The confusion matrix shows the following:

- Type I error: A total of three instances, where three healthy cases were misclassified as erosion.
- Type II error: A total of eight instances, where six crack faults were misclassified as healthy, and two erosion faults were misclassified as healthy.
- Type III error: A total of 35 instances, where 15 crack faults were misclassified as erosion, and 20 erosion faults were misclassified as crack.

These values indicate that while the model effectively distinguishes healthy cases, it still struggles to differentiate between fault types (crack and erosion).

MLP classifier performs poorly in this scenario, as reflected by its cross-validation accuracy of 0.471 and test accuracy of 0.482. The model's precision of 0.442 indicates a high number of false positives, and with a recall of 0.482, it misses a significant number of actual faults. The confusion matrix shows the following:

- Type I error: A total of 35 instances, where 35 were misclassified as crack when it was healthy.
- Type II error: A total of six instances, where six crack cases were misclassified as healthy.
- Type III error: A total of 61 instances, where 61 erosion faults were misclassified as cracks.

The high frequency of Type III and Type I errors indicates that the MLP has difficulty differentiating between crack and erosion faults, as well as distinguishing faulty conditions from healthy ones.

XGBoost performs excellently, achieving a cross-validation accuracy of 0.983 and test accuracy of 0.985. The model achieves perfect metrics for precision (0.985), recall (0.985), and an F1 score of 0.985. The confusion matrix confirms the model's strong performance:

- Type I error: Zero instances of errors.
- Type II error: Zero instances of errors.
- Type III error: There were a total of three instances, with two cases misclassifying erosion as crack and one case of crack being misclassified as erosion.

XGBoost performs perfectly in identifying healthy cases and fault types, but misclassification is present when we only use time-domain features.

Random forest also performs very well with a cross-validation accuracy of 0.985 and a test accuracy of 0.975. The model shows similarly high precision, recall, and F1 score, all close to 0.975. The confusion matrix shows the following:

- Type I error: Zero instances of errors.
- Type II error: Zero instances of errors.
- Type III error: A total of five instances, with two cases misclassifying erosion as crack and three cases misclassifying crack as erosion.

Random forest can effectively distinguish between faults and healthy conditions; however, its error rate in differentiating specific fault types is higher than that of XGBoost.

SVM (support vector machine) shows moderate performance, achieving a cross-validation accuracy of 0.638 and a test accuracy of 0.599. The model achieves a precision of 0.671 and a recall of 0.599, reflecting some ability to correctly predict faults. However, the confusion matrix reveals a moderate number of errors:

- Type I error: A total of seven instances, where seven healthy cases were misclassified as erosion.
- Type II error: A total of 29 instances, where 9 crack faults were misclassified as healthy and 20 erosion faults were misclassified as healthy.
- Type III error: A total of 43 instances, with 41 crack faults misclassified as erosion and 2 erosion faults as cracks.

The SVM model struggles most with Type III errors, where it misclassifies fault types, though it performs better in distinguishing healthy cases.

3.3.3. Category 3: Frequency-Domain Fault Features Only

The performance metrics in Table 7 provide insights into the model's ability to classify faults using frequency-domain features. The evaluation examines precision, recall, and F1 score, along with the confusion matrix to identify the occurrence of Type I, Type II, and Type III errors.

Table 7. Model performance comparison based on frequency-domain features only.

Metric	Logistic Regression	MLP Classifier	XGBoost	Random Forest	SVM
Cross-Validation Accuracy	0.597	0.708	0.939	0.930	0.715
Test Accuracy	0.584	0.716	0.909	0.944	0.726
Precision	0.656	0.720	0.908	0.945	0.735
Recall	0.584	0.716	0.909	0.944	0.726
F1-Score	0.537	0.711	0.908	0.943	0.721
Confusion Matrix	[33, 4, 25] [8, 11, 42] [3, 0, 71]	[40, 13, 9] [7, 36, 18] [3, 6, 65]	[58, 3, 1] [4, 51, 6] [0, 4, 70]	[62, 0, 0] [2, 52, 7] [0, 2, 72]	[40, 11, 11] [7, 36, 18] [1, 6, 67]

Logistic regression performed modestly with a test accuracy of 0.584, precision of 0.656, and recall of 0.584. The confusion matrix reveals the following:

- Type I error: There were three instances in total where a healthy case was misclassified as a crack.
- Type II error: There were 67 instances in total, with 25 crack faults misclassified as healthy and 42 erosion faults misclassified as healthy.
- Type III error: A total of 12 instances where 4 crack fault cases were classified as erosion and 8 erosion cases were classified as crack.

The results indicate that logistic regression faces considerable challenges in accurately identifying faults, particularly with Type II errors, where the model incorrectly indicates a healthy condition.

MLP classifier achieved slightly better performance, with a test accuracy of 0.716, precision of 0.720, and recall of 0.716. However, it still misclassifies a considerable number of instances:

- Type I error: A total of nine instances, with three healthy cases misclassified as crack and six as erosion.
- Type II error: A total of 27 instances, with 9 crack cases and 18 erosion cases classified as healthy.
- Type III error: A total of 20 instances were misclassified: 13 crack faults were labeled as erosion, and 7 erosion faults were labeled as crack.

Although the MLP classifier improves overall accuracy, it struggles with Type III errors, frequently misclassifying between crack and erosion as well as Type II errors.

XGBoost showed excellent performance with a test accuracy of 0.909, precision of 0.908, and recall of 0.909. The confusion matrix confirms this strong performance:

- Type I error: A total of four instances, where four healthy cases were predicted as erosion.
- Type II error: A total of seven instances, where one crack fault and six erosion faults were predicted as healthy.
- Type III error: A total of seven instances occurred, where three crack faults were misclassified as erosion and four erosion faults were misclassified as crack.

Random forest performed similarly well with a test accuracy of 0.944, precision of 0.945, and recall of 0.944, indicating that the model is highly effective at fault classification:

- Type I error: A total of two instances, where two healthy cases were predicted as erosion.

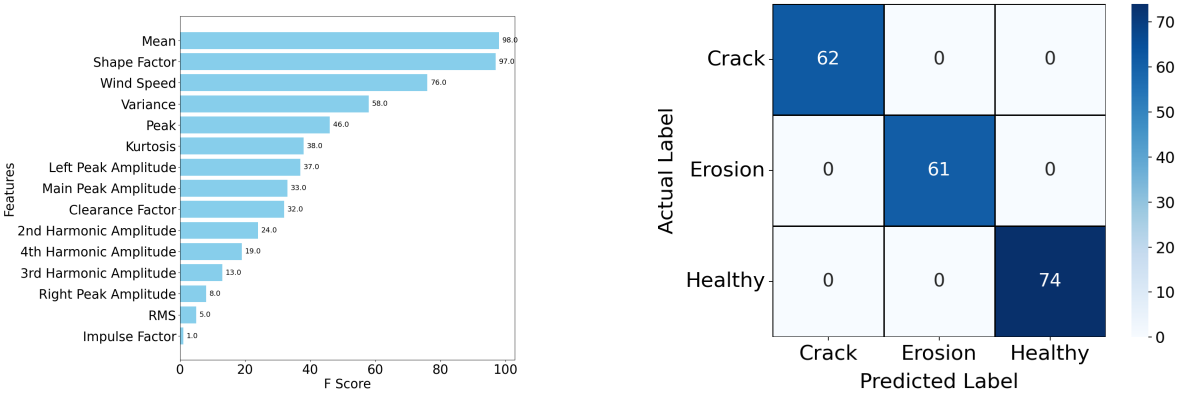
- Type II error: A total of seven instances, where seven erosion faults were predicted as healthy.
- Type III errors (misclassification between faults): A total of two instances, where two erosion faults were misclassified as crack.

SVM (support vector machine) had moderate performance, achieving a test accuracy of 0.726, precision of 0.735, and recall of 0.726. However, it misclassifies instances more frequently:

- Type I error: A total of seven instances, where one was misclassified from healthy to crack and six from healthy to erosion.
- Type II error: A total of 29 instances, where 11 crack faults and 18 erosion faults were misclassified as healthy.
- Type III error: A total of 18 instances, where 11 crack cases were misclassified as erosion and 7 erosion cases were misclassified as crack.

The SVM model's performance is moderate, but it has significant Type II and Type III errors.

Figure 7 presents a comprehensive graphical representation of the XGBoost model's overall performance, providing insights into both the feature importance and classification accuracy. Figure 7a visualizes the feature importance, highlighting the key features that contribute most significantly to the model's predictive power. Meanwhile, Figure 7b displays the confusion matrix of the XGBoost model, offering a detailed breakdown of its classification performance across the three classes: crack, erosion, and healthy. Together, these figures provide a thorough understanding of how the XGBoost model performs and which features most influence its accuracy in detecting faults within these specific categories based on the graph top 5 features that contribute more are mean, shape factor, wind speed, variance, and, fundamental frequency peak.



(**a**) (**b**)

Figure 7. (a) XGBoost feature importance, (**b**) XGBoost confusion matrix.

Figure 8a visualizes the confusion matrix for the random forest model, which demonstrates strong classification performance across the three classes: crack, erosion, and healthy. The model correctly identifies all instances in each class, with no misclassification. Figure 8b displays the confusion matrix for the SVM model. The matrix reveals some misclassifications, particularly between the crack and erosion classes, while the model shows better performance in classifying healthy instances (label 2). These results offer insights into the strengths and limitations of the SVM model when applied to wind turbine blade fault detection.

Figure 9 presents a comprehensive view of the performance of the logistic regression and MLP classifier models, focusing on their classification accuracy. Figure 9a visualizes the confusion matrix for the logistic regression model, showing its performance across the three classes: crack, erosion, and healthy. The model struggles with distinguishing crack and erosion, with notable misclassifications between these two classes. However, it

performs well in classifying the healthy instances, correctly identifying 69 out of 74 cases. Figure 9b displays the confusion matrix for the MLP classifier model. The matrix highlights difficulties in classifying both crack and erosion cases, with many misclassifications between these two categories. However, similar to logistic regression, the model performs better in identifying healthy instances, correctly classifying all 74 cases. These results provide insight into the model's strengths and areas for improvement in fault detection.

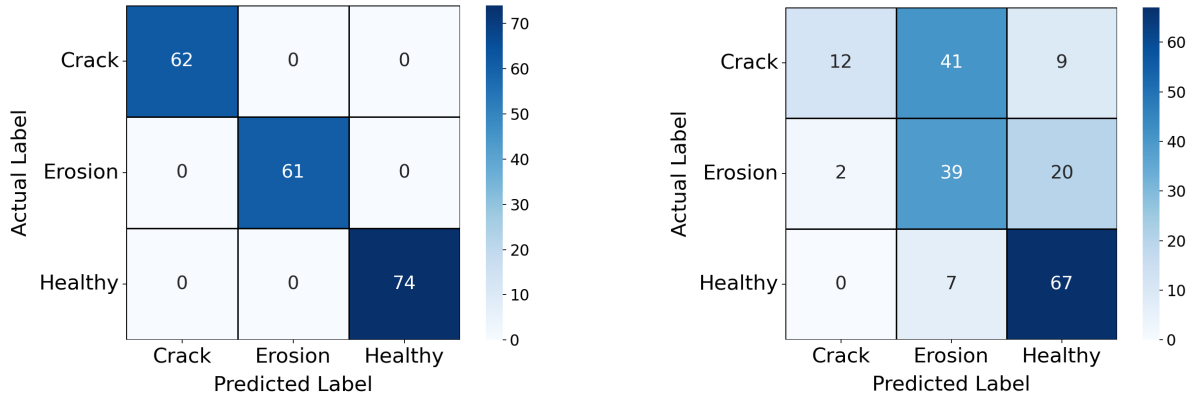


Figure 8. (**a**) Random forest confusion matrix, (**b**) support vector machine confusion matrix.

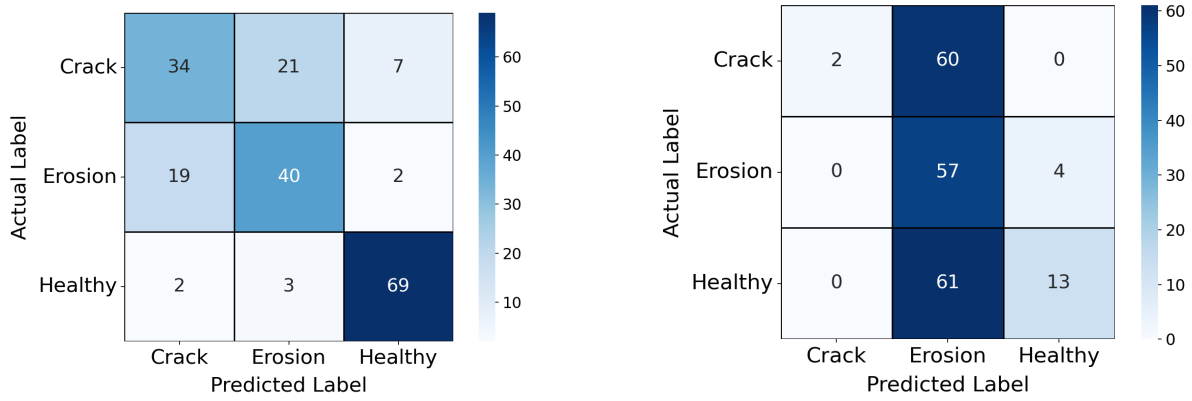


Figure 9. (**a**) Logistic regression confusion matrix, (**b**) MLP classifier confusion matrix.

3.4. Results Related to Error Quantification

Figure 10a illustrates model performance using time-domain features. MLP displays significant Type III errors, with 61 misclassifications between fault types (crack and erosion). Logistic regression (LR) also has a high number of Type III errors (35) but faces additional challenges with Type I (3) and Type II (8) errors, indicating difficulty in identifying true healthy cases and distinguishing faults from healthy instances. Notably, MLP registers the highest Type I errors (35), signaling poor accuracy in recognizing healthy cases. In contrast, XGBoost and random forest show minimal errors across all categories, with only three and five Type III errors, respectively, reflecting strong performance.

Figure 10b shows the results with frequency-domain features. In this configuration, LR exhibits notable Type II errors (67), highlighting a challenge in identifying faulty cases correctly. SVM also shows a considerable number of Type II errors (29), indicating similar struggles with fault detection. MLP demonstrates a moderate number of Type I errors (9) and Type II errors (27), but its Type III errors remain significant at 20, reflecting ongoing challenges in distinguishing between crack and erosion faults. XGBoost and random forest maintain strong performance across all error types, showing minimal misclassifications.

Figure 10c highlights error quantification using combined features. MLP shows the highest Type I errors (61), indicating difficulties in identifying healthy cases accurately. Type III errors are also high for MLP, with 60 misclassifications between crack and erosion faults, reflecting significant challenges in fault type distinction. SVM demonstrates notable Type II errors (29), while LR exhibits moderate Type I (5) and Type III (40) errors, showing room for improvement in both healthy case identification and fault distinction. XGBoost and random forest achieve excellent performance with zero errors across all types, indicating outstanding classification accuracy.

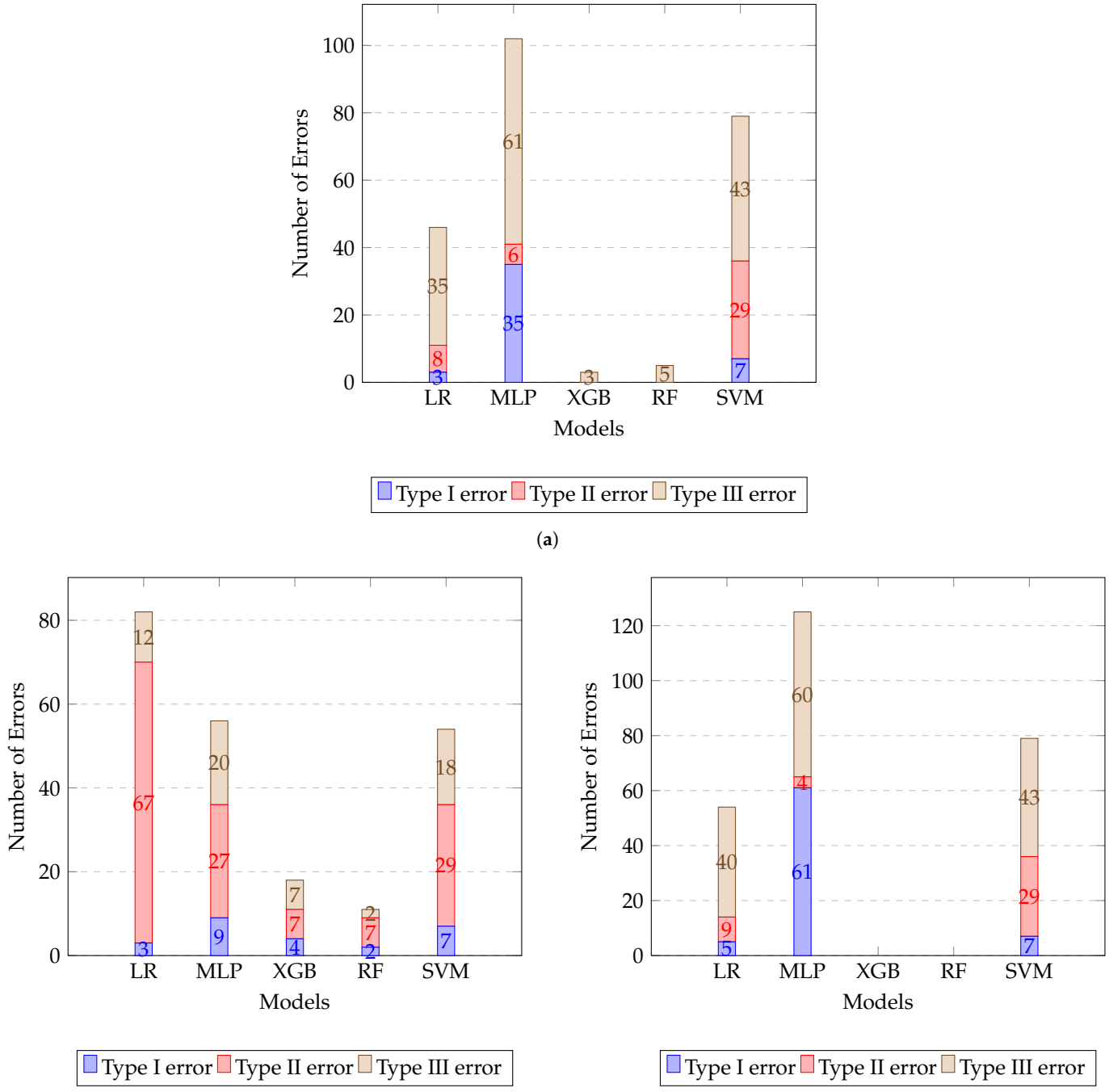


Figure 10. Comparison between Type I, Type II, and Type III errors for the models (time domain, frequency domain, and combined features). (**a**) Error quantification based on time-domain features. (**b**) Error quantification based on frequency-domain features. (**c**) Error quantification based on combined features.

4. Discussion

In this section, five main results are discussed: (1) the observed time-domain features of crack and erosion faults, (2) the observed frequency-domain features of crack and erosion faults, (3) the effect of combining time and frequency features on the performance of different classification analytics, (4) the performances of several classification analytics in detecting crack and erosion faults, (5) the error quantification of several classification analytics.

The obtained time-domain features have managed to capture the amplitude modulation in the vibration time wave produced by the crack fault. For the erosion fault case, the obtained time-domain features have managed to capture asymmetricity and flatness to the vibration time wave produced by the erosion fault. The most relevant time-domain features are the mean, shape factor, variance, peak, and kurtosis. Some of the applied machine learning algorithms, i.e., XGBoost and random forest, based on these relevant time-domain features performed well in terms of accuracy, precision, and error quantification.

The obtained frequency-domain features have managed to capture the sidebands around the fundamental frequency in the frequency domain, produced by the crack fault. For the erosion fault case, the obtained frequency-domain features have managed to capture harmonics produced by the erosion fault. The most relevant frequency-domain features are the amplitudes at the fundamental frequency, sideband, and harmonics. However, the classification performance significantly drops compared to the time-domain feature case, especially for non-ensemble models like logistic regression and MLP classifiers. The XGBoost and random forest algorithms still performed well, although lower than the time features case. The time-domain features performed better than the frequency-domain features.

Combining time- and frequency-domain features led to the best overall performance for several machine learning algorithms. With XGBoost and random forest, the classification accuracy is perfect, as shown in Table 8. This finding is supported by [49], who emphasized that combining time- and frequency-domain analyses significantly improves fault detection accuracy in wind turbines. Another study performed by [11] demonstrates the application of discrete wavelet transform for the same dataset [41], which achieved a classification accuracy of 98% accuracy.

Table 8. Test accuracy comparison.

Algorithm	Accuracy with Only Time-Domain Features	Accuracy with only Frequency Domain Features	Accuracy with Combined Time–Frequency-Domain Features
Logistic regression	0.766	0.584	0.726
MLP	0.482	0.716	0.365
XGBoost	0.985	0.909	1.000
Random Forest	0.975	0.944	1.000
SVM	0.599	0.726	0.599

To check the overfitting issue, Table 9 illustrates the performance of all models using different feature sets. Ensemble models like XGBoost and random forest exhibit minimal signs of overfitting, maintaining high accuracy in both cross-validation and test phases. For example, XGBoost achieves 0.983 cross-validation accuracy and 1.000 test accuracy with combined features, indicating strong generalization. In contrast, the MLP classifier shows significant underfitting, with test accuracy dropping to 0.365 after achieving 0.503 in cross-validation, signaling that it struggles to capture the data complexities. Logistic regression, while maintaining moderate performance across feature sets, experiences a slight accuracy decline from 0.767 in cross-validation to 0.726 in the test set for combined features, suggesting potential underfitting. The consistent performance of SVM (e.g., 0.638 cross-validation and test accuracy for combined features) reflects stable but lower performance compared to ensemble models, implying it may not capture the intricate

patterns in the data as effectively. These trends indicate that ensemble models generally avoid overfitting, while simpler models may struggle with underfitting, especially in more complex feature combinations.

Table 9. Cross-validation accuracy comparison.

Algorithm	Accuracy with Only Time-Domain Features	Accuracy with Only Frequency-Domain Features	Accuracy with Combined Time–Frequency-Domain Features
Logistic Regression	0.767	0.597	0.767
MLP	0.471	0.708	0.503
XGBoost	0.983	0.939	0.976
Random Forest	0.985	0.930	0.980
SVM	0.638	0.715	0.638

The poor performance of models like MLP, logistic regression, and SVM can be attributed to their inability to capture complex interactions between features, especially when relying solely on either time- or frequency-domain features. MLP, being a neural network model, requires a large amount of data and extensive tuning of hyperparameters to generalize well. Without these, it tends to overfit or underfit the data, especially for subtle and complex faults like erosion or cracks, as noted by [6]. Logistic regression, being a linear model, is inherently limited in handling non-linear interactions and is prone to misclassifications when the decision boundaries between classes are not distinct enough, which aligns with findings by [47]. SVM, while capable of handling non-linear decision boundaries through kernel functions, struggles when the feature space is not sufficiently rich, such as when relying solely on time- or frequency-domain features, as reported by [46]. These models lack the flexibility of ensemble methods that can combine multiple weak learners to improve performance.

XGBoost offers more precise control over model complexity, which reduces the likelihood of overfitting while maintaining high accuracy [44]. However, random forest can still be advantageous when computational efficiency and simplicity are priorities, as noted by [45]. In this study, the performance of XGBoost and random forest are almost similar in terms of test accuracy and error quantification. However, for the time-domain feature case, the XGBoost shows better performance in the error quantification, where two failure events and two false alarm events are corrected and predicted, compared to the random forest.

5. Conclusions

This study underscores the essential role of signal processing techniques across time, frequency, and combined time–frequency domains, integrating these features with established machine learning methods to classify faults, such as erosion and cracks, in wind turbine blades.

Time-domain features offer valuable insights but are insufficient to fully capture the complexities of these faults alone. Similarly, frequency-domain features are crucial for analyzing spectral properties, yet limited when used in isolation. A hybrid approach, combining both feature types, significantly enhances model performance, as demonstrated in this study. For instance, XGBoost showed a 1.52% improvement when using combined time and frequency features over time-domain features alone, and an approximate 10% improvement over frequency-domain features alone in this specific data case.

Among the five machine learning models evaluated, the ensemble methods XGBoost and random forest outperformed the others, achieving flawless classification with zero errors. Overall, this study underscores the importance of hybrid feature approaches and advanced ensemble models to achieve high accuracy in fault detection and classification for wind turbine blades.

Author Contributions: Conceptualization, W.A.; methodology, W.A. and I.E.-T.; software, W.A.; validation, W.A. and I.E.-T.; formal analysis, W.A. and I.E.-T.; investigation, W.A. and I.E.-T.; writing—original draft preparation, W.A.; writing—review and editing, W.A., I.E.-T., K.E.T.G. and A.D.; visualization, W.A.; supervision, I.E.-T., K.E.T.G., and A.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article; further inquiries can be directed to the corresponding author.

Acknowledgments: We would like to acknowledge the invaluable contributions of the Renewable Energy Lab in the Mechanical Engineering Department (University of Mustanisiryah, Baghdad, Iraq), who performed the experiments and shared the dataset publicly.

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

Abbreviations

The following abbreviations are used in this manuscript:

MLP	multi-layer perceptron
LR	logistic regression
XGB	XGBoost
RF	random forest
SVM	support vector machine
FFT	fast Fourier transform
RMS	root mean square
THD	total harmonic distortion
ANOVA	analysis of variance
WPETF	wavelet package energy transmissibility function
DWT	discrete wavelet transform
DAQ	data acquisition system

References

1. Doe, J.; Smith, J. Global Wind Energy Statistics. *Renew. Energy J.* **2023**, *45*, 123–134.
2. Hamzehei, S.; Asadi, E. A review of structural health monitoring techniques for wind turbine blades. *Renew. Sustain. Energy Rev.* **2019**, *92*, 1345–1356.
3. Zhu, X.; Chen, F. Review on the research of structural health monitoring for wind turbine blade. *Renew. Energy* **2018**, *116*, 422–432.
4. Pourazarm, S.S.; Gardoni, P. A review of the state-of-the-art and future directions in wind turbine structural health monitoring. *Renew. Energy* **2016**, *99*, 35–45.
5. Garcia, M.; Patel, R. Traditional Methods for Fault Detection in Wind Turbines. *Mech. Syst. Signal Process.* **2020**, *56*, 345–356.
6. Lee, S.; Kim, H. Machine Learning Techniques for Fault Detection. *Artif. Intell. Eng.* **2019**, *22*, 67–78.
7. Stanescu, D.; Digulescu, A.; Ioana, C.; Candel, I. Early-Warning Indicators of Power Cable Weaknesses for Offshore Wind Farms. In Proceedings of the OCEANS 2023—MTS/IEEE U.S. Gulf Coast, Biloxi, MS, USA, 25–28 September 2023; pp. 1–6. [CrossRef]
8. Nandipati, S.; Nichenametla, A.N.; Waghmare, A.L. Cost-Effective Maintenance Plan for Multiple Defect Types in Wind Turbine Blades. In Proceedings of the 2018 Annual Reliability and Maintainability Symposium (RAMS), Reno, NV, USA, 22–25 January 2018; pp. 1–5. [CrossRef]
9. Bocewicz, G.; Banaszak, Z.; Bocewicz, D. Integrated Preventive–Proactive–Reactive Offshore Wind Farms Maintenance Planning. *IEEE Trans. Ind. Appl.* **2024**, 1–10. [CrossRef]
10. Katsaprakakis, D.A.; Papadakis, N.; Ntintakis, I. A comprehensive analysis of wind turbine blade damage. *Energies* **2021**, *14*, 5974. [CrossRef]
11. Ogaili, A.A.; Hamzah, M.N.; Jaber, A.A.; Ghane, E. Application of discrete wavelet transform for condition monitoring and fault detection in wind turbine blades: An experimental study. *Eng. Technol. J.* **2024**, *42*, 104–116. [CrossRef]
12. Wang, X.; Liu, Z.; Zhang, L.; Heath, W.P. Wavelet package energy transmissibility function and its application to wind turbine blade fault detection. *IEEE Trans. Ind. Electron.* **2022**, *69*, 13597–13606. [CrossRef]

13. Rangel-Rodriguez, A.H.; Granados-Lieberman, D.; Amezquita-Sanchez, J.P.; Bueno-Lopez, M.; Valtierra-Rodriguez, M. Analysis of vibration signals based on machine learning for crack detection in a low-power wind turbine. *Entropy* **2023**, *25*, 1188. [CrossRef] [PubMed]
14. Jaikrishna M, A.; V, S.; Dhanraj, J.A.; Velmurugan, K.; Sirisamphanwong, C.; Ngoenmeesri, R.; Sirisamphanwong, C. Transfer learning-based fault detection in wind turbine blades using radar plots and deep learning models. *Energy Sources Part A Recover. Util. Environ. Eff.* **2023**, *45*, 10789–10801. [CrossRef]
15. Sethi, M.R.; Hemasudheer, B.; Sahoo, S.; Kanoongo, S. A Comparative Study on Diagnosing Wind Turbine Blade Fault Conditions using Vibration Data through META Classifiers. In Proceedings of the 2022 4th International Conference on Energy, Power and Environment (ICEPE), Patna, India, 24–25 June 2022; pp. 1–5.
16. Li, X.; Xu, Y.; Peng, Y. Limitations of Traditional Vibration Analysis in Early Fault Detection. *J. Sound Vib.* **2016**, *385*, 48–55.
17. Wang, W.; Wong, A.; Wu, J. Acoustic Emission Techniques for Early Fault Detection in Mechanical Systems: Challenges and Improvements. *IEEE Trans. Reliab.* **2017**, *66*, 805–816. [CrossRef]
18. Zhao, J.; Yan, X.; Jia, M. Critical Review of Ultrasonic Testing in Fault Detection: The Accuracy and Early Detection Dilemma. *Ultrasonics* **2018**, *87*, 21–29.
19. Kim, J.; Kang, D.; Lee, S. Challenges in Vibration-Based Fault Detection for Early Fault Diagnosis. *Mech. Syst. Signal Process.* **2019**, *128*, 782–795.
20. Yang, W.; Tavner, P.J.; Crabtree, C.J.; Feng, Y.; Qiu, Y. Wind turbine condition monitoring: Technical and commercial challenges. *Wind Energy* **2014**, *17*, 673–693. [CrossRef]
21. Tcherniak, D.; Mølgaard, L.L. Active vibration-based SHM system: Demonstration on an operating Vestas V27 wind turbine. In Proceedings of the 8th European Workshop On Structural Health Monitoring (EWSHM 2016), Bilbao, Spain, 5–8 July 2016.
22. Yin, Z.; Han, T.; Zhang, Y. Advances in Vibration Analysis for Early Fault Detection in Rotating Machinery. *IEEE Trans. Ind. Electron.* **2017**, *64*, 7375–7385.
23. Chin, C.; Hameed, Z.; Vasquez, R. Enhancements in Acoustic Emission Techniques for Reliable Early Fault Detection. *IEEE Access* **2018**, *6*, 17417–17429.
24. Gao, R.; Wang, L.; Sun, Z. Improving Accuracy of Ultrasonic Testing for Fault Detection Using Machine Learning. *J. Intell. Manuf.* **2020**, *31*, 1235–1248.
25. Andersen, P.; Thomsen, K.; Pedersen, H. Improving Vibration-Based Fault Detection with Hybrid Approaches. *J. Mech. Sci. Technol.* **2020**, *34*, 2895–2905.
26. Larsen, M.; Nielsen, J. Combining Acoustic Emissions with Machine Learning for Enhanced Fault Detection. *Sensors* **2021**, *21*, 6245.
27. Hong, Y.; Yoo, C.; Kang, J. Vibration Analysis of Wind Turbine Blades Using Fiber Bragg Grating Sensors. *Renew. Energy* **2014**, *71*, 213–217.
28. Fattahi, S.; Zabihollah, A.; Zareie, S. Vibration Monitoring of Wind Turbine Blade Using Fiber Bragg Grating. *Wind Eng.* **2010**, *34*, 721–732. [CrossRef]
29. Nielsen, J.; Andersen, P.; Thomsen, K. Advances in Ultrasonic Testing for Fault Detection: Integration with AI Algorithms. *J. Intell. Manuf.* **2019**, *30*, 1079–1089.
30. Smith, J.; Zhang, L.; Lee, H. Enhanced Acoustic Emission Signal Processing for Fault Detection in Wind Turbine Blades Using Machine Learning Techniques. *Sensors* **2023**, *23*, 4987.
31. Rodriguez, A.; Patel, N.; Choi, J. A Review of Advanced Ultrasonic Techniques for Wind Turbine Blade Fault Detection: Integrating Machine Learning Approaches. *Energies* **2021**, *15*, 5672.
32. Andersen, P.; Thomsen, L. Hybrid Models Combining Vibration Analysis and Machine Learning for Enhanced Fault Detection in Wind Turbines. *Renew. Energy* **2020**, *163*, 1365–1374.
33. Gao, Y.; Li, H.; Chen, Z. Improving Early Fault Detection in Wind Turbine Blades: Ultrasonic Testing Combined with Machine Learning. *J. Renew. Sustain. Energy* **2020**, *12*, 035202.
34. Famoso, L.; Rossi, G.; Chen, M. A Hybrid Model for Fault Detection in Wind Turbine Blades Using Vibration Analysis and Deep Learning Techniques. *Energies* **2024**, *17*, 1627. [CrossRef]
35. Zhou, Y.; Zhang, L.; Chen, G. Early Fault Detection Using Enhanced Vibration Analysis Techniques. *J. Mech. Sci. Technol.* **2021**, *35*, 4035–4046.
36. Sarlak, M.; Farhadi, D. An Intelligent Model Based on Phase Space Analysis for Fault Classification in Single Circuit Transmission Lines. *J. Model. Eng.* **2020**, *18*, 227–243.
37. Mohamad, T.H.; Nazari, F.; Nataraj, C. A Review of Phase Space Topology Methods for Vibration-Based Fault Diagnostics in Nonlinear Systems. *J. Vib. Eng. Technol.* **2020**, *8*, 393–401.
38. Medina, R.; Macancela, J.-C.; Lucero, P.; Cabrera, D.; Cerrada, M.; Sánchez, R.-V.; Vásquez, R.E. Vibration Signal Analysis Using Symbolic Dynamics for Gearbox Fault Diagnosis. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 2195–2214.
39. Garcia-Calva, T.; Morinigo-Sotelo, D.; Fernandez-Cavero, V.; Romero-Troncoso, R. Early Detection of Faults in Induction Motors—A Review. *Energies* **2022**, *15*, 7855.
40. Sharif, A.; Zhai, G.; Jia, J.; Min, X.; Zhu, X.; Zhang, J. An accurate and efficient 1-D barcode detector for medium of deployment in IoT systems. *IEEE Internet Things J.* **2020**, *8*, 889–900. [CrossRef]

41. Ogaili, A.A.F.; Jaber, A.A.; Hamzah, M.N. Wind turbine blades fault diagnosis based on vibration dataset analysis. *Data Brief* **2023**, *49*, 109414. [CrossRef]
42. Harris, F.J. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proc. IEEE* **1978**, *66*, 51–83. [CrossRef]
43. Smith, S.W. *The Scientist and Engineer's Guide to Digital Signal Processing*; California Technical Publishing: San Diego, CA, USA, 1997.
44. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
45. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
46. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
47. Hosmer, D.W.; Lemeshow, S.; Sturdivant, R.X. *Applied Logistic Regression*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
48. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Representations by Back-Propagating Errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
49. Fu, X.; Fang, Y.; Xu, Y.; Xu, H.; Ma, G.; Peng, N. Current Status of Research on Fault Diagnosis Using Machine Learning for Gear Transmission Systems. *Machines* **2024**, *12*, 679. [CrossRef]