applied sciences

MDPI

Article

An Analytical Benchmark of Feature Selection Techniques for Industrial Fault Classification Leveraging Time-Domain Features

Meltem Süpürtülü [1,2,*], Ayşenur Hatipoğlu [1,3] and Ersen Yılmaz [1]

[1] Electrical-Electronic Engineering Department, Bursa Uludag University, Bursa 16059, Türkiye; aysenur.hatipoglu@tai.com.tr (A.H.); ersen@uludag.edu.tr (E.Y.)
[2] TOFAS R&D Center, Turkish Automobile Factory, Bursa 16110, Türkiye
[3] TUSAS Uludağ University R&D Center, Turkish Aerospace Industries, Bursa 16059, Türkiye
* Correspondence: meltem.supurtulu@tofas.com.tr

Abstract: The growing size and complexity of industrial datasets have intensified the need for efficient fault diagnostics tools. This study addresses the challenge of handling large-scale data by developing a data-driven architecture for fault classification in industrial systems. To extract meaningful insights, 15 time-domain features were combined with 5 Feature Selection Methods to optimize model performance by eliminating redundant features. The sensor data and selected features were analyzed using the Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) algorithms to enable accurate fault detection and prediction. The proposed framework was validated using publicly available datasets, namely the Case Western Reserve University (CWRU) bearing dataset and the National Aeronautics and Space Administration Ames Prognostics Center of Excellence (NASA PCoE) lithium-ion battery dataset. The results demonstrate the framework's adaptability and high efficacy across diverse scenarios, achieving an average F1-score exceeding 98.40% using only 10 selected features. This finding highlights the effectiveness of embedded Feature Selection Methods in improving classification performance while reducing computational complexity. The study underscores the potential of the proposed framework as a foundational tool in intelligent manufacturing, offering a versatile solution to enhance fault diagnostics in diverse industrial applications.

Keywords: fault classification; feature selection; long short-term memory; support vector machines; time-domain feature extraction

Academic Editors: Fatemeh Davoudi Kakhki, Maria Kyrarini, Beata Mrugalska and Steven A Freeman

Received: 19 December 2024
Revised: 21 January 2025
Accepted: 23 January 2025
Published: 31 January 2025

Citation: Süpürtülü, M.; Hatipoğlu, A.; Yılmaz, E. An Analytical Benchmark of Feature Selection Techniques for Industrial Fault Classification Leveraging Time-Domain Features. *Appl. Sci.* **2025**, *15*, 1457. https://doi.org/10.3390/app15031457

1. Introduction

Machine Learning (ML) algorithms utilize sensory data acquired throughout the production process as features to identify errors and predict failures before they occur. By implementing ML, reactive maintenance can be replaced by predictive or contingent maintenance, which is more efficient in reducing unnecessary costs and ensuring equipment reliability. However, current diagnostic and monitoring methods are typically task-specific, highlighting the need for advanced techniques to develop a more comprehensive framework to deliver satisfactory performance across diverse applications [1].

In general, the diagnosis and monitoring of industrial systems relies on two main approaches: model-based and data-driven methods. Model-based techniques for fault monitoring use mathematical models based on physical principles to predict system behaviors, making them most effective when system dynamics are well understood. Faults can be diagnosed by monitoring discrepancies between the predicted and actual values [2].

Recently, data-driven techniques have garnered considerable attention, driven by the large volumes of data produced by sensor systems throughout industrial processes [3,4]. Unlike model-based methods, which require detailed system modeling and fatigue life calculations, data-driven techniques can extract hidden patterns from large datasets without domain-specific knowledge, making them highly adaptable and scalable. Typically, sensory signals collected experimentally undergo feature extraction to interpret the raw data. Once features are extracted, traditional classification and regression techniques are applied to make predictions on a case-by-case basis [5].

Feature selection serves a vital function in identifying the most significant features for classification and regression tasks, improving algorithm efficiency during the training phase and model interpretability and generalization, thereby reducing overfitting [6,7]. This process enhances computational efficiency by reducing memory usage and training time, particularly for large datasets, while also making models more interpretable. The presence of redundant features can increase dimensionality, require excessive storage, extend algorithm runtime, and ultimately reduce overall efficiency [8,9]. To mitigate these challenges, selecting the appropriate feature size is essential for developing effective custom diagnostics [10]. In addition, employing cross-validation ensures the robustness of selected features across different data subsets. Depending on the dataset and model, filter, wrapper, or embedded methods can be used to optimize feature selection.

This study proposes a pipeline to enhance classification success by extracting time-domain (TD) features from raw data, such as Minimum (Min), Maximum (Max), Absolute Max (AbsMax), Mean, Standard Deviation (Std), Root Mean Square (RMS), Skewness, Kurtosis, Variance (Var), Peak-to-Peak (P2P), Impulse Factor (IF), Crest Factor (CF), Shape Factor (SF), Mobility (Hjorth Parameters), and Complexity (Hjorth Parameters) [11–15]. Feature Selection Methods (FSM) such as the Fisher Score (FS), Mutual Information (MI), Sequential Feature Selection (SFS), Recursive Feature Elimination (RFE), and Random Forest Importance (RFI) are employed to highlight the most critical features, each contributing to noise reduction and dimensionality optimization. The classification results are analyzed through accuracy, precision, recall, and F1-score, utilizing the SVM algorithm from traditional ML methods and Long Short-Term Memory (LSTM) algorithm from Deep Learning (DL) methods [16–21]. The performance of the proposed fault diagnostic model is validated on the CWRU bearing dataset [22] and the NASA PCoE lithium-ion battery dataset [23].

This research is primarily concerned with comparing various FSMs on datasets drawn from different domains, utilizing both ML and DL techniques. The goal is to reduce model complexity, lower processing loads, and improve performance by eliminating redundant features. The goal is to assess how effectively these methods improve classification performance. The study emphasizes embedded FSM, which directly integrates the feature selection process with the model training. This approach is considered to be more efficient and robust. The study demonstrates that embedded methods can reduce model complexity and improve diagnostic accuracy.

The main contributions of this study are listed below:

1. Comparison of multiple feature selection methods: The study employs a diverse set of FSMs, including FS, MI, SFS, RFE, and RFI. This multi-method approach allows for effective noise reduction, dimensionality optimization, and improved classification accuracy;
2. Comparison of ML and DL algorithms: The classification performances of SVMs, one of the classical ML methods, and LSTM, one of the DL methods, on time series data, were compared;

3. Comprehensive model validation: The performance of the proposed fault diagnostic model is rigorously validated using well-established benchmark datasets, namely the CWRU bearing dataset and the NASA PCoE lithium-ion battery dataset. This ensures the reliability of the model across multiple domains;
4. Emphasis on embedded feature selection methods: The study highlights the advantages of embedded FSMs, such as decision tree-based techniques, which are integrated directly with model training.

The structure of the paper is organized as follows: Section 2 reviews the related work, while Section 3 outlines the materials and methods. Section 4 discusses the model evaluation, and Section 5 provides a discussion of the proposed model. Finally, Section 6 presents the conclusion of the paper.

2. Related Works

Feature extraction is a fundamental step in ML, as it transforms raw signal data into structured information that allows algorithms to detect underlying patterns, ultimately improving accuracy, computational efficiency, and model complexity [24]. When processing signal data, feature extraction spans multiple domains—time, frequency, and time–frequency—each providing unique insights [7]. In [25], nine TD features (Std, Peak, Skewness, Kurtosis, CF, RMS, Clearance Factor (ClF), SF, and IF) were extracted to diagnose faults in bearings. A methodology aimed at enhancing fault diagnosis and prediction in bearings was outlined by [26], focusing on the extraction of 10 TD features (Kurtosis, Skewness, ClF, CF, IF, SF, Peak Factor, RMS, and Std) from vibration signals combined with the application of data fusion techniques. In [27], nine TD features, including Max, Min, Mean, Std, RMS, Skewness, Kurtosis, CF, and Form factor, were extracted from vibration data from the CWRU bearing dataset at a frequency of 48 kHz to capture essential statistical properties for fault classification. Various neural network architectures, including CNN, LSTM, RNN, and Gated Recurrent Unit (GRU), were employed for 10-class classification. Ref. [28] applied feature extraction to identify 10 meaningful TD features, which are Mean, Std, RMS, Peak, SF, CF, IF, Skewness, Kurtosis, and ClF, to identify the meaningful features associated with battery deterioration. In [29], authors developed a real-time health prognostic algorithm by extracting 39 features across multiple signal curves (for 3 curves, separately: Total Harmonic Distortion (THD), RMS, Peak, Std, Mean, Signal-to-Noise Ratio, ClF, IF, CF, Kurtosis, Skewness, SF, Signal-to-Noise (SNR), and Distortion Ratio).

Feature selection serves as a vital step in highlighting the most important features for classification, facilitating the development of well-trained models that deliver superior recognition accuracy. Nevertheless, constructing an optimal classification framework is inherently challenging, as a lack of sufficient features can lead to the failure of numerous classifiers. On the other hand, the presence of redundant features can undermine classification success by impairing the model's skill in discerning valuable patterns in the dataset. This redundancy often gives rise to issues such as increased dimensionality, heightened storage requirements, extended algorithm runtime, and diminished computational efficiency. Addressing these challenges necessitates the implementation of a comprehensive and highly specialized diagnostic framework to ensure precise feature selection [7–9]. In [26], a subset of the 72 extracted features was identified using the Random Forest (RF) algorithm, while 14 dimensionality reduction techniques were employed to determine the most appropriate prognostic health indicator. Ref. [28] used the Extreme Gradient Boosting (XGBoost) algorithm to identify the 15 most important features out of 30 for model training, leveraging its ability to handle large datasets and complex feature interactions. In [29], filter methods were employed to select the top 19 statistically relevant features for predictive modeling, enhancing training efficiency. Ref. [30] prioritized key features using FS and Mahalanobis

distance (MD), with MD being particularly useful for handling correlated features and detecting outliers. Meanwhile, Ref. [31] used the Least Absolute Shrinkage and Selection Operator (LASSO) to rank 46 features, simplifying the model by shrinking irrelevant feature coefficients to 0. In [32], Sequential Feature Selection (SFS), a wrapper method, was used to identify the 10 most important features related to battery degradation, by iteratively evaluating feature combinations based on model performance. The Pearson Correlation was employed to assess the linear relationships between features and degradation, while High Gradient Boosting algorithms were used to capture complex, non-linear interactions. This combination reduced computational complexity and improved predictive accuracy on the NASA battery dataset. The proposed study introduces a comparison of FSMs, focusing on improving fault classification accuracy while simplifying model complexity. It has also been demonstrated that high-performance models can be developed across datasets with different domains, using feature subsets selected by embedded methods.

3. Materials and Methods

The proposed pipeline consists of three main stages: extracting 15 (Min, Max, Abs-Max, Mean, Std, RMS, Skewness, Kurtosis, Var, P2P, IF, CF, SF, Mobility, and Complexity) commonly used TD features, selecting features using various methods, and applying classification techniques. In the feature selection stage, the filter, wrapper, and embedded methods are applied to the extracted features. In the classification stage, the selected features are analyzed using advanced Machine Learning (ML) and Deep Learning (DL) techniques. The general procedure is outlined in Algorithm 1 and Figure 1.

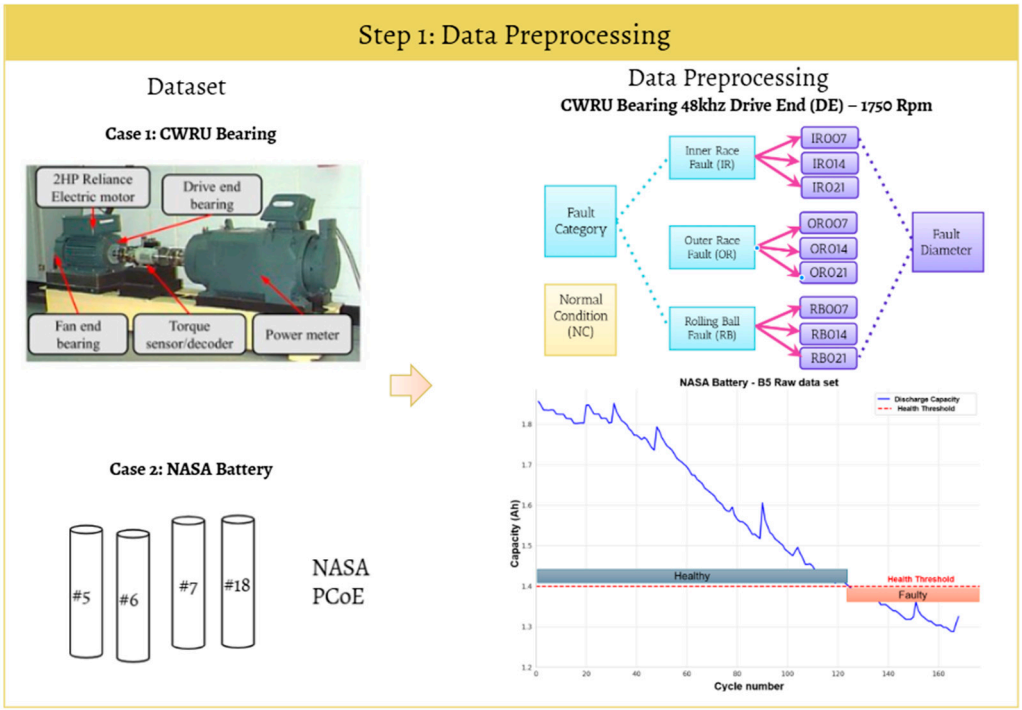


Figure 1. *Cont.*

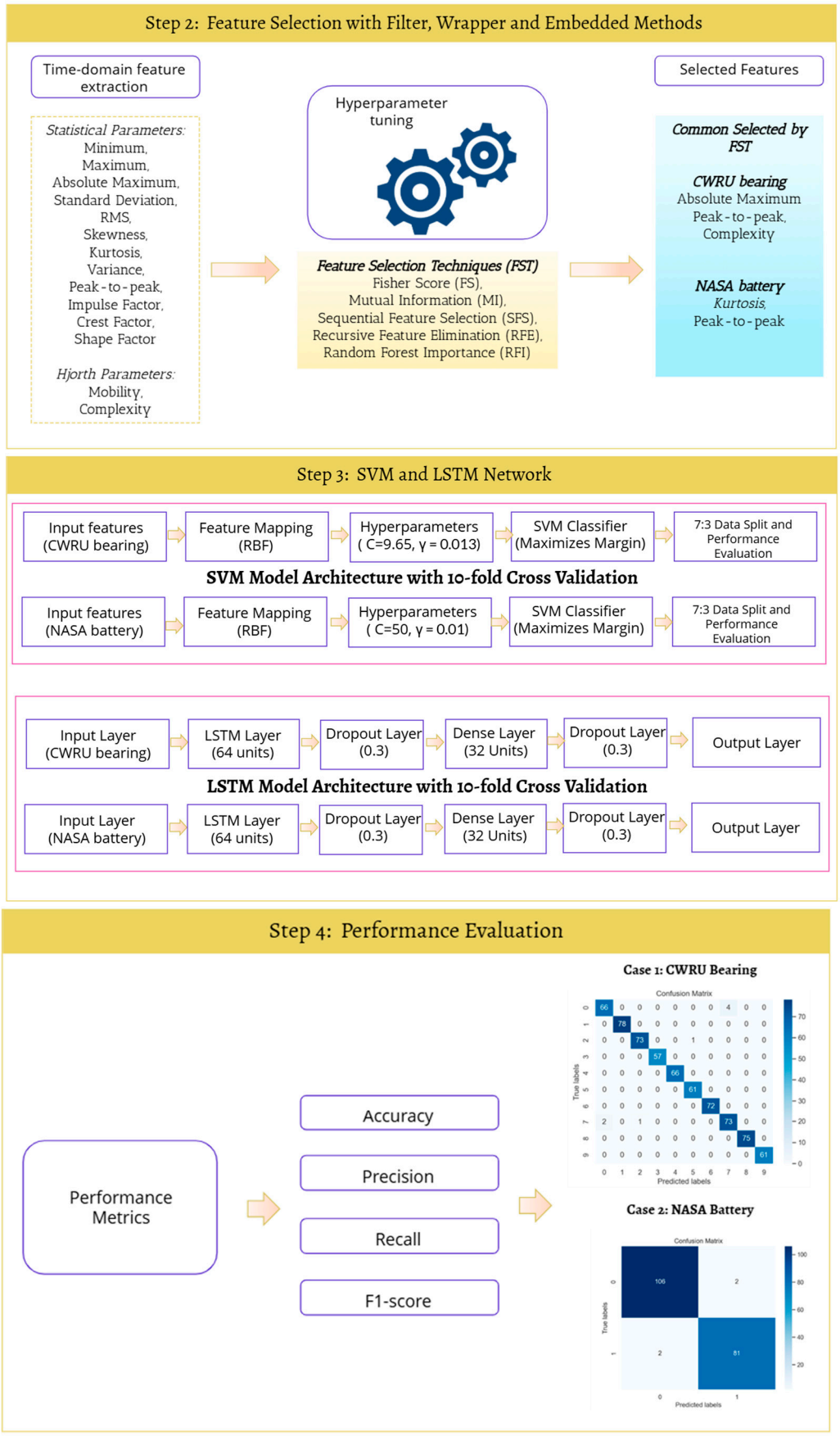


Figure 1. System flowchart.

Algorithm 1: General Procedure for Proposed Approach.

1. Preprocessing
Input: Raw dataset (e.g., time-series data), Sampling rate
Output: Segmented data
a. Clean and normalize the dataset
b. Data labeling and segmentation
2. Feature Extraction
Input: Segmented data
Output: Extracted features
a. Time-domain feature extraction
3. Feature Selection
Input: Time-domain features
Output: Selected features
a. Apply each *Feature Selection* Method
b. Retain top N features based on importance scores
4. Performance Evaluation
Input: Selected features
Output: Model accuracy, precision, recall, and F1-score
a. Split the dataset into training and test sets
b. Train and test the model (SVM, LSTM) using selected features
c. Compare prediction results

3.1. Datasets

Case 1: The CWRU bearing dataset [22] is used as a benchmark in this study, as it provides comprehensive data for fault diagnosis. During the data preprocessing stage, the data were divided into segments for the calculation of time-domain features, due to the large number of data points. The data were segmented into non-overlapping sections of 2048 points each. This segmentation resulted in a total of 230 segments. The data were organized into files containing the respective classes. Time-domain features were extracted from each segment for the entire dataset. Since there are 10 files corresponding to 10 different fault types, the total number of segments is calculated as $230 \times 10 = 2300$. Consequently, the complete dataset has a size of 2048×2300. A detailed summary of the dataset is provided in Table 1 [33].

Table 1. The CWRU bearing dataset (48 kHz Drive End (DE)—1750 Rpm).

Fault Label	Fault Category	Fault Diameter (inches)	Sample Size
0	N	0	230
1	BF	0.007	230
2	BF	0.014	230
3	BF	0.021	230
4	IRF	0.007	230
5	IRF	0.014	230
6	IRF	0.021	230
7	ORF	0.007	230
8	ORF	0.014	230
9	ORF	0.021	230

N: Normal, BF: Ball fault, IRF: Inner race fault, and ORF: Outer race fault.

Case 2: The NASA battery dataset [23] is used as a benchmark to estimate the SoH of lithium-ion batteries, which is crucial for assessing battery performance and lifespan in applications like electric vehicles. This study focuses on four batteries—#5, #6, #7, and

#18—selected for their detailed operational profiles and impedance measurements. The dataset consists of the operational characteristics and impedance readings of 18,650 lithium-ion batteries. Each battery was initially charged with a constant current of 1.5 A until the voltage reached 4.2 V. Afterwards, a constant voltage was applied until the current decreased to below 20 mA, and the battery was discharged at a rate of 2A. The cut-off voltages of the batteries #5, #6, #7 and #18 are given in Table 2. These specific battery selection and operating conditions are employed in this study to gather relevant data to estimate SoH. The end of life (EoL) of a battery is defined as the point when it surpasses a specified limit for either capacity reduction or increased internal resistance, which typically occurs when the remaining capacity drops to between 70 and 80% or when the internal resistance rises to 150–200%, depending on the requirements set by the manufacturer and the specific application. In this work, the SoH classification has selected the End of Life (EoL) Failure Threshold (80%). The tests were stopped when the batteries fulfilled the EoL criteria, which indicated a 20% reduction in nominal capacity [34].

Table 2. The NASA battery dataset (room temperature).

Bat. No.	Discharge		Operating Conditions	
	Constant Current	Cut-Off Voltage	Nominal Capacity	Data Length
#5	2.0 A	2.7 V	2.0 Ah	168 cycles
#6	2.0 A	2.5 V	2.0 Ah	168 cycles
#7	2.0 A	2.2 V	2.0 Ah	168 cycles
#18	2.0 A	2.5 V	2.0 Ah	132 cycles

3.2. Feature Extraction

To enhance the model's accuracy in classifying error categories, new features were derived from the original dataset through time-domain (TD) and statistical analysis [4,7,25–29]. This study used 15 TD features extracted in different combinations, to evaluate the F1-scores of models classifying error categories. Statistical analysis helped to determine the most significant features, ensuring that only relevant data points were used for training. By testing various combinations of these features, the study improved classification accuracy and optimized model performance, as evidenced by higher F1-scores, which balance precision and recall. Table 3 contains the details of the TD features.

Table 3. Details for time-domain features.

No.	Feature Name	Formula	Description
1	Minimum (Min)	$\min(x_i)$	The lowest amplitude observed in the signal.
2	Maximum (Max)	$\max(x_i)$	The highest amplitude recorded in the signal.
3	Absolute Max (AbsMax)	$\max(\lvert x_i \rvert)$	The maximum absolute amplitude attained by the signal.
4	Mean	$x_m = \frac{1}{N}\sum_{i=1}^{N}(x_i)$	The mean amplitude of the signal over its duration.
5	Standard Deviation (Std)	$x_{std} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - x_m)^2}$	This represents a measure of the dataset's distribution, providing insights into its spread and central tendency.

Table 3. *Cont.*

No.	Feature Name	Formula	Description
6	Root Mean Square (RMS)	$x_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}$	This is used to measure the size of changing quantities.
7	Skewness	$x_{skewness} = \frac{\sum_{i=1}^{N}(x_i - x_m)^3}{(N-1){x_{std}}^3}$	This is the symmetry of the distribution of values in a population.
8	Kurtosis	$x_{kurtosis} = \frac{\sum_{i=1}^{N}(x_i - x_m)^4}{(N-1){x_{std}}^4}$	From all the values collected, the gradient of the distribution pattern determines the steepness.
9	Variance (Var)	$x_{var} = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_m)^2$	This shows the degree to which the column numbers vary.
10	Peak-to-Peak (P2P)	$\max(\lvert x_i \rvert) - \min(x_i)$	The amplitude range of the signal is quantified by computing the difference between its maximum and minimum values, providing a measure of the signal's overall variability.
11	Impulse Factor (IF)	$IF = \frac{\max(\lvert x_i \rvert)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}}$	Ratio of AbsMax to RMS.
12	Crest Factor (CF)	$CF = \frac{\max(\lvert x_i \rvert)}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i \rvert}$	The ratio of the average square root of a wave to its peak is known as its RMS value.
13	Shape Factor (SF)	$SF = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i \rvert}$	The ratio of the average value of a wave's amplitude to that wave's average square root value.
14	Mobility (Hjorth Parameters)	$Mobility = \sqrt{\frac{var(x'(t))}{var(x(t))}}$	The square root of the ratio of the signal variances derived from the TD to the signal itself.
15	Complexity (Hjorth Parameters)	$Complexity = \frac{Mobility\ (x'(t))}{Mobility\ (x(t))}$	This refers to the change in frequency by comparing the similarity of the signal to a pure sine wave.

Note: x is the signals series.

3.3. Feature Selection

With the continuous expansion of datasets in scale, diversity, and complexity, reducing data size has become vital to maintaining efficient performance. Features are often divided into four distinct groups: unnecessary, irrelevant, strongly relevant to the target, or weakly relevant but non-redundant [6,7]. Feature selection focuses on building an effective model with reduced computational complexity by prioritizing the most relevant and unique features for classification and prediction. With feature selection, a subgroup of features is determined, with the features obtained using statistical methods used to create a model. In the literature, the FSMs are generally collected into three categories [8,9]. In addition to these methods, hybrid methods have also been developed [35]. Among the widely adopted techniques are the filter method, appreciated for its computational efficiency, and the wrapper method, which, although computationally expensive, is renowned for its reliability [36]. These methods integrate feature selection with modeling techniques to systematically assess features. Conversely, embedded methods merge feature selection and classification within a unified framework, executing both processes concurrently [37]. Feature selection has numerous applications across diverse fields such as engineering (fault detection), climate (weather prediction), health (disease diagnosis), ecology (species distribution modeling), environment (pollution forecasting), finance (fraud detection), tech-

nology (ML optimization), and security (intrusion detection) [38–40]. Its significance lies in its ability to improve estimation accuracy and precision by eliminating irrelevant and redundant data, which in turn reduces model training time and overfitting. Feature selection simplifies models, reducing complexity and memory requirements while ensuring higher performance. By selecting a subset of relevant, noise-free data, the time required for model training decreases, and the model becomes less complex and easier to interpret. Before feature extraction, it is crucial to clean the dataset by removing low-variance variables, handling missing values, and addressing categorical variables with identical or excessive levels [9].

Filter methods for feature selection utilize statistical criteria to assess and rank features according to their relation to the target variable, independently of any learning algorithm [36]. Filter methods serve as a preprocessing step to rank features based on their relevance, which are then selected and revealed by an estimator using various variable sorting techniques [25]. As they are applied before the classification process, they are called filter methods. After ranking features, the top-performing ones are selected for use in model training or further analysis. Filter methods are used as a preprocessing step, independent of the classifier, to reduce data dimensionality by selecting only the most significant features. Filtering methods are known for their simplicity, speed, scalability to high-dimensional data, and efficient computational complexity [37]. Typically, filter methods are divided into two categories: univariate and multivariate [6,7,36]. Examples of common filter methods include Mutual Information, which measures the dependency between variables; Pearson Correlation, which evaluates linear relationships between numerical variables; Chi-Square, which assesses the independence between categorical variables; Information Gain, which calculates feature importance based on entropy reduction; Fisher Score, which assesses the ratio of variance between and within classes; and ANOVA, which tests differences in means across groups. A detailed table of filter methods is provided in [6]. The Pearson Correlation method is used for numeric values, whereas the Chi-Square method is used for categorical values. The following subtitles give information about filter methods.

3.3.1. Mutual Information (MI)

It is a statistical measure used to quantify the level of dependence between two random variables [41]. Unlike other filtering methods, it can also catch non-linear dependencies [37]. Ranking criteria based on information theory assess the interdependence of two variables, with MI represented mathematically as $I(X;Y)$, as shown in Equation (1).

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{1}$$

where the joint probability distribution function of X and Y is represented by $p(x,y)$, while $p(x)$ and $p(y)$ indicate the marginal probability distributions of X and Y, respectively. Moreover, MI can be reformulated using entropy, as shown in Equations (2)–(5).

$$I(X;Y) = H(X) - H(X|Y) \tag{2}$$

$$H(X) = -\sum_{x \in X} p(x) \log(p(x)) \tag{3}$$

$$H(X|Y) = -\sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) \log(p(x|y)) \tag{4}$$

$$H(X,Y) = -\sum_{x,y} p(x,y) \log(p(x,y)) \tag{5}$$

where $H(X)$ is the marginal entropy, $H(X|Y)$ represents the conditional entropy, and $H(X,Y)$ refers to the joint entropy of X and Y. Marginal entropy $H(X)$ quantifies the uncertainty associated with the random variable X, while $H(X|Y)$ measures the residual uncertainty in X after Y is known. This formulation underscores the essence of Mutual Information as the measure of information that one variable provides about the other, capturing both linear and non-linear dependencies [25,41].

3.3.2. Fisher Score (FS)

FS is one of the most commonly utilized techniques for identifying highly discriminative feature subsets. One of the most descriptive features is selected after the ranking of each feature has been calculated [42,43], as follows:

$$F(j) = \frac{\sum_{k=1}^{c} n_k \left(\mu_k^j - \mu_j\right)^2}{\left(\sigma^j\right)^2}, \quad j = 1, 2, \ldots, p \tag{6}$$

$$\sigma^j = \sum_{k=1}^{c} n_k \left(\sigma_k^j\right)^2 \tag{7}$$

In Equations (6) and (7), n_k is the number of classes k, μ_j and σ^j are the Mean and Std relating to feature j, and μ_k^j and σ_k^j are the Mean and Std of class k relating to feature j, respectively. The most discriminative features are chosen from the top m-ranked features with the highest number of scores.

Wrapper methods are advanced Feature Selection Techniques that assess the performance of a Machine Learning (ML) model by systematically evaluating different feature subsets to determine the optimal combination of features [37]. These methods rely on the validation error of the trained model to determine which features to select, with features being iteratively added or removed based on their impact on model accuracy. This method involves evaluating all potential feature subsets to identify the most optimal combination of features. By adding and removing features from the subset, the best set is determined. For large data, this method is not cost-effective in terms of time or computational complexity. Wrapper methods often lead to overfitting. In this study, two wrapper methods are employed: SFS and RFE.

3.3.3. Sequential Forward Selection (SFS)

The SFS algorithm incrementally adds features to a model, aiming to enhance the performance index while minimizing the count of chosen features [25]. The performance index, often based on metrics such as classification accuracy or precision, guides the feature selection process, ensuring the best features are chosen to optimize the model. SFS initiates the process with an empty feature set and progressively incorporates one feature at each step, selecting the feature that maximizes the objective function. This process continues iteratively, with each new feature being evaluated for its contribution to the performance index until the predetermined number of features has been selected. However, SFS is considered a naive approach because it does not account for dependencies between features, potentially leading to suboptimal feature subsets.

3.3.4. Recursive Feature Elimination (RFE)

RFE is an iterative method used for eliminating features until the desired number of features remains. This method recursively removes attributes one by one according to a scoring function. In every iteration, the feature with the lowest score is removed, and the model is reconstructed using the remaining features. This process is repeated until only the most important features are retained. Afterwards, it builds a model for the attributes

that are still present. The features that significantly influence the prediction of the target attribute depend on the model's accuracy [44].

Embedded methods in ML simultaneously handle feature selection and model training. During the training process, they identify the most relevant features to improve model performance, while excluding those that contribute less or are redundant. In embedded methods, ML models are integral to feature selection, as they account for interdependencies among features and the dynamics of input–output variable interaction [45]. Embedded methods offer several advantages, including reducing overfitting and optimizing both the model and the feature selection process simultaneously.

3.3.5. Random Forest Importance (RFI)

Random Forest (RF) is a robust ensemble learning approach that enhances both the accuracy and reliability of a model by integrating the predictions from multiple decision trees, each trained on different data subsets, to generate more robust predictions [46,47]. It uses the bagging (bootstrap aggregation) technique, where multiple samples of the dataset are created with replacement, and each sample is used to train an individual tree. By averaging the predictions of these noisy but unbiased models, RF reduces variance and increases robustness. During the tree-growing process, RF uses the bootstrap technique on the dataset, creating random subsets of input variables to ensure that each tree is de-correlated from the others [9]. This randomness, injected through the selection of random feature subsets for each tree, helps prevent overfitting while maintaining prediction accuracy. RFI ranks features based on their contribution to reducing prediction error by measuring the contribution of each feature to the model's predictive accuracy. This method reduces overfitting compared to traditional decision trees and provides more accurate predictions by leveraging multiple diverse trees. RF chooses m variables at random from p variables. This procedure is driven by the regression predictor, which is mathematically denoted as follows:

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{8}$$

In Equation (8), B represents the total number of trees, $T_b(x)$ is the prediction made by the b-th tree for the input x, and $\hat{f}_{rf}^{B}(x)$, the final prediction made using the Random Forest algorithm, is calculated as the average of the predictions from all B trees. Feature importance is a measure that determines the growing tree split point using RF techniques. As a result, features with a high correlation show little or no measure of importance. RFI employs Out-of-Bag (OOB) samples generated at the base of each tree to assess and record prediction performance [47].

3.4. Machine Learning

3.4.1. Support Vector Machines (SVMs)

SVMs are robust ML algorithms employed for both regression and classification tasks. Considering a dataset of training examples divided into two distinct categories, the SVM training process builds a model capable of assigning new examples to one of the two groups, resulting in a non-probabilistic binary linear classifier. SVMs accomplish this by dividing the data across a hyperplane $f(x) = 0$. Effective classification in SVM models depends on constructing a hyperplane that maintains the maximum margin from the nearest data points, identified as support vectors. There are separate objects with labels that fall into the

positive and negative classifications in the input dataset x_i ($i = 1, 2 \ldots, N$). The hyperplane that splits the provided data is produced by Equation (9) for linear data [48].

$$y = f(x) = W^T x + b = \sum_{i=1}^{N} W_i x_i + b = 0 \quad (9)$$

The vector W has N dimensions, and the scalar b represents a bias term, where N denotes the number of samples in the dataset. The separating hyperplane $f(x) = 0$ is defined by the scalar b and the N-dimensional vector W.

Successful classification with SVMs relies on choosing the hyperplane that maximizes the distance between the hyperplane and the closest data points. A broader margin implies reduced generalization error, where the data points nearest to the hyperplane are instrumental in establishing this margin. Non-linear classification issues can be tackled by applying a kernel function [49]. Kernel functions serve to project data into a higher dimensional feature space, enabling the achievement of linear separability. Frequently utilized kernel functions consist of the linear kernel, polynomial kernel, Radial Basis Function (RBF), and the sigmoid kernel, each designed to handle particular data structures effectively. The kernel trick allows SVMs to compute the separation in this transformed space without explicitly performing the transformation, making the process computationally efficient. For regression tasks, the SVM algorithm is extended as a Support Vector Regression (SVR), which fits data within a margin of tolerance to predict continuous values. Although the SVM algorithm is primarily a binary classifier, multi-class classification can be achieved.

3.4.2. Long Short-Term Memory (LSTM)

The LSTM algorithm is widely recognized for its effectiveness in time series prediction, owing to its capability to capture long-term dependencies critical for time series analysis [50]. LSTM networks employ forget, input, and output gates to manage and control the flow of information throughout the network. The forget gate establishes what past information should be discarded, the input gate updates the cell state with new relevant information, and the output gate controls what information is sent as the final output. The fundamental aspect of LSTM is the cell state, a critical component that acts as a memory pipeline, preserving important information across time steps. These gates rely on activation functions like sigmoid, which figures out how much information should pass through, and tanh, which scales the input. The network is given an input (x_t) and the output produced by the prior hidden layer (h_{t-1}) at time instant t. Equations (10)–(15) show how to calculate an LSTM cell.

$$\text{Forget gate} \quad f_t = \sigma\left[U_f h_{t-1} + V_f x_t + b_f\right] \quad (10)$$

$$\text{Input gate} \quad i_t = \sigma[U_i h_{t-1} + V_i x_t + b_i] \quad (11)$$

$$\text{Input activation} \quad g_t = \tanh\left[U_g h_{t-1} + V_g x_t + b_g\right] \quad (12)$$

$$\text{Cell state} \quad C_t = C_{t-1} \odot f_t + i_t \odot g_t \quad (13)$$

$$\text{Output gate} \quad O_t = \sigma[U_o h_{t-1} + V_o x_t + b_0] \quad (14)$$

$$\text{Hidden state} \quad h_t = O_t \odot \tanh(C_t) \quad (15)$$

where b_f, b_i, b_g, and b_0 represent the biases of the respective gates, and (U_f,V_f), (U_i, V_i), (U_g, V_g), and (U_o, V_o) denote the input weights associated with each gate [50].

This research employed the Adam optimizer alongside an adaptive learning rate mechanism to improve the efficiency of the training process. The optimizer's strategy includes maintaining a moving average of historical gradients and adapting the learning rate using an adaptive momentum mechanism. This technique significantly improves

training performance, especially for intricate models. To tackle overfitting, dropout was employed as a regularization strategy. Overfitting happens when models capture noise and unnecessary details from training data, hindering their ability to perform well on unseen data. By introducing dropout, the model achieved better regularization and improved generalization.

3.5. Performance Metrics

This work employed four widely utilized evaluation metrics in ML to further assess the classification outcomes. These metrics were employed to evaluate the performance of ML algorithms. The assessment metrics used are as follows: (i) accuracy, (ii) precision, (iii) recall, and (iv) F1-score [1]. The formulations are given in the following Equations (16)–(19):

$$accuracy = \frac{(TP + TN)}{(\ TP + TN + FP + FN)} \times 100\% \tag{16}$$

$$precision = \frac{TP}{(\ TP + FP)} \times 100\% \tag{17}$$

$$recall = \frac{TP}{(\ TP + FN)} \times 100\% \tag{18}$$

$$F1 - score = 2 \times \frac{precision \times recall}{(\ precision + recall)} \times 100\% \tag{19}$$

where the abbreviations TP, FP, FN, and TN refer to true positives, false positives, false negatives, and true negatives, respectively. F1-score is often favored over accuracy to address the challenges of model evaluation and selection in scenarios where class distributions are imbalanced, reducing the risk of misleading results. Considering the impact of both false negatives and false positives is an essential step in achieving a comprehensive analysis of error costs in the evaluation process. In this study, the F1-score is emphasized for its ability to provide a comprehensive evaluation that captures all error types, rather than prioritizing a single metric. The F1-score offers a detailed and balanced evaluation by addressing both false negatives and false positives, making it particularly effective for imbalanced datasets. Selecting this metric ensures compatibility with the study's objectives and satisfies its specific evaluation requirements.

4. Model Evaluation

The CWRU bearing dataset and NASA battery dataset are used in this study due to their extensive and well-documented features, making them ideal for TD feature extraction and predictive modeling.

Case 1: The CWRU bearing dataset is used to extract fifteen TD features, which help in identifying potential bearing faults by analyzing vibration signals. The CWRU bearing dataset focuses on mechanical fault detection classification using SVM and LSTM models. The problem is formulated as a fault diagnosis problem, with normal and three main faulty conditions for each faulty diameter. The ranking for FSMs for the CWRU bearing dataset can be seen in Table 4. Features like AbsMax, P2P, and Complexity were consistently selected by multiple methods, highlighting their strong relevance. This feature selection process helps to reduce dataset dimensionality and boosts the capability of Machine Learning (ML) models to accurately classify faults. The graphics of the most frequently selected features are given in Figure 2.

Table 4. Feature selection for the CWRU bearing dataset.

Feature	Feature Selection Methods				
	MI	**FS**	**SFS**	**RFE**	**RFI**
Min	•		•	•	
Max	•	•		•	•
AbsMax	•	•	•	•	•
Mean		•	•		•
Std	•	•			•
RMS	•	•			•
Skewness		•	•	•	
Kurtosis		•	•	•	•
Var	•				•
P2P	•	•	•	•	•
IF			•		
CF			•	•	
SF	•	•		•	
Mobility	•		•	•	•
Complexity	•	•	•	•	•

MI: Mutual Information, FS: Fisher Score, SFS: Sequential Feature Selection, RFE: Recursive Feature Elimination, and RFI: Random Forest Importance. "•" are the selected features by FSMs.

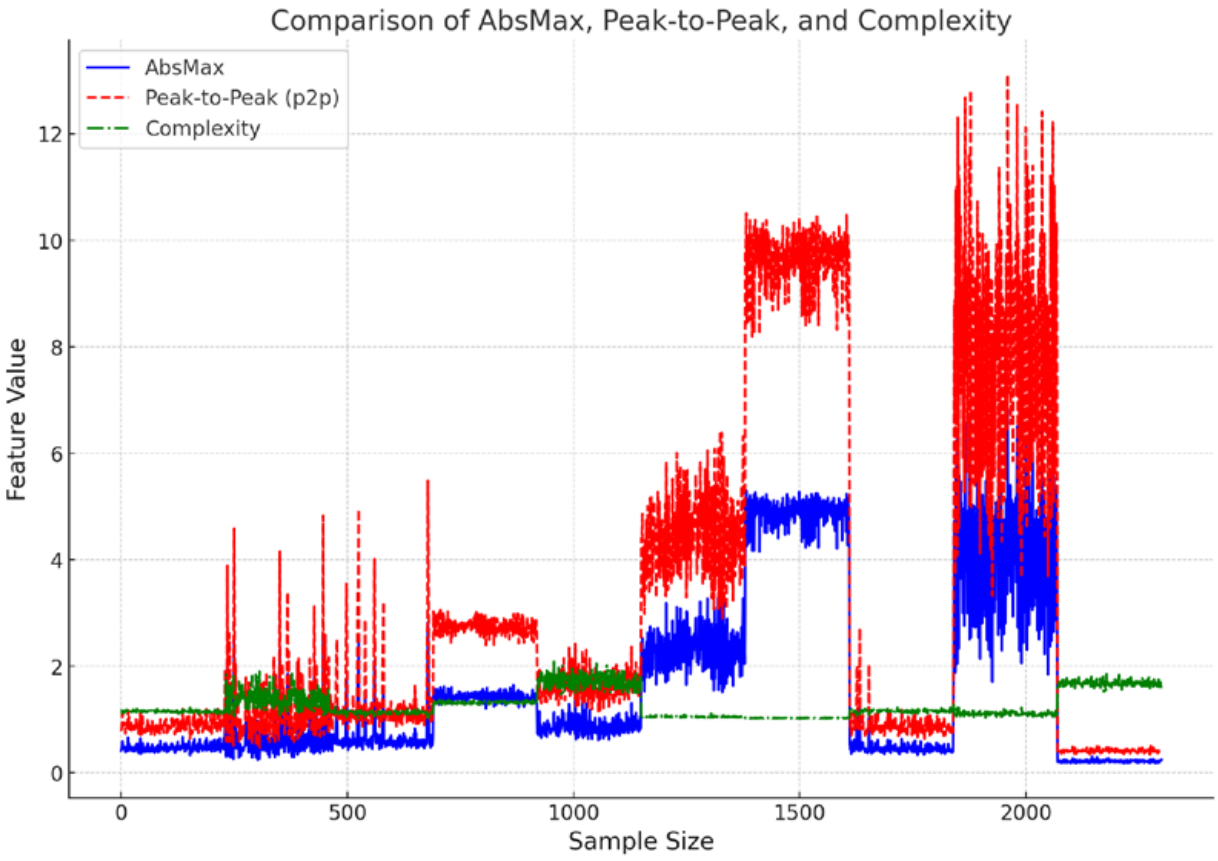


Figure 2. Graphic representations of the Absolute Maximum, Peak-to-Peak, and Complexity features.

In the SVM algorithm, the RBF kernel is applied to manage non-linearly separable data by projecting the input features into a higher dimensional space for improved separability. The penalty parameter c is set to 9.65, balancing margin maximization and classification errors, while γ is set to 0.013, controlling the influence of individual data points for the bearing dataset. A 70–30% training-to-testing split is used to train and evaluate the models of datasets.

The LSTM model for the bearing dataset consists of 1 input LSTM layer with 64 units and a dense layer with 32 units, respectively, and an output layer. For multi-class classification problems, Categorical Cross-Entropy serves as the primary loss function to guide model optimization and SoftMax as an activation function. Two dropout ratios of 0.3 and, after the first dropout layer, one batch normalization layer, is applied for preventing overfitting. Then, 10-fold cross-validation is performed on both algorithms to ensure robust and generalizable performance. The features expressed in bold are the features selected by all FS methods in Table 4.

Table 5 presents the performance evaluation of SVM and LSTM models using different FSMs on the CWRU bearing dataset. The models were evaluated using accuracy, precision, recall, and F1-score, with MI, FS, SFS, RFE, and RFI. The table highlights that both the SVM and LSTM algorithms achieved their best performance when using 10 features selected by RFI-10, with the SVM algorithm achieving an accuracy of 98.12% and F1-score of 98.19% and the LSTM algorithm achieving an accuracy of 98.48% and F1-score of 98.41%. In general, selecting 10 features led to better model performance compared to selecting 5 features. This study shows that careful feature selection can significantly impact the effectiveness of ML models in fault classification, with RFI emerging as the most successful method. The performance metrics in bold show the best results in Table 5.

Table 5. Performance evaluation with selected Feature Selection Methods for CWRU bearing dataset.

Method	Performance Metrics (%)			
	Accuracy	**Precision**	**Recall**	**F1-Score**
SVM-all feature	97.83	98.06	97.94	97.93
SVM-MI-5	85.80	81.92	86.38	83.00
SVM-MI-10	87.97	83.52	88.31	85.10
SVM-FS-5	69.42	70.23	70.67	67.75
SVM-FS-10	92.75	93.91	93.28	93.05
SVM-SFS-5	96.81	97.05	96.95	96.94
SVM-SFS-10	97.97	98.17	98.08	98.07
SVM-RFE-5	88.99	84.30	89.21	85.98
SVM-RFE-10	88.55	88.94	88.89	87.33
SVM-RFI-5	92.61	94.32	93.30	92.96
SVM-RFI-10	**98.12**	**98.23**	**98.22**	**98.19**
LSTM-all feature	98.39	98.32	98.38	98.32
LSTM-MI-5	91.09	91.53	91.20	90.66
LSTM-MI-10	98.30	98.27	98.26	98.23
LSTM-FS-5	85.30	85.74	85.83	84.99
LSTM-FS-10	96.97	96.60	96.65	96.50
LSTM-SFS-5	97.87	97.79	97.88	97.80
LSTM-SFS-10	98.43	98.41	98.41	98.37
LSTM-RFE-5	91.26	90.78	90.98	90.04
LSTM-RFE-10	98.26	98.23	98.25	98.20
LSTM-RFI-5	95.13	95.30	95.16	94.99
LSTM-RFI-10	**98.48**	**98.48**	**98.43**	**98.41**

Case 2: The NASA battery dataset is used to extract 45 TD features from voltage, current, and temperature measurements. The proposed FSMs are evaluated using the SVM and LSTM models. Table 6 displays the feature selection results for the NASA battery dataset. Kurtosis and P2P were consistently selected across multiple methods for both voltage and current, indicating their strong relevance in battery health monitoring. The features expressed in bold are the features selected by all FS methods in Table 6. The graphics of the most frequently selected features are given in Figure 3.

Table 6. Feature selection for NASA battery dataset.

Feature	Voltage					Current					Temperature				
	MI	**FS**	**SFS**	**RFE**	**RFI**	**MI**	**FS**	**SFS**	**RFE**	**RFI**	**MI**	**FS**	**SFS**	**RFE**	**RFI**
Min				●										●	
Max		●	●											●	
AbsMax														●	
Mean	●				●	●				●					
Std								●							
RMS	●				●	●				●					
Skewness						●									
Kurtosis			●	●		●	●	●		●		●		●	●
Var						●		●						●	
P2P				●			●	●				●	●		
IF	●	●			●					●					
CF	●	●			●		●			●		●			
SF						●									
Mobility			●						●						
Complexity			●	●									●		

"●" are the selected features by FSMs.

The penalty parameter c is set to 50 and γ is set to 0.01. The datasets are split into training and testing subsets, with 70% designated for training and the remaining 30% for testing purposes, allowing the models to be trained and subsequently evaluated on separate data. The LSTM model consists of 1 input LSTM layer with 64 units and a dense layer with 32 units, respectively, and an output layer. For multi-class classification problems, the Binary Cross-Entropy is implemented as the loss function, while the sigmoid function is utilized as the activation mechanism within the model, and Adam as the optimizer. A dropout ratio of 0.3 is applied to prevent overfitting. A total of 50 training epochs and 32 batch size are applied, with early stopping for both datasets.

After early stopping, the learning rate scheduler is applied for preventing overfitting. Then, 10-fold cross-validation is applied to both algorithms to ensure the results are robust and generalizable across different data splits.

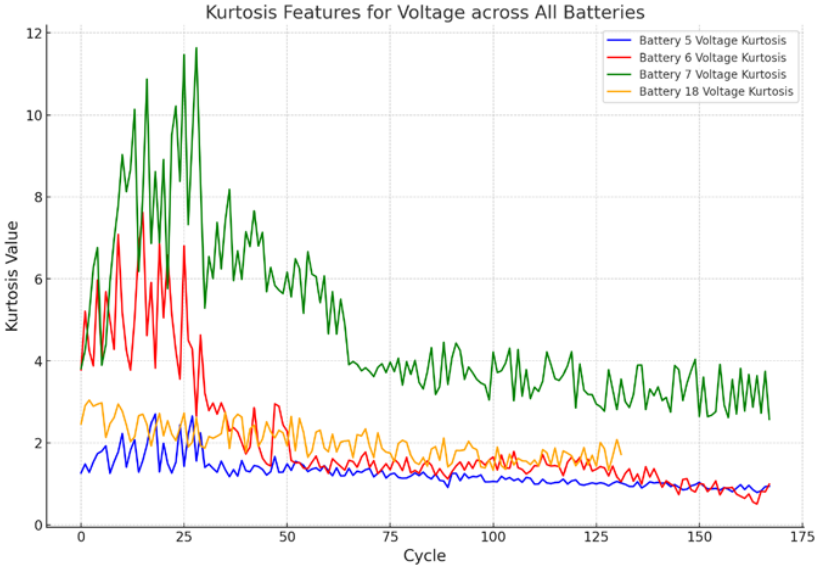
Table 7 presents the performance evaluation of SVM and LSTM models using different FSMs on the NASA PCoE lithium-ion battery dataset. The table highlights that LSTM achieved its best performance when using 10 features selected by RFI-10. The LSTM model achieved an accuracy of 99.06% and an F1-score of 98.80%. The performance metrics in bold show the best results in Table 7.

Table 7. Performance evaluation with selected Feature Selection Methods for NASA battery dataset.

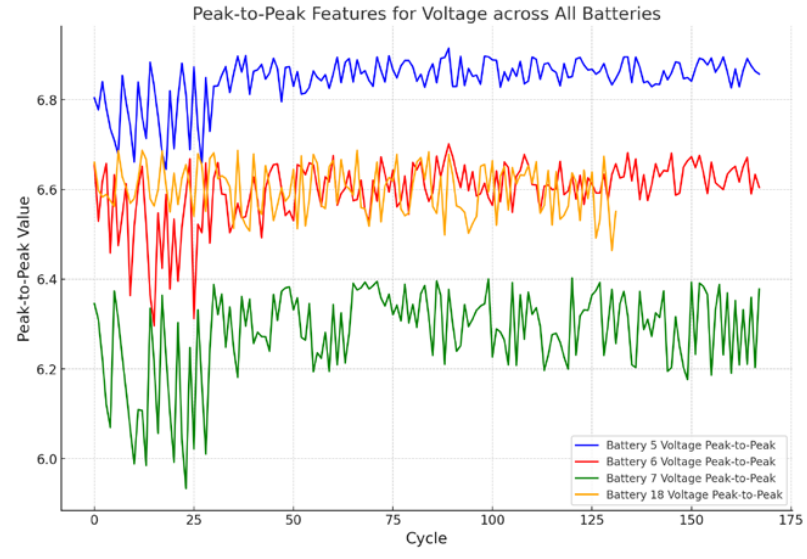
Method	Performance Metrics (%)			
	Accuracy	**Precision**	**Recall**	**F1-Score**
SVM-all feature	96.34	96.14	96.48	96.29
SVM-MI-5	96.86	96.65	97.08	96.82
SVM-MI-10	96.34	96.14	96.48	96.29
SVM-FS-5	95.81	95.58	96.17	95.77
SVM-FS-10	94.76	94.53	95.10	94.71
SVM-SFS-5	88.48	88.50	88.00	88.21
SVM-SFS-10	88.48	88.37	88.14	88.25
SVM-RFE-5	93.19	93.03	93.15	93.09
SVM-RFE-10	96.34	96.22	96.34	96.28

Table 7. *Cont.*

Method	Performance Metrics (%)			
	Accuracy	**Precision**	**Recall**	**F1-Score**
SVM-RFI-5	96.87	96.65	97.02	96.82
SVM-RFI-10	96.34	96.14	96.48	96.29
LSTM-all feature	98.59	98.04	98.64	98.30
LSTM-MI-5	98.75	97.47	99.70	98.54
LSTM-MI-10	98.11	97.47	98.19	97.76
LSTM-FS-5	98.27	97.72	98.19	97.88
LSTM-FS-10	98.90	98.44	99.06	98.72
LSTM-SFS-5	94.50	93.99	93.35	93.55
LSTM-SFS-10	98.11	97.76	97.92	97.78
LSTM-RFE-5	97.32	96.74	97.32	96.99
LSTM-RFE-10	98.43	97.02	99.31	98.11
LSTM-RFI-5	98.27	97.74	98.15	97.89
LSTM-RFI-10	**99.06**	**99.13**	**98.54**	**98.80**



(**a**)



(**b**)

Figure 3. Graphic representations of the selected features (**a**) Kurtosis and (**b**) P2P.

5. Discussion

Case 1: This study demonstrates that utilizing 10 features (Mobility, Standard, Variance, Mean, RMS, Complexity, P2P, Kurtosis, Max, and AbsMax) selected by RFI is sufficient to attain 98.19% and 98.41% F1-scores in the CWRU bearing dataset for the SVM and LSTM models, respectively, as indicated in Table 8 and Figure 4. Using all features, 98.39% accuracy and 98.32% F1-scores were obtained with the LSTM model. It was observed that the performance increased after feature selection.

Table 8. Feature selection results for all datasets.

Method	Observation	Dataset 1 (CWRU Bearing)	Dataset 2 (NASA Battery)
SVM	No. of features	15	45
	No. of feature selection	10	10
	F1-score in % (with total features)	97.93	96.29
	F1-score in % (with selected features)	98.19	96.82
	Feature Selection Method	RFI	RFI, MI
LSTM	No. of features	15	45
	No. of feature selection	10	10
	F1-score in % (with total features)	98.32	98.30
	F1-score in % (with selected features)	98.41	98.80
	Feature Selection Method	RFI	RFI

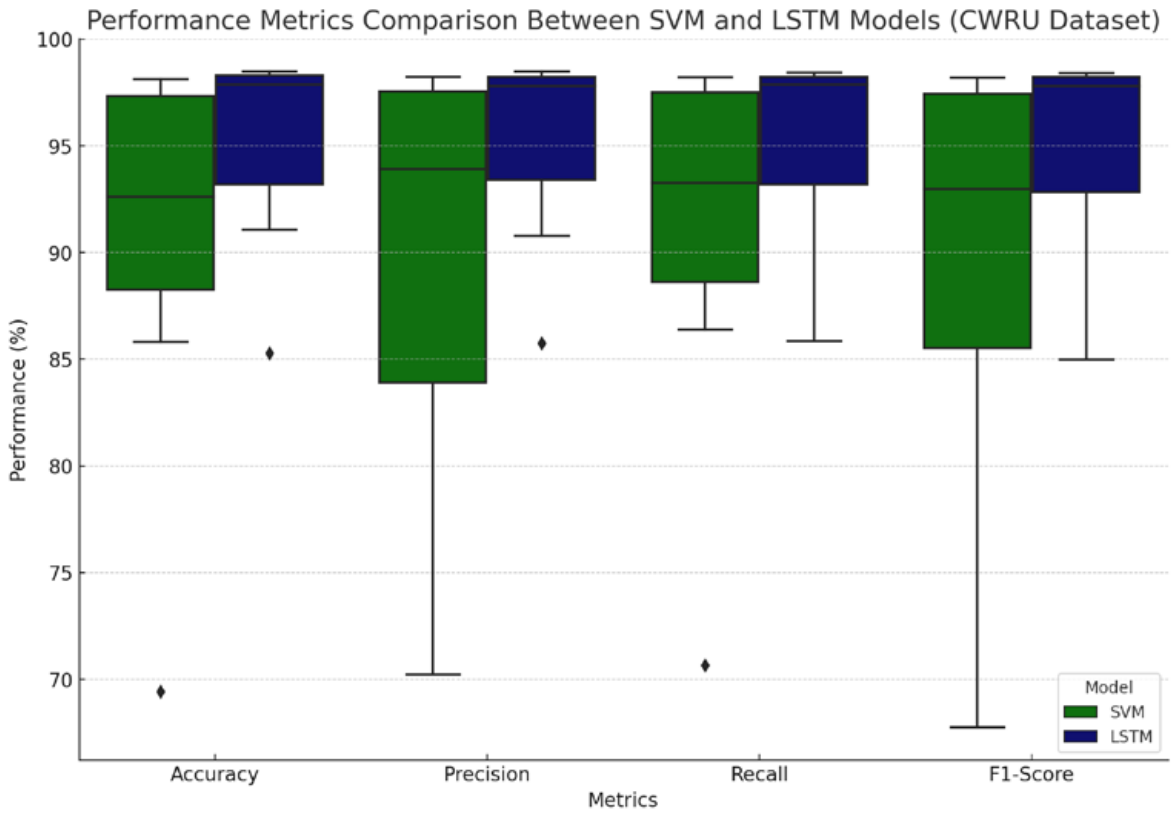


Figure 4. Boxplot comparison of SVM and LSTM performance metrics for CWRU bearing dataset.

Ref. [4] utilized 2800 samples distributed across 10 classes, analyzed using 5-fold cross-validation, whereas our research employed 2300 samples obtained from 230 windows across the same 10 classes, using a 10-fold cross-validation approach. This study also extracted 14 unique TD and statistical features, achieving 97% accuracy with the RF method and 95.27% accuracy when utilizing raw data with CNN. After incorporating a filter and a sliding window technique, the CNN demonstrated substantial performance improvement, achieving an accuracy rate of 98.50%. Furthermore, the sensitivity, recall, and F1-scores of the model were recorded as 98.60%, 98.57%, and 98.57%, respectively.

In ref. [27], Max, Min, Mean, Standard, RMS, Skewness, Kurtosis, Crest Factor, and Shape Factor features are extracted from the 48 kHz CWRU bearing dataset. Using CNN, LSTM, RNN, Gated Recurrent Unit (GRU), and Bidirectional LSTM methods, CNN and LSTM models achieved 95.00% accuracy, RNN and GRU models achieved 97.00% accuracy, and the Bidirectional LSTM model achieved 96.00% accuracy.

In another study presented in the literature, 12 kHz DE, 48 kHz DE, and Fan End (FE) data of the CWRU bearing dataset were used [51]. The 48 kHz DE dataset results of this study were compared with our current study. The shaft speed rpm information of the data is not shared. The dataset includes 10 classes of data, each consisting of 50 samples. Each sample contains 2048 sample points. A total of 27-time, frequency, and entropy energy features were obtained. Using the ReliefF Feature Selection Method, the best 11 features were determined. These features are Kurtosis, P2P, Var, SF, Margin Factor, RMS Frequency (RMSF), Frequency Variance (VF), Root Frequency Variance (RVF), Power Spectrum H1, Power Spectrum H2, and Intrinsic Mode Function6 (IMF6) E6. Methods such as the Adaboost, SVM, GDBT, KNN, Gray Relational Degree (GRD), Bagging, CNN, and LSTM methods were employed. The performance, evaluated using the accuracy metric, is depicted in the accompanying graph. While the results are not presented explicitly as percentage values, the outcomes of the SVM and LSTM algorithms were found to be similar to those of our approach. The literature comparison is given in Table 9.

Table 9. The CWRU bearing 48 kHz DE dataset literature results.

Method	Feature Extraction and Selection Technique	Performance Metrics (%)
[4]	14 TD and statistical features with RF	Accuracy: 95.27% (raw data) 98.57% (all features selected)
[27]	9 TD features without selection	CNN and LSTM Accuracy: 95.00% GRU and RNN Accuracy: 97.00% Bi-LSTM Accuracy: 98.00%
[51]	27 features obtained. 11 features selected with ReliefF	SVM Accuracy: ~98.00% LSTM Accuracy: ~92.00%
Proposed method	SVM-RFI: 10 TD features	Accuracy: 98.12% *
	LSTM-RFI: 10 TD features	Accuracy: 98.48% *

* The table gives the accuracy results for comparison with the literature.

This work aims to address the problem in a simpler and more efficient way. Our aim is to present a method that achieves high performance in the prediction process by determining an efficient feature selection process.

Case 2: This study demonstrates that utilizing 10 features (Voltage Mean, Voltage RMS, Voltage IF, Voltage CF, Current Mean, Current RMS, Current IF, Current CF, Temperature, and Kurtosis) selected by RFI is sufficient to attain 96.82% and 98.80% F1-scores in the

NASA battery dataset for SVM and LSTM, respectively, as indicated in Table 8 and Figure 5. Using all features, 98.59% accuracy and 98.30% F1-scores were obtained with the LSTM model. It was observed that the performance increased after feature selection.

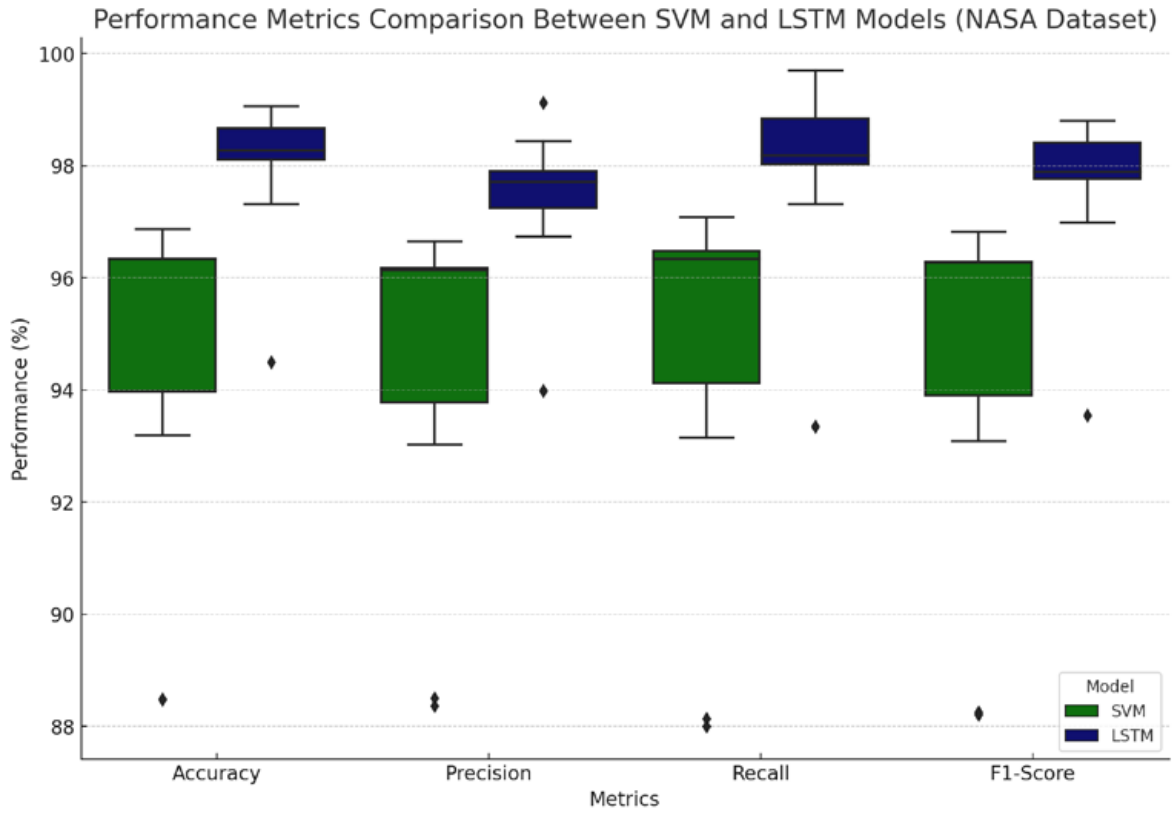


Figure 5. Boxplot comparison of SVM and LSTM performance metrics for NASA battery dataset.

While SoH estimation is typically approached as a regression problem in the literature for the NASA battery dataset, this study treats it as a classification problem, offering a different perspective on battery health estimation. Regression methods estimate the SoH as a continuous variable, providing precise numerical values. Conversely, classification approaches discretize this continuous spectrum into distinct categories or classes. By categorizing SoH into discrete levels, classification facilitates a more intuitive interpretation of battery performance, aligning it with specific operational or performance thresholds. This categorical approach can enhance decision-making processes in certain applications by prioritizing interpretability and expediting actions based on broader performance categories rather than granular values.

Moreover, classification methods often yield more robust and stable results within defined ranges, as they focus on distinguishing between broader categories rather than striving for precise predictions of continuous SoH values, which may be subject to noise or measurement inaccuracies. For instance, as shown in Table 10 of Ref. [52], SoH was categorized into three discrete classes. However, since this analysis utilized the Oxford dataset, it was not directly compared in the discussion section, to ensure consistency in the comparative evaluation.

Ref. [53] utilized a two-class SoH classification approach with an optimized SVM model on the NASA dataset, reporting an accuracy of 90.48%, recall of 77.47%, and precision of 100.00%. In contrast, the proposed method demonstrates significant improvements by incorporating both feature extraction and optimization processes. Specifically, the use of 15 TD features, later refined to 10 TD features using RFI, combined with LSTM, yielded an accuracy of 99.06%, recall of 99.13%, and precision of 98.54%.

Table 10. The NASA battery dataset (room temperature) literature results.

Method	SoH Classification Technique	Features	Model	Performance Metrics (%)
[53]	Two-class	NA	Optimized SVM without feature extraction and selection	Accuracy: 90.48% Recall: 77.47% Precision: 100.00%
Proposed method	Two-class	15 TD	SVM-RFI: 10 TD features	Accuracy: 96.34% Recall: 96.14% Precision: 96.48%
			LSTM-RFI: 10 TD features	Accuracy: 99.06% Recall: 99.13% Precision: 98.54%

In ref. [54], a two-class SoH classification approach was implemented using an LSTM model on a proprietary dataset. The study reported an accuracy of 88%, recall of 91%, and precision of 95%. This demonstrates the capability of LSTM models to effectively capture time-series dependencies and deliver high performance in battery health classification tasks. The use of a proprietary dataset highlights the importance of tailored data collection and preprocessing in achieving robust and reliable results for SoH classification. These findings further validate the utility of Deep Learning techniques, such as LSTM, in advancing battery management systems by providing accurate and actionable insights. These findings underscore the effectiveness of the proposed methodology, which leverages feature optimization to enhance performance while maintaining computational efficiency.

From the perspective of computational complexity, reducing the number of input features from 15 to 10 across both datasets significantly decreases the dimensionality of the LSTM input matrix. This reduction directly alleviates the computational burden by minimizing the number of matrix multiplications and weight update operations—components that are among the most computationally demanding during the training process. Moreover, the streamlined feature space results in a lower kernel computation overhead, thereby improving the overall efficiency of the model. This reduction in complexity not only optimizes resource utilization, but also facilitates faster convergence, as the model processes fewer parameters while maintaining its capacity to capture essential patterns. Importantly, this approach ensures that the integrity and robustness of the analysis are not compromised, achieving a balance between computational efficiency and predictive performance.

The proposed approach, while effective in many contexts, is not without limitations that merit acknowledgment. Firstly, the method may struggle to adequately capture periodic or harmonic features within the data, potentially resulting in the loss of critical information in scenarios where such patterns are significant. Secondly, its capacity to model long-term dependencies or time-varying patterns in signals may be limited, restricting its applicability in cases requiring a thorough understanding of complex temporal dynamics. These limitations underscore the need for future research aimed at refining the methodology, particularly through the incorporation of advanced techniques designed to enhance the modeling of periodicity and long-term dependencies.

The works presented under the related works heading generally utilize methods that involve complex or computationally intensive feature extraction techniques. This work aims to address the problem in a simpler and more efficient way. Our aim is to present a method that achieves high performance in the prediction process by determining an efficient feature selection process.

6. Conclusions

This work aims to improve fault classification on the CWRU bearing dataset and SoH detection on the NASA battery dataset by applying TD feature extraction and FSMs. A total of 15 features for the bearing dataset and 45 for the battery dataset were extracted, capturing key statistical and signaling features. Five FSMs, namely MI, FS, SFS, RFE, and RFI, were applied to determine the optimal feature sets. The performance of these methods was evaluated using SVM and LSTM techniques. The experimental study shows that RFI, an embedded Feature Selection Method, can effectively identify important features by reducing the number of redundant features, achieving an F1-score of 98.41% with 10 features out of 15 for the bearing dataset and 98.80% with 10 features out of 45 for the battery dataset. FSMs improve classification accuracy and reduce computational complexity by focusing on compelling features. In addition to these findings, the experimental study demonstrates that the P2P feature is a highly discriminative characteristic, consistently selected across both datasets. RFI, an embedded method, integrates feature selection directly into the model training process and has been noted for its strong performance.

Author Contributions: Conceptualization, M.S. and A.H.; methodology, M.S., A.H. and E.Y.; software, M.S. and A.H.; validation, M.S., A.H. and E.Y.; writing—review and editing, M.S. and A.H.; supervision, E.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under project numbers 118C083 (M.S.) and 118C100 (A.H.), within the scope of the 2244 Industrial Ph.D. Program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used to support the findings of this study is available upon request. The experimental data of the study use the bearing datasets from the Case Western Reserve University (CWRU) (https://engineering.case.edu/bearingdatacenter/apparatus-and-procedures (accessed on 14 July 2022)) and the NASA AMES Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository (accessed on 14 July 2022)).

Conflicts of Interest: Author Meltem Süpürtülü was employed by the Turkish Automobile Factory Inc. (TOFAS). Author Ayşenur Hatipoğlu was employed by the Turkish Aerospace Industries Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Yuan, Y.; Ma, G.; Cheng, C.; Zhou, B.; Zhao, H.; Zhang, H.T.; Ding, H. A general end-to-end diagnosis framework for manufacturing systems. *Nat. Sci. Rev.* **2020**, *7*, 418–429. [CrossRef] [PubMed]
2. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [CrossRef]
3. Hendriks, J.; Dumond, P.; Knox, D.A. Towards better benchmarking using the CWRU bearing fault dataset. *Mech. Syst. Signal Process.* **2022**, *169*, 108732. [CrossRef]
4. Magar, R.; Ghule, L.; Li, J.; Zhao, Y.; Farimani, A.B. FaultNet: A Deep Convolutional Neural Network for Bearing Fault Classification. *IEEE Access* **2021**, *9*, 25189–25199. [CrossRef]
5. Udmale, S.S.; Nath, A.G.; Singh, S.K. Application of Industry 4.0 and Meta Learning for Bearing Fault Classification. In Proceedings of the IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hangzhou, China, 4–6 May 2022; pp. 1455–1460. [CrossRef]
6. Jović, A.; Brkić, K.; Bogunović, N. A review of feature selection methods with applications. In Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1200–1205. [CrossRef]
7. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [CrossRef]

8. Venkatesh, B.; Anuradha, A. A Review of Feature Selection and Its Methods. *Cybern. Inf. Technol.* **2019**, *19*, 3–26. [CrossRef]
9. Kuhn, M.; Johnson, K. *Feature Engineering and Selection: A Practical Approach for Predictive Models*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2019; pp. 227–240. [CrossRef]
10. Lee, C.Y.; Le, T.A.; Lin, Y.T. A Feature Selection Approach Hybrid Grey Wolf and Heap-Based Optimizer Applied in Bearing Fault Diagnosis. *IEEE Access* **2022**, *10*, 56691–56705. [CrossRef]
11. Aburakhia, S.A.; Myers, R.; Shami, A. A Hybrid Method for Condition Monitoring and Fault Diagnosis of Rolling Bearings with Low System Delay. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3519913. [CrossRef]
12. Carbonó dela Rosa, M.E.; Velasco Herrera, G.; Nava, R.; Quiroga González, E.; Sosa Echeverría, R.; Sánchez Álvarez, P.; Gandarilla Ibarra, J.; Velasco Herrera, V.M. A New Methodology for Early Detection of Failures in Lithium-Ion Batteries. *Energies* **2023**, *16*, 1073. [CrossRef]
13. Miao, Y.; Wang, J.; Zhang, B.; Li, H. Practical framework of Gini index in the application of machinery fault feature extraction. *Mech. Syst. Signal Process.* **2022**, *165*, 108333. [CrossRef]
14. Lei, Y.; He, Z.; Zi, Y. Application of an intelligent classification method to mechanical fault diagnosis. *Expert Syst. Appl.* **2009**, *36*, 9941–9948. [CrossRef]
15. Tang, Z.; Wang, M.; Ouyang, T.; Che, F. A wind turbine bearing fault diagnosis method based on fused depth features in time–frequency domain. *Energy Rep.* **2022**, *8*, 12727–12739. [CrossRef]
16. Rojas, A.; Nandi, A.K. Detection and Classification of Rolling-Element Bearing Faults using Support Vector Machines. In Proceedings of the IEEE Workshop on Machine Learning for Signal Processing, Mystic, CT, USA, 28–30 September 2005; pp. 153–158. [CrossRef]
17. Konar, P.; Chattopadhyay, P. Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs). *Appl. Soft Comput.* **2011**, *11*, 4203–4211. [CrossRef]
18. Nuhic, A.; Terzimehic, T.; Soczka-Guth, T.; Buchholz, M.; Dietmayer, K. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *J. Power Sources* **2013**, *239*, 680–688. [CrossRef]
19. Zhang, S.; Zhang, S.; Wang, B.; Habetler, T.G. Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review. *IEEE Access* **2020**, *8*, 29857–29881. [CrossRef]
20. Francos, D.F.; Rego, D.M.; Romero, O.F.; Betanzos, A.A. Automatic bearing fault diagnosis based on one-class ν-SVM. *Comput. Ind. Eng.* **2013**, *64*, 357–365. [CrossRef]
21. Khaleghi, S.; Beheshti, S.H.; Berecibar, M.; Van Mierlo, J. A data-driven method based on recurrent neural network method for online capacity estimation of lithium-ion batteries. In Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC), Gijon, Spain, 18 November–16 December 2020; pp. 1–7. [CrossRef]
22. Case Western Reserve University (CWRU) Bearing Data Center. Available online: https://engineering.case.edu/bearingdatacenter/apparatus-and-procedures (accessed on 14 July 2022).
23. Saha, B.; Goebel, K. Battery Data Set. NASA Ames Prognostics Data Repository. Available online: http://ti.arc.nasa.gov/project/prognostic-data-repository (accessed on 14 July 2022).
24. Gao, T.; Yang, J.; Wang, W.; Fan, X. A domain feature decoupling network for rotating machinery fault diagnosis under unseen operating conditions. *Reliab. Eng. Syst. Saf.* **2024**, *252*, 110449. [CrossRef]
25. Van, M.; Kang, H.J. Bearing-fault diagnosis using non-local means algorithm and empirical mode decomposition-based feature extraction and two-stage feature selection. *IET Sci. Meas. Technol.* **2015**, *9*, 671–680. [CrossRef]
26. Buchaiah, S.; Shakya, P. Bearing fault diagnosis and prognosis using data fusion-based feature extraction and feature selection. *Measurement* **2022**, *188*, 110506. [CrossRef]
27. Althobiani, F. A Novel Framework for Robust Bearing Fault Diagnosis: Preprocessing, Model Selection, and Performance Evaluation. *IEEE Access* **2024**, *12*, 59018–59036. [CrossRef]
28. Ardeshiri, R.R.; Liu, M.; Ma, C. Multivariate stacked bidirectional long short-term memory for lithium-ion battery health management. *Reliab. Eng. Syst. Saf.* **2022**, *224*, 108481. [CrossRef]
29. Ang, E.Y.M.; Paw, Y.C. Efficient linear predictive model with short-term features for lithium-ion batteries state of health estimation. *J. Energy Storage* **2021**, *44*, 103409. [CrossRef]
30. Wu, S.D.; Wu, C.W.; Wu, T.Y.; Wang, C.C. Multi-Scale Analysis Based Ball Bearing Defect Diagnostics Using Mahalanobis Distance and Support Vector Machine. *Entropy* **2013**, *15*, 416–433. [CrossRef]
31. Dave, V.; Thakker, H.; Vakharia, V. Fault identification of ball bearings using Fast Walsh Hadamard Transform, LASSO feature selection, and Random Forest classifier. *FME Trans.* **2022**, *50*, 202–209. [CrossRef]
32. Jorge, I.; Mesbahi, T.; Samet, A.; Boné, R. Time Series Feature Extraction for Lithium-Ion Batteries State-Of-Health Prediction. *J. Energy Storage* **2023**, *59*, 106436. [CrossRef]
33. Wu, K.; Tao, J.; Yang, D.; Xie, H.; Li, Z. A Rolling Bearing Fault Diagnosis Method Based on Enhanced Integrated Filter Network. *Machines* **2022**, *10*, 481. [CrossRef]

34. Kupper, C.; Weißhar, B.; Rißmann, S.; Bessler, W.G. End-of-life prediction of a lithium-ion battery cell based on mechanistic aging models of the graphite electrode. *J. Electrochem. Soc.* **2018**, *165*, A3468–A3480. [CrossRef]
35. Elavarasan, D.; Vincent, P.M.D.R.; Srinivasan, K.; Chang, C.Y. A Hybrid CFS Filter and RF-RFE Wrapper-Based Feature Extraction for Enhanced Agricultural Crop Yield Prediction Modeling. *Agriculture* **2020**, *10*, 400. [CrossRef]
36. Tiwari, A.K. Feature Selection Algorithms for Classification and Clustering. In *Cognitive Analytics*; IGI Global: Hershey, PA, USA, 2020. [CrossRef]
37. Beraha, M.; Metelli, A.M.; Papini, M.; Tirinzoni, A.; Restelli, M. Feature Selection via Mutual Information: New Theoretical Insights. *arXiv* **2019**, arXiv:1907.07384v1. [CrossRef]
38. Chen, R.C.; Dewi, C.; Huang, S.W.; Caraka, R.E. Selecting critical features for data classification based on machine learning methods. *J. Big Data* **2020**, *7*, 52. [CrossRef]
39. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene Selection for Cancer Classification using Support Vector Machines. *Mach. Learn.* **2002**, *46*, 389–422. [CrossRef]
40. Haddadpajouh, H.; Mohtadi, A.; Dehghantanaha, A.; Karimipour, H.; Lin, X.; Choo, K.K.R. A Multikernel and Metaheuristic Feature Selection Approach for IoT Malware Threat Hunting in the Edge Layer. *IEEE Internet Things J.* **2021**, *8*, 4540–4547. [CrossRef]
41. Hoque, N.; Bhattacharyya, D.K.; Kalita, J.K. MIFS-ND: A mutual information-based feature selection method. *Expert Syst. Appl.* **2014**, *41*, 6371–6385. [CrossRef]
42. Yılmaz, E. An expert system based on Fisher score and LS-SVM for cardiac arrhythmia diagnosis. *Comput. Math. Methods Med.* **2013**, *2013*, 849674. [CrossRef]
43. Bishop, C. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006; pp. 298–299.
44. Chen, X.W.; Jeong, J.C. Enhanced recursive feature elimination. In Proceedings of the 6th International Conference on Machine Learning and Applications (ICMLA 2007), Cincinnati, OH, USA, 13–15 December 2007; pp. 429–435. [CrossRef]
45. Chen, Y.; Ma, L.; Yu, D.; Zhang, H.; Feng, K.; Wang, X.; Song, J. Comparison of feature selection methods for mapping soil organic matter in subtropical restored forests. *Ecol. Indic.* **2022**, *135*, 108545. [CrossRef]
46. Hespeler, S.; Fuqua, D. Online State of Charge Prediction in Next Generation Vehicle Batteries using Deep Recurrent Neural Networks and Continuous Model Size Control. *J. Energy Power Technol.* **2021**, *3*, 1–24. [CrossRef]
47. Hastie, T.; Tibshirani, R.; Friedman, J. Random forests. In *The Elements of Statistical Learning*; Springer: New York, NY, USA, 2009; pp. 587–604.
48. Kang, M.; Kim, J.; Kim, J.M.; Tan, A.C.C.; Kim, E.Y.; Choi, B.K. Reliable fault diagnosis for low-speed bearings using individually trained support vector machines with kernel discriminative feature analysis. *IEEE Trans. Power Electron.* **2015**, *30*, 2786–2797. [CrossRef]
49. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
50. Chen, J.C.; Chen, T.L.; Liu, W.J.; Cheng, C.C.; Li, M.G. Combining empirical mode decomposition and deep recurrent neural networks for predictive maintenance of lithium-ion battery. *Adv. Eng. Inform.* **2021**, *50*, 101405. [CrossRef]
51. Zhang, X.; Zhao, B.; Lin, Y. Machine learning-based bearing fault diagnosis using Case Western Reserve University Data. *IEEE Access* **2021**, *29*, 155598–155608. [CrossRef]
52. Puente, D.; Amelibia, J.; Cumplido, I.; Hernandez, A.; Ugarte, I.; Duo, A. Data-driven methodology for optimal Lithium-ion battery RUL prediction. *Preprint* **2023**. [CrossRef]
53. Yang, C.; Ge, H.; Jin, H.; Liu, S. Airborne Lithium Battery Health Assessment: An Improved Support Vector Machine Algorithm for Imbalanced Sample Sets. *Aerospace* **2024**, *11*, 467. [CrossRef]
54. Harippriya, S.; Vigneswaran, E.E.; Jayanthy, S. Battery management system to estimate battery aging using deep learning and machine learning algorithms. *J. Phys. Conf. Ser.* **2022**, *2325*, 012004. [CrossRef]