

Article

A Density-Based Feature Space Optimization Approach for Intelligent Fault Diagnosis in Smart Manufacturing Systems

Junyoung Yun ¹, Kyung-Chul Cho ², Wonmo Kang ³, Changwan Kim ⁴, Heung Soo Kim ⁵
and Changwoo Lee ^{4,*}

¹ Department of Mechanical Design and Production Engineering, Konkuk University, 120 Neungdong-ro, Gwangjin-gu, Seoul 05029, Republic of Korea; jun980520@konkuk.ac.kr

² Energy DX Research Department, Korea Textile Machinery Convergence Research Institute, Gyeongsan-si 38542, Republic of Korea; chokc@kotmi.re.kr

³ School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85281, USA; wonmo.kang@asu.edu

⁴ School of Mechanical Engineering, Konkuk University, 120 Neungdong-ro, Gwangjin-gu, Seoul 05029, Republic of Korea; goodant@konkuk.ac.kr

⁵ Department of Mechanical, Robotics and Energy Engineering, Dongguk University-Seoul, 30 Pildong-ro 1-gil, Jung-gu, Seoul 04620, Republic of Korea; heungsoo@dgu.edu

* Correspondence: leewoo1220@konkuk.ac.kr; Tel.: +82-2-450-3570

Abstract

In light of ongoing advancements in smart manufacturing, there is a growing need for intelligent fault diagnosis methods that maintain reliability under noisy, high-variability operating conditions. Conventional feature selection strategies often struggle when data contain outliers or suboptimal feature subsets, limiting their diagnostic utility. This study introduces a density-based feature space optimization (DBFSO) framework that integrates feature selection with localized density estimation to enhance feature space separability and classifier efficiency. Using k-nearest neighbor density estimation, the method identifies and removes low-density feature vectors associated with noise or outlier behavior, thereby sharpening the feature space and improving class discriminability. Experiments using roll-to-roll (R2R) manufacturing data under mechanical disturbances demonstrate that DBFSO improves classification accuracy by up to 36–40% when suboptimal feature subsets are used and reduces training time by 60–71% due to reduced feature space volume. Even with already-optimized feature sets, DBFSO provides consistent performance gains and increased robustness against operational variability. Additional validation using a bearing fault dataset confirms that the framework generalizes across domains, yielding improved accuracy and significantly more compact, noise-resistant feature representations. These findings highlight DBFSO as an effective preprocessing strategy for intelligent fault diagnosis in intelligent manufacturing systems.

Keywords: data-processing; fault diagnosis; feature engineering; feature space optimization; roll-to-roll system; smart manufacturing

MSC: 68T01



Academic Editors: Yu Xue, Yong Zhang, Bing Xue and Ferrante Neri

Received: 19 November 2025

Revised: 8 December 2025

Accepted: 12 December 2025

Published: 14 December 2025

Citation: Yun, J.; Cho, K.-C.; Kang, W.; Kim, C.; Kim, H.S.; Lee, C. A Density-Based Feature Space Optimization Approach for Intelligent Fault Diagnosis in Smart Manufacturing Systems. *Mathematics* **2025**, *13*, 3984. <https://doi.org/10.3390/math13243984>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rapid advancement of smart manufacturing systems has increased the demand for intelligent diagnostic tools capable of early fault detection [1,2] and real-time decision-making in complex industrial environments [3]. Among critical components, rotary

systems—such as those found in wind turbines [4], electric vehicles [5–7], and roll-to-roll (R2R) production lines—are especially vulnerable to mechanical faults that compromise operational stability [8,9]. Vibration signals acquired from such systems offer essential insights for identifying issues such as bearing wear, imbalance, and roll eccentricity [10,11], which, if left undetected, can lead to product defects or costly downtime [12,13]. For example, tension disturbances in R2R systems caused by roll eccentricity or shaft misalignment may propagate through production lines, resulting in unstable processing and yield losses [14,15].

To address these challenges, various machine learning models have been developed for fault diagnosis. However, their performance is often highly dependent on the quality and structure of the underlying feature space. Recent efforts have explored fault prognostics [16], remaining useful life (RUL) prediction [17–19], and eccentricity compensation using vibration or tension data; however, many of these approaches suffer from performance degradation under noisy conditions, sparse data distributions, or underrepresented fault types [20]. This demonstrates the need for more robust and adaptive feature processing techniques that can address real-world industrial variability.

Recent studies have introduced deep learning frameworks that incorporate multi-sensor fusion and semi-supervised learning to enhance fault diagnosis under varying operational conditions [21,22]. These models often employ contrastive learning, domain adaptation, and temperature-guided optimization to extract domain-invariant features from partially labeled or noisy datasets. Some methods also separate domain-specific and shared representations to improve generalizability. Although such methods have achieved high diagnostic accuracy, they often require complex architectures and significant computational resources, limiting their suitability for real-time or lightweight industrial applications.

In parallel, data-driven diagnostic methods face ongoing challenges when applied to large, noisy datasets—particularly those with uneven or sparse sample distributions. To mitigate these issues, several studies have explored feature selection techniques that evaluate the discriminative power of feature combinations [23–25]. Although such approaches can improve classification accuracy, their performance often decreases in the presence of real-world noise or imbalanced data, which can obscure critical fault patterns [26,27].

These limitations highlight the need for diagnostic methods that are lightweight, interpretable, and robust to noise—especially those capable of enhancing the feature space without relying on deep models. While density-based methods such as Density Peak Clustering (DPC) and Local Outlier Factor (LOF) have been utilized for data mining and anomaly detection, these are primarily designed for unsupervised tasks. Their focus lies on identifying cluster centers or detecting deviations based on local density, rather than actively optimizing the feature space to enhance class separability within a supervised fault diagnosis framework. This clear distinction highlights the necessity of our proposed supervised density-based filtering approach for targeted feature refinement [28,29].

Despite extensive research across feature engineering, density-based analysis, and deep learning-based diagnosis, a critical gap persists. Conventional feature selection methods primarily assess feature importance but do not correct structural distortions in the feature space caused by noise, outliers, or sparse sample distributions—issues that are common in real manufacturing environments. Likewise, classical density-based methods do not incorporate class labels and therefore cannot selectively preserve discriminative samples that contribute to class separation. Although deep learning models can extract robust representations, their dependence on large datasets and high computational resources hinders deployment in lightweight industrial systems.

Therefore, there remains a clear research gap for a supervised, density-aware, and computationally efficient approach that directly refines the feature space by removing low-density, unrepresentative samples while preserving high-density, discriminative structures. Such a method is particularly valuable in R2R processes, where mechanical variability and tension disturbances often lead to degraded feature quality and reduced diagnostic performance. This gap provides the foundation and motivation for the method proposed in this study.

In this context, we propose a novel density-based feature space optimization (DBFSO) method. Unlike conventional approaches that preprocess raw sensor signals, DBFSO operates directly on the constructed feature space. This method aims to enhance feature separability by filtering out sparse or noisy data points based on their local density. Specifically, features are extracted under both normal and abnormal conditions to form a structured three-dimensional space [30–32]. A k -nearest neighbor (k -NN) algorithm is then used to compute the local density of each point, assigning density scores across the volume [33–37].

Based on the research gap identified above, the present study addresses the following research questions: (i) whether a supervised density-based filtering approach can enhance the separability of engineered feature spaces by selectively removing low-density and unrepresentative samples, (ii) how such refinement influences classification accuracy and computational efficiency under different feature subsets and operating conditions, including noisy or sparse data distributions, and (iii) whether the proposed method generalizes across different types of mechanical faults, particularly in R2R tension disturbances and bearing vibration anomalies, without relying on computationally heavy deep learning models. These research questions guide the development, evaluation, and validation of the proposed DBFSO framework.

The proposed approach is supported by prior studies on density-based clustering and classification, which show that points in low-density regions—often corresponding to noise or outliers—typically have minimal influence on classification outcomes [38,39]. Data points in the lowest percentile of these scores are systematically removed using percentile-based thresholds. This filtering process refines the feature space, retaining only high-density discriminative samples, thereby improving classifier performance [40,41].

To validate the generalizability and effectiveness of DBFSO, we apply it to two distinct industrial datasets: (1) an R2R system exhibiting tension fluctuations and roll eccentricity and (2) a bearing fault dataset including cage defects and imbalance. By evaluating the performance across different systems, we demonstrate that DBFSO effectively improves classification accuracy and robustness in diverse mechanical scenarios.

Experimental results show that DBFSO enhances classification performance even when combined with suboptimal or noisy feature sets. Although gains are also observed under ideal feature combinations, the greatest benefits of this method are noted when the data quality is compromised, which highlights its potential for real-world fault diagnosis in smart manufacturing environments. Compared with conventional feature selection methods, DBFSO offers a lightweight and interpretable solution that significantly improves both the reliability and efficiency of data-driven diagnostics.

2. Theoretical Background

2.1. Problem Statement

The primary challenge in data-driven fault diagnosis for mechanical systems in smart manufacturing lies in accurately classifying fault conditions within extensive datasets that are often noisy and sparsely distributed. Conventional feature selection methods, such as minimum redundancy maximum relevance and analysis of variance, primarily evaluate

individual feature relevance without considering the combined effects of multiple features, thereby limiting their effectiveness in capturing complex fault patterns. To address this limitation, approaches that construct two- or three-dimensional feature spaces and assess the separability between normal and faulty conditions have been proposed. However, these methods are susceptible to noise and outliers, which can distort feature distributions and compromise classification performance.

With the increasing complexity of industrial systems, the need for real-time monitoring and rapid fault detection to ensure operational stability has increased. With machines continuously generating considerable amounts of data, the efficient extraction of pertinent information while minimizing processing time is necessary. Conventional diagnostic models often struggle with high dimensionality and noise inherent in such datasets, resulting in decreased accuracy. In this study, a fault diagnosis model was developed utilizing the proposed density-based filtering method to address these challenges. By integrating this approach with a support vector machine, the performance of the model was systematically evaluated, revealing significant enhancements in classification accuracy and diagnostic reliability, particularly in environments with high noise levels or sparsely distributed data.

2.2. Feature Engineering and Optimal Feature Combination

As a crucial process in machine learning, feature engineering involves the transformation of raw data into meaningful variables using domain knowledge. It is crucial in model performance, as the type and quality of input data significantly influence the outcome [42]. Feature engineering comprises two main components: feature extraction and selection [43,44]. The former creates new features from the original data, whereas the latter filters out irrelevant or redundant features, thereby producing concise and informative subsets. Both techniques reduce the dimensionality of extensive datasets, thereby preventing overfitting and improving model efficiency. In this study, feature variables, such as skewness, kurtosis, and standard deviation, were extracted from the raw data and coordinated into sets of three to form novel features. These new features were then utilized in fault detection and classification models.

Lee et al. introduced feature variable dimensional coordination (FDC), a framework designed to optimize the feature space for fault detection and classification [45]. FDC aims to increase diagnostic accuracy by carefully selecting feature combinations that maximize the separability between normal and faulty conditions. This method groups feature variables into sets of three, creating a three-dimensional feature space. The effectiveness of these feature sets was evaluated based on the volume of features and Mahalanobis distance, a metric that measures the distance between normal and faulty data distributions. The combination of features that yielded the highest score was considered the most effective for classification. The FDC number is calculated as follows:

$$FDC_N = Mn_d \left(\frac{V_2 - V_1}{V_2 + V_1} \right), \quad (1)$$

where V_1 represents the volume of a healthy condition, V_2 represents the volume of a defect condition, and Mn_d represents the Mahalanobis distance between the two conditions.

Although feature selection enhances classification accuracy by choosing the best combinations of features, its effectiveness can be compromised in the presence of noise or sparsely distributed datasets. The FDC method, which relies on feature volume, is particularly vulnerable to the presence of outliers or noisy data points. Given that the FDC method relies on feature volume, even a single noisy or outlier data point can distort feature distributions, causing overlaps between normal and faulty conditions and significantly impacting the computed FDC scores. To address this challenge, we introduce a density-

based filtering mechanism that eliminates low-density points from the feature space. This refinement ensures that only the most representative data clusters are retained, thereby improving the robustness and effectiveness of fault detection models.

2.3. Comparison with Existing Density-Based Methods

Several density-based methods, such as DPC and LOF, have been applied to tasks such as feature selection and noise reduction. However, these methods differ fundamentally from DBFSO in both objective and function within classification pipelines. DPC focuses on identifying high-density regions to assign cluster centers, rendering it well suited for unsupervised clustering but less applicable to supervised feature space refinement [28,29]. By contrast, LOF detects anomalies based on deviations in local density but does not explicitly optimize the feature space to improve classification accuracy.

DBFSO is purpose-built for supervised classification scenarios where normal and abnormal class labels are available. It aims to refine the feature space with class separability in mind, enabling more targeted and effective learning. By estimating local feature densities via the k-NN approach, DBFSO selectively filters out sparsely distributed data points that are likely to introduce noise or obscure decision boundaries. This enhances the discriminative power of the classifier. Unlike DPC and LOF, which operate in an unsupervised manner, DBFSO is label guided, ensuring that feature space optimization directly supports the classification objective.

The motivation for adopting a density-based filtering approach lies in the consistent structure of fault-related data: abnormal conditions often form compact, high-density clusters in feature space because of repeatable fault signatures. By contrast, the noise of ambiguous samples typically appears in low-density regions. By quantifying local densities, DBFSO can systematically identify and remove such points before training. This leads to a more representative and cleaner feature distribution, ultimately improving both the robustness and accuracy of the classification model, especially in fault diagnosis tasks where noticeable distinctions between normal and abnormal states are critical.

2.4. Support Vector Machine (SVM) Algorithm for Fault Classification

SVM is a supervised learning algorithm commonly utilized in pattern recognition and classification tasks, including machine fault diagnosis [45,46]. Based on statistical learning theory it aims to optimize classification by maximizing the margin between distinct classes. By identifying an optimal hyperplane that effectively separates data points, SVM enhances classification accuracy, particularly in high-dimensional feature spaces. The effectiveness of the SVM is attributed to its ability to create a decision boundary with the largest possible margin, ensuring robust generalization. However, managing outliers is crucial, as they directly impact the decision boundary. Hard margins, which offer high precision, can lead to overfitting, whereas soft margins, which allow misclassification, can prevent underfitting but may reduce classification accuracy. Balancing these factors is essential for achieving optimal performance in fault diagnosis and similar applications [47].

2.5. k-NN Algorithm and Density Estimation

The k-NN algorithm is a widely utilized nonparametric method for density estimation. It offers adaptability to complex data structures without requiring prior assumptions regarding data distribution. Unlike kernel density estimation, which uses a fixed bandwidth, k-NN dynamically adjusts the neighborhood size based on local data density. This flexibility enables k-NN to capture density variations effectively in irregular or multidimensional datasets, rendering it suitable for diverse applications.

In k-NN density estimation, the local density of a point is determined by its distance to the k -th nearest neighbor, with shorter distances indicating higher densities. The density

at a point x , denoted as $\rho(x)$, can be expressed in terms of the distance to its k -th nearest neighbor, $r_k(x)$, and the total number of points in the dataset, n . Assuming a Euclidean space with dimensionality d , the simplified formula is as follows:

$$\rho(x) = \frac{k}{n \cdot V_d \cdot r_k(x)^d}, \quad (2)$$

where k represents the number of nearest neighbors employed in the calculation, n represents the total number of points in the dataset, $r_k(x)$ represents the Euclidean distance from x to its KNN , d represents the dimensionality of the data space, and V_d represents the volume of a unit hypersphere in a d -dimensional space, computed as

$$V_d = \frac{\pi^{d/2}}{\Gamma(d/2 + 1)}. \quad (3)$$

The parameter k influences the balance between sensitivity to local noise (variance) and the ability to generalize the true underlying distribution (bias) [48,49]. Optimal k -selection is often guided by minimizing the asymptotic mean integrated square error to achieve robust density estimates.

Beyond its statistical properties, k -NN density estimation is particularly well suited to the objectives of this study. The DBFSO framework relies on identifying sparse, low-density regions within the engineered feature space in order to remove unrepresentative or noisy samples. Because k -NN provides a strictly local and nonparametric density measure, it adapts naturally to datasets with uneven distributions, localized clusters, or sparse faulty samples—conditions commonly observed in roll-to-roll tension signals and bearing vibration data. Moreover, the relatively low dimensionality of the constructed 3D feature space allows k -NN to operate with high computational efficiency. Finally, the k -NN formulation integrates directly with the percentile-based filtering strategy of DBFSO, enabling straightforward identification of low-density points without the need for global distribution assumptions or bandwidth tuning. These considerations collectively motivated the selection of k -NN as the density estimator in this work.

3. Proposed Methodology

3.1. Quantification of Feature Density in Three-Dimensional Feature Space

The DBFSO method leverages feature engineering within a three-dimensional feature space to analyze data obtained from the system. The feature extraction involves the use of statistical variables, such as the mean, kurtosis, and skewness. These extracted features were used to construct a feature space with three distinct variables. The density of the scattered feature points in this space is quantified using the k -NN algorithm.

The implementation of the k -NN algorithm for density estimation is shown in Figure 1. The density calculation requires the following: (a) a three-dimensional scatter plot, where the feature points are color-coded based on their calculated density, and (b) a two-dimensional projection at $y = 0.2$ to showcase the density distribution on a specific plane. The parameter k is defined as the square root of the dataset size ($k = \sqrt{N}$) to ensure a representative analysis [50]. To provide a rigorous justification for this choice in the fault diagnosis feature spaces, a comprehensive sensitivity analysis was conducted to evaluate the impact of the k parameter on the final classification accuracy. The analysis was performed on representative datasets with varying noise levels, testing a wide range of k values, including 5, 10, 100, $\sqrt{N}/2$, $\sqrt{N}$, and $2\sqrt{N}$. The results conclusively demonstrated that the model's performance remained highly stable across all these different k values, exhibiting a standard deviation of less than 0.5% in classification accuracy. The framework's density-

based approach is inherently robust to minor changes in the neighborhood parameter k because it relies on the overall density distribution of the feature space rather than on a fixed number of nearest neighbors, which can fluctuate more dramatically. This demonstrates the exceptional stability of the DBFSO framework and confirms that the $k = \sqrt{N}$ heuristic is a rigorously justifiable and effective choice for this application. As shown in Figure S1 (Supplementary Material), the classification accuracy shows minimal variation with changes in k , with the selected heuristic providing a reliable trade-off between performance and computational efficiency. The calculated density values were normalized to a range from 0 to 1 for consistent interpretation and visualization and then plotted accordingly.

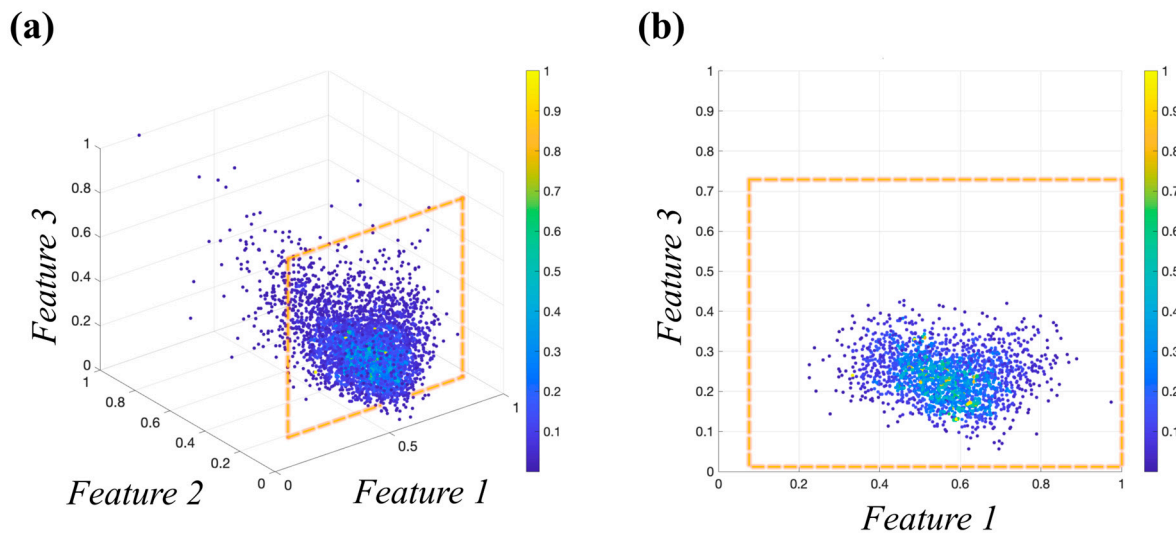


Figure 1. Density calculation based on the k-NN algorithm: (a) three-dimensional scatter and (b) two-dimensional projection (dashed box showing $y = 0.2$).

3.2. Thresholding for Low-Density Point Removal

By utilizing the calculated density values, each scattered data point is assigned a corresponding density score, aiding in the determination of a filtering threshold. To accommodate the inherent variability across datasets and environmental conditions, the filtering level is designed to be adaptable. In this study, the threshold is parameterized as a percentile γ of the density values, providing a flexible framework in which users can dynamically adjust the filtering level. The threshold is at a specific percentile of the density distribution, and only samples with density values greater than or equal to the threshold are retained. The filtered dataset $\chi_{filtered}$ is formally defined as follows:

$$\chi_{filtered} = x_i \in \chi \mid \rho(x_i) \geq \text{Percentile}(\rho, \gamma) \quad (4)$$

This filtering step helps suppress the influence of noise and outliers by focusing the classification on high-density, structure-representative regions.

The threshold can be adjusted dynamically based on the statistical properties of the density distribution, such as the mean or median density, to align with specific data characteristics. Alternatively, the optimal threshold can be determined empirically via a validation dataset, selecting the value that maximizes performance metrics, such as classification accuracy, recall, and precision [51,52]. This adaptability ensures that the method remains robust across various scenarios, regardless of whether noise reduction is prioritized or sparse anomalies are critical.

It is important to note that the density threshold used in this study is not based on an absolute density value or a fixed ranking, but is explicitly defined as a percentile-based

cutoff computed from the relative ranking of local densities of the training dataset. This percentile-based approach is chosen because absolute density values vary across datasets and feature spaces, whereas relative percentiles provide a scale-invariant and distribution-adaptive criterion for robust filtering.

The impact of varying the density threshold, ranging from 100% to 90%, through a series of scatter plots is shown in Figure 2. The figure is presented from two complementary perspectives: (a) density-based visualization, in which data points are color-coded according to their density values, and (b) condition-based visualization, in which data points are classified and color-coded as red for fault data and blue for normal data. The axis ranges were determined based on the data range to ensure optimal visualization and accurate dataset representation. This dual representation underscores the impact of density-based filtering on the overall data distribution and its implications for classifying normal and fault conditions.

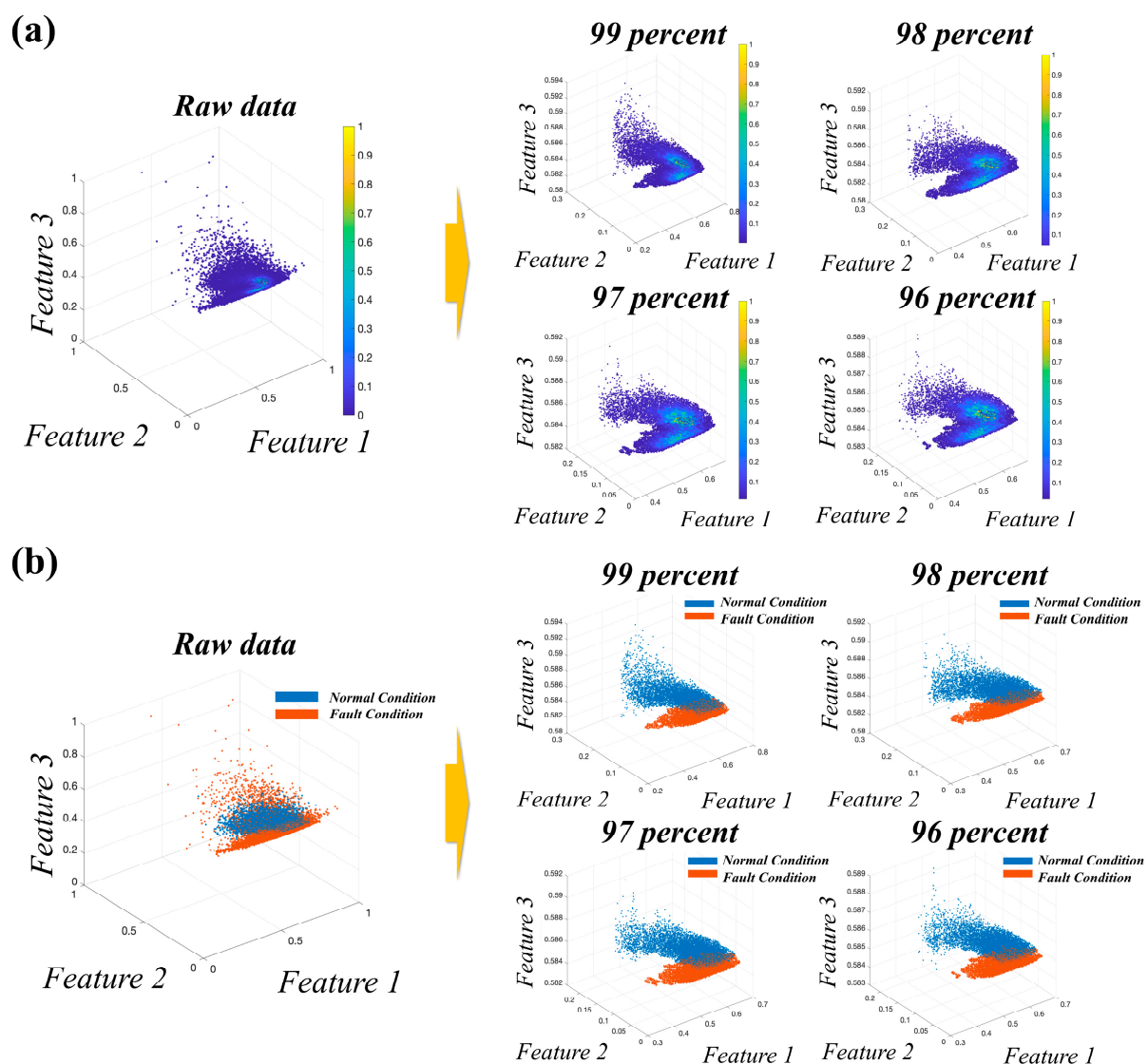


Figure 2. Impact of density threshold: (a) density-based and (b) condition-based visualization.

As shown in Figure 2, the filtering process retained the core, dense, or critical data within the dataset while eliminating less relevant, low-density data points. This approach ensures that essential patterns and significant information are preserved, thereby enhancing the reliability of the subsequent analysis and classification of normal and fault conditions.

3.3. Integration with Feature Selection Method for Enhanced Accuracy

The DBFSO methodology operates within a three-dimensional feature space, leveraging feature combinations selected by the FDC method. The FDC method evaluates feature combinations based on Mahalanobis distance, identifying those that maximize class separability. Before applying DBFSO, a baseline pipeline was established to provide a comparison. In this “without DBFSO” condition, the feature extraction followed the same procedure: statistical variables such as mean, skewness, and kurtosis were extracted and coordinated into three-dimensional feature combinations using the FDC method. No density-based filtering was applied, and all extracted feature points—including low-density or noisy points—were retained. The resulting feature sets were normalized to the [0, 1] range and directly used as inputs for a SVM classifier. SVM hyperparameters were selected via cross-validation on the training set, and the training, validation, and testing splits were identical to those used in the DBFSO-enhanced pipeline. This setup ensures a fair and reproducible comparison and isolates the effect of DBFSO on classification performance.

DBFSO enhances these feature combinations by applying density-based filtering to improve the robustness of the feature space. By using k-NN for density estimation, DBFSO eliminates low-density points that often represent noise or outliers, retaining only the most representative clusters. This sequential approach—first scoring feature combinations through FDC and then optimizing them through DBFSO—ensures a comprehensive enhancement of the classification model. It should be noted that FDC is used solely to generate feature combinations of varying expected quality, and the evaluation of DBFSO does not rely on the strict optimality of these combinations. DBFSO demonstrated consistent performance improvements across top-ranked, low-ranked, and noise-perturbed feature sets, confirming its robustness to variations in initial feature scoring. The DBFSO method was validated under four scenarios:

1. Optimal combination: The best feature set selected based on the Mahalanobis distance was evaluated, ensuring maximum class separability and effectiveness.
2. Suboptimal combination: The least effective feature sets were tested to assess the robustness of the method in less-than-ideal scenarios.
3. Noise resilience analysis: Simulated noise was introduced into the data to validate the performance of the method under noisy and sparsely distributed conditions.
4. Application of DBFSO: The generalizability of the proposed method was verified utilizing bearing fault data.

This comprehensive validation ensures that the DBFSO method is not only effective in ideal scenarios but also adaptable and reliable under suboptimal and random feature selection conditions. By leveraging the DBFSO framework, the selection and optimization of feature combinations were further refined, leading to significantly improved overall performance and robustness. Additional details are provided in Section 4.1.

3.4. Implementation of the DBFSO Method

The DBFSO method was executed in seven systematic steps, as shown in Figure 3, and was categorized into three distinct sections. The first section involves data acquisition, the second focuses on feature engineering, and the third addresses the development and validation of the fault diagnosis model via the DBFSO framework.

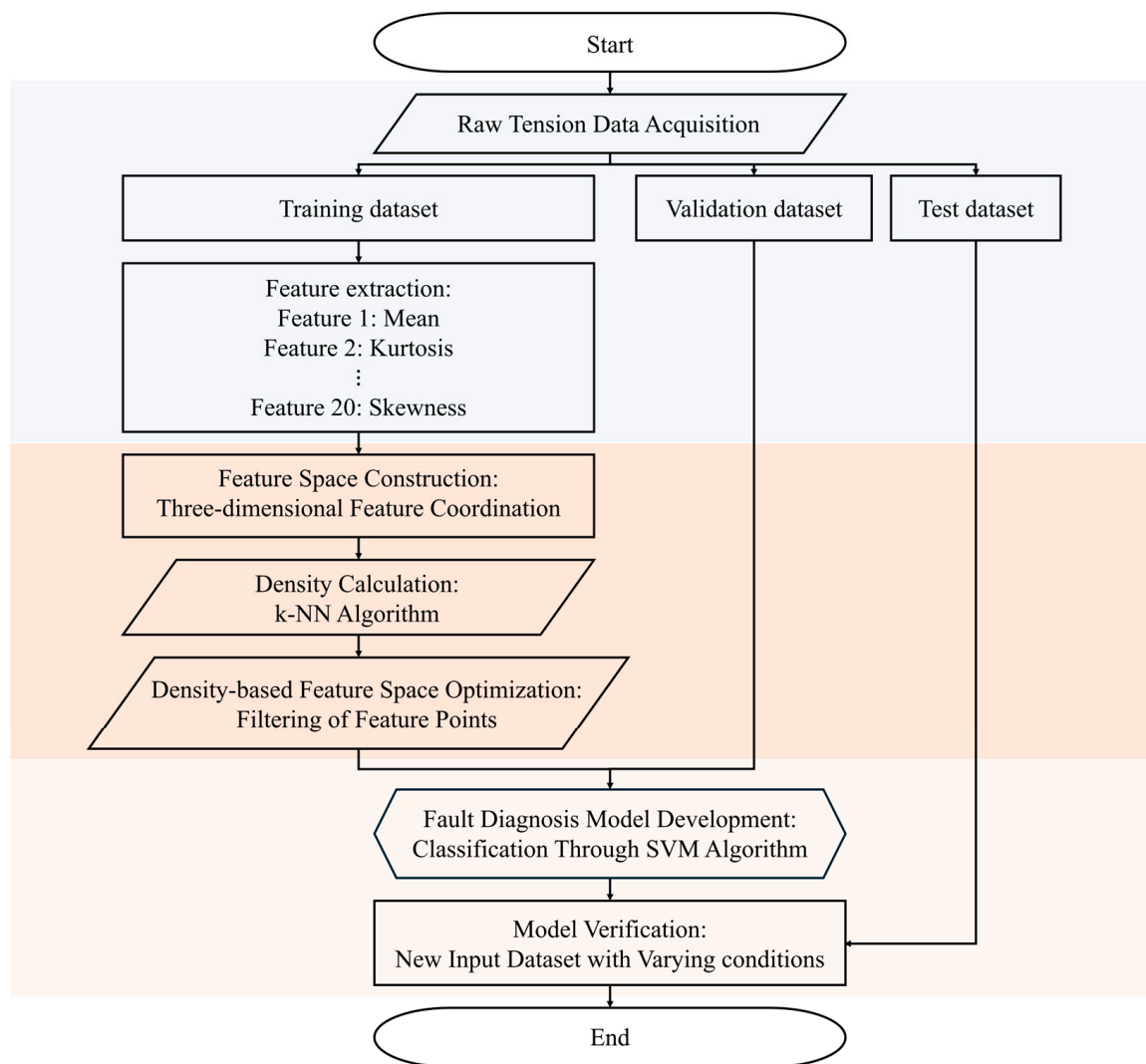


Figure 3. Flowchart of the DBFSO method.

Step 1: Raw data were collected from system sensors and categorized as normal or abnormal. These time series data serve as the foundation for preprocessing and feature extraction, with the data being initially split into training, validation, and test datasets before any further processing.

Step 2: Key statistical and signal-based features, such as the mean, kurtosis, and root mean square, were extracted from the vibration data within the predefined time windows. This step results in a multidimensional feature matrix that captures the patterns of normal and abnormal conditions. Twenty statistical features were utilized in this step (Table S1, Supplementary Material).

Step 3: The extracted features were mapped onto a three-dimensional feature space to differentiate between normal and abnormal data. This framework highlights data distributions and relationships for further analysis.

Step 4: The KNN algorithm calculated the local density for each feature point, identifying dense clusters and low-density regions. These density values guided subsequent data refinement.

Step 5: Low-density points were filtered using a threshold based on density percentiles, enhancing the distinction between different conditions and reducing noise in the feature space.

Step 6: The SVM model was trained using the optimized feature space to classify the data as normal or faulty, with a focus on achieving high accuracy in fault detection.

Step 7: The model was tested using validation datasets under various conditions to assess its robustness. This ensures reliable diagnostic performance across diverse scenarios.

3.5. Data Acquisition and Preprocessing

This section presents the experimental framework used to validate the proposed DBFSO method. An R2R web transport system was utilized as a testbed to assess the effectiveness of DBFSO in fault diagnosis under various operating conditions. The subsections provide an overview of the system architecture, fault conditions, and data acquisition methodology.

3.5.1. R2R System and Fault Conditions

The R2R system utilized in this study is a web transport platform designed for the continuous processing of flexible substrates under controlled tension. The system includes an infeeder, an outfeeder, and a rewinder, with multiple rollers guiding the web. These systems are commonly utilized in advanced manufacturing applications, including flexible electronics, printed sensors, and functional coatings, where precise tension regulation is crucial for ensuring product quality and process stability.

Roll eccentricity is a primary mechanical anomaly that impacts R2R systems. This occurs when the rotational axis of a roller deviates from its geometric center. This misalignment induces periodic tension fluctuations, resulting in web misalignment, reduced coating uniformity, and defects in the printed electronics. Two distinct fault conditions were introduced to explore the effect of eccentricity: (1) eccentricity in Span 2 (infeeder–outfeeder) and (2) eccentricity in Span 3 (outfeeder–rewinder). These conditions were selected to assess the DBFSO method under varying mechanical disturbances, as the eccentricity in different spans uniquely modulates tension dynamics. By analyzing the resulting tension profile fluctuations, the proposed DBFSO method can be thoroughly evaluated in terms of its ability to filter noise and enhance classification accuracy. A schematic of the R2R system and its key components is shown in Figure 4.

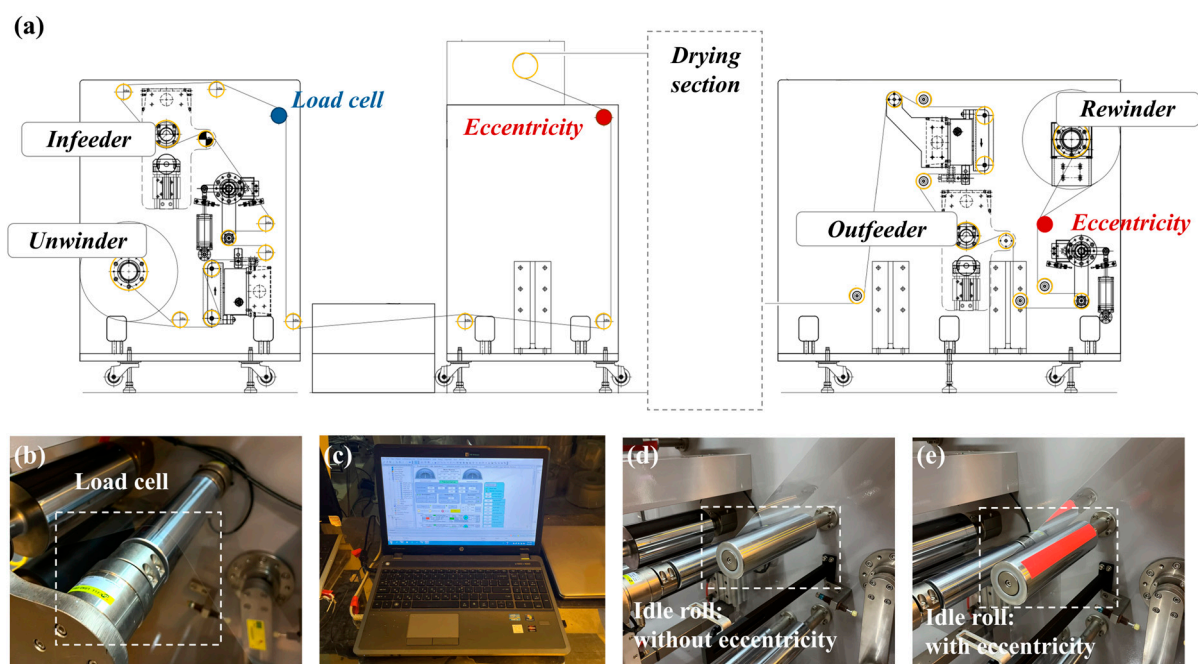


Figure 4. Experimental setup for data acquisition: (a) schematics of the roll-to-roll system, (b) load cell, (c) data acquisition module, (d) idle roll without eccentricity, and (e) with eccentricity.

3.5.2. Experimental Setup and Data Collection

A high-precision load cell (DCELL Inc., RTB15, rated capacity: 50 kgf, sensitivity: 1 mV/V) was integrated into the R2R system to capture real-time tension variations under both normal and faulty conditions (Specifications listed in Table S2, Supplementary Material). This integration allowed for continuous monitoring of fluctuations in the web, serving as the primary data source for fault detection. Data acquisition was performed using a National Instruments cDAQ-9174/IN9329 module at a sampling rate of 2000 Hz (as listed in Table 1), ensuring high-fidelity signal capture with minimal latency. The key experimental parameters are listed in Table 1. Nine datasets were collected across three operational conditions—normal operation, eccentricity in Span 2, and eccentricity in Span 3—with each condition repeated thrice to ensure statistical reliability (Figure S2, Supplementary Material).

Table 1. Data acquisition setup and experimental design.

	Parameter	Value
Operating Conditions	Speed [mpm]	5
	Tension [N/m]	177
Sensor	Load Cell	DCELL.INC RTB15
	Data Acquisition Module	NI cDAQ-9174/IN9329
	Sampling Rate [Hz]	2000
	Sampling Duration [s]	300
Eccentricity	Thickness [μm]	40
	Width [mm]	25
System Condition	Normal	3 trials
	Eccentricity in Span 2 (Infeeder-Outfeeder)	3 trials
	Eccentricity in Span 3 (Outfeeder-Rewinder)	3 trials

3.5.3. Cross-Validation, Data Splitting, and Model Setup

To ensure a rigorous and reproducible methodology, a strict cross-validation strategy was implemented. The collected data were first labeled as normal or abnormal and then randomly partitioned into three distinct subsets: a training set (70%), a validation set (15%), and a test set (15%). This split ensures a robust evaluation by keeping a portion of the data completely unseen until the final step.

Feature extraction was performed on time series segments using a fixed window size of 100 samples. This value was empirically selected because it provided a stable representation of the signal patterns across both classes. The SVM classification algorithm was implemented in MATLAB 2024a.

For each cross-validation fold, the DBFSO framework was applied strictly and exclusively to the training dataset. All density calculations, neighborhood structures, and threshold determinations were computed using only the samples contained in the corresponding training split. The validation and test datasets were never used during any stage of density estimation or filtering, thereby ensuring that no information from the evaluation data influenced the DBFSO process or the classifier training. After DBFSO filtering, the reduced training set was used to train the SVM classifier. The SVM hyperparameters were then optimized through a grid search with fivefold cross-validation performed only on the training and validation subsets of each fold. The hyperparameter configuration that achieved the highest validation accuracy was selected and subsequently fixed when evaluating the classifier on the untouched test set. This protocol guarantees a fair comparison between methods and eliminates any possibility of data leakage across folds.

4. Validation Scenarios and Comparative Analysis

4.1. Classification Accuracy and Model Performance

This section evaluates the classification accuracy and model performance across the constructed cases, defined as follows: (Case 1) Normal and eccentricity in Span 2 with an optimal combination; (Case 2) normal and eccentricity in Span 2 with a suboptimal combination; (Case 3) normal and eccentricity in Span 3 with an optimal combination; and (Case 4) normal and eccentricity in Span 3 with a suboptimal combination. The optimal and suboptimal combinations were selected based on the FDC method. For clarity, the cases with and without additional noise were labeled as Case 1-2 through Case 4-2 and Case 1-1 through Case 4-1, respectively. The analysis commenced with a focus on Case 1-1.

The results, categorized by the DBFSO levels, highlighting the effects of data filtering on accuracy, training time, and feature separation, are presented in Figure 5 and Table 2. The range considered, from 100% to 95%, was selected based on classification outcomes observed across a broader range (100–90%), revealing significant trends (Figures S3 and S4, Supplementary Material). The accuracy values were averaged over five trials. We have included error bars to demonstrate the variability in the results; however, the standard deviation was consistently low, which is a strong indicator of the framework's stability and the reproducibility of its performance.

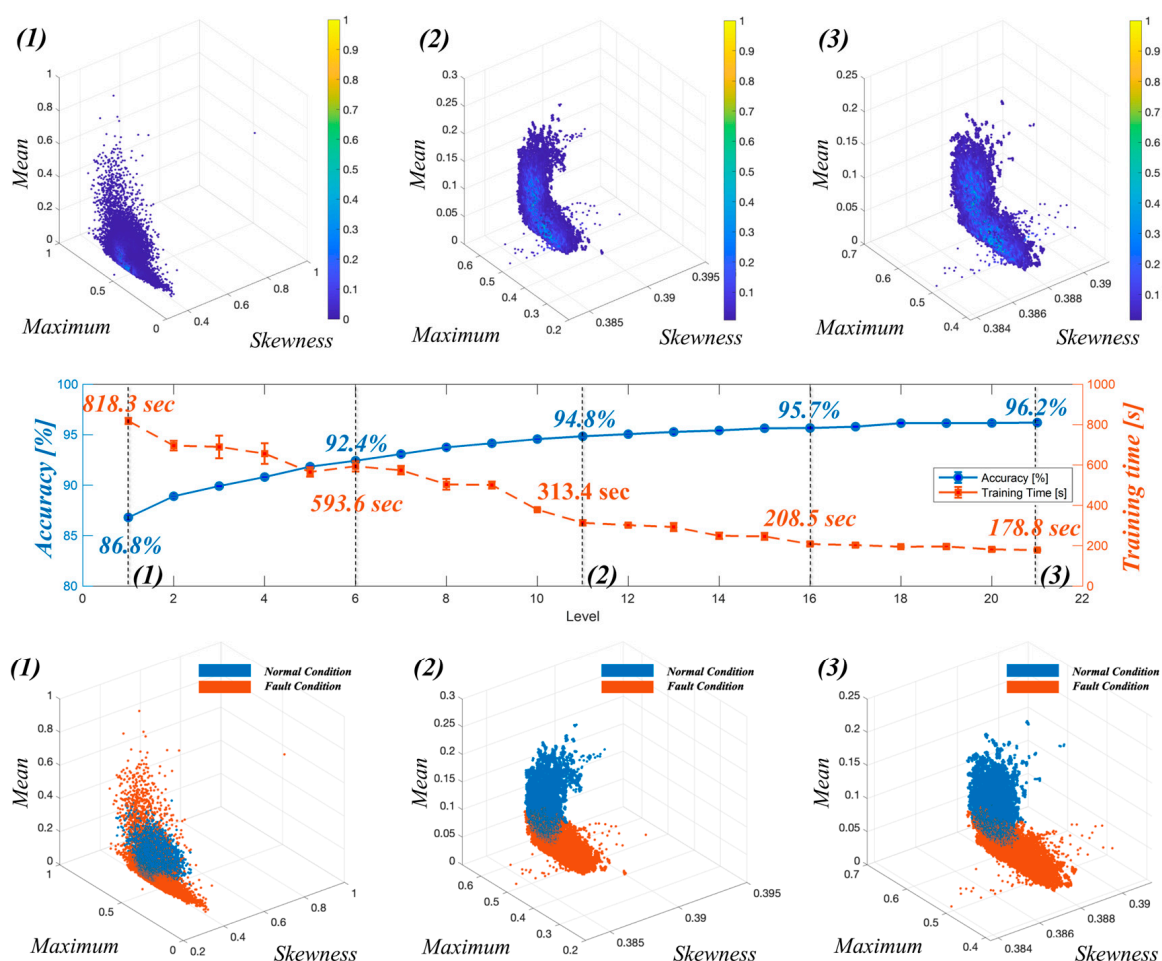


Figure 5. Case 1-1 results categorized by density-based feature space optimization (DBFSO) levels: (top) density-based visualization, (middle) accuracy and training time plots, and (bottom) condition-based visualization.

Table 2. Classification results of Case 1-1 by filtering level: optimal combination.

Optimal Combination	Parameter	DBFSO by Percentage				
		100.0%	98.75%	97.5%	96.25%	95.0%
Mean Maximum Skewness	Accuracy [%]	86.8	92.4	94.8	95.7	96.2
	Training time [s]	818.3	593.6	313.4	208.5	178.8
	Precision [%]	79.5	85.1	86.8	88.2	90.5
	Recall [%]	86.8	92.4	94.8	95.6	96.2
	F1 score [%]	83.0	88.6	90.6	91.8	93.3
	Data size [Mb]	201	144	126	112	106

An impressive improvement of $10.8 \pm 0.7\%$ in accuracy (from $86.8 \pm 0.3\%$ to $96.2 \pm 0.4\%$) was achieved, showing a significant enhancement even after selecting the optimal feature combination. A notable surge in accuracy was observed from 100% to 90%, followed by convergence beyond this threshold. Simultaneously, the training time decreased progressively by 78.1% (from 818.3 ± 15.4 s to 178.8 ± 12.1 s), demonstrating the efficiency of the filtering process. However, at Level 6 (denoted as (2) in Figure 5), a slight increase in training time (28 s) was observed, despite an additional 0.6% improvement in accuracy. Notably, the percentage improvements reported herein represent the ratio of change relative to the initial value rather than the absolute difference between the two values. This divergence may result from increased data complexity or density, necessitating greater computational effort for model optimization, possibly resulting from enhanced feature interactions or a more refined data distribution.

The visualization of the feature space further supports these findings. As filtering progressed, the separation between normal and fault features became more pronounced, with the elimination of low-density, scattered points contributing to enhanced classification performance. Density-based visualization revealed that high-density regions were concentrated at the core of the feature volumes. By eliminating low-density points from the periphery, the classification accuracy was improved by isolating meaningful patterns, as shown in Figure 5.

In addition to improvements in accuracy and training time, the filtering process also reduces the effective data size required for model training. As shown in Table 2, the dataset size decreased from 201 Mb at 100.0% to 106 Mb at 95.0%, representing a reduction of nearly 47.2%. This significant compression of the feature space was achieved without compromising classification performance, which highlights the lightweight nature of the DBFSO method. By requiring fewer data points and lower storage capacity, DBFSO offers a practical advantage for real-time industrial applications in resource-constrained settings.

A comparative analysis of the improvements in accuracy and training time with and without the DBFSO method for the four cases is shown in Figure 6. To enhance clarity and facilitate comparisons between the subplots in Figure 6a–d, a standardized y-axis range was applied, ensuring a consistent range difference (for example, 50% and 1500 s). In Case 1-1 (Figure 6a), the accuracy increased by $10.8 \pm 0.7\%$ (from $86.8 \pm 0.3\%$ to $96.2 \pm 0.4\%$), whereas the training time decreased by 78.1% (from 818.3 ± 15.4 s to 178.8 ± 12.1 s). Similarly, in Case 3-1 (Figure 6c), the accuracy increased by $15.3 \pm 0.3\%$ (from $81.5 \pm 0.2\%$ to $94.0 \pm 0.1\%$), whereas the training time decreased by 74.7% (from 843.5 ± 30.2 s to 213.5 ± 18.9 s). These results demonstrate substantial enhancements in both accuracy and computational efficiency when utilizing optimal combinations.

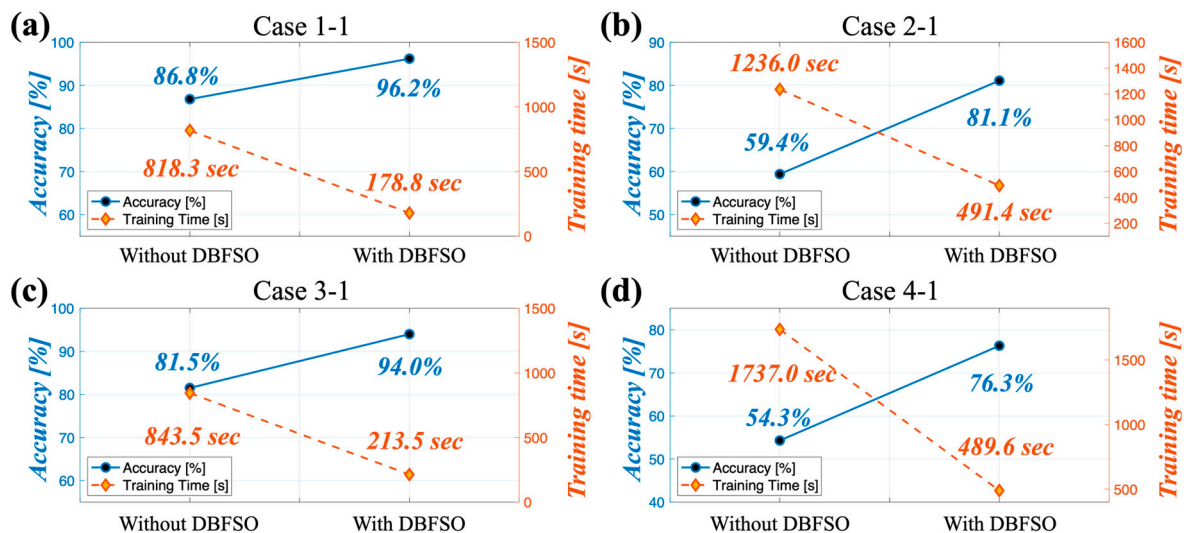


Figure 6. Results of improvement in accuracy and training time with and without the DBFSO method (100% and 95%): (a) Case 1-1, (b) Case 2-1, (c) Case 3-1, and (d) Case 4-1.

In the suboptimal combination cases (b) and (d), the accuracy was significantly improved. In Case 2-1 (Figure 6b), the accuracy increased by $36.5 \pm 0.6\%$ (from $59.4 \pm 0.4\%$ to $81.1 \pm 0.2\%$), whereas the training time decreased by 60.2% (from 1236.0 ± 12.8 s to 491.4 ± 32.9 s). Similarly, in Case 4-1 (Figure 6d), the accuracy increased by $40.5 \pm 0.7\%$ (from $54.3 \pm 0.5\%$ to $76.3 \pm 0.2\%$), whereas the training time decreased by 71.8% (from 1737.0 ± 35.7 s to 489.6 ± 10.2 s).

The suboptimal cases (b) and (d) demonstrated a significantly greater improvement in accuracy than the optimal combination cases. Specifically, the accuracy improvements in Case 2-1 ($36.5 \pm 0.6\%$) and Case 4-1 ($40.5 \pm 0.7\%$) were substantially greater than those observed in Case 1-1 ($10.8 \pm 0.7\%$) and Case 3-1 ($15.3 \pm 0.3\%$). This consistent trend indicates that the DBFSO method is particularly effective in enhancing feature selection and classification performance in suboptimal scenarios, as shown in Figures S5–S7 and Tables S3–S5 of the Supplementary Material. The results provide compelling evidence that the DBFSO method can unlock substantial accuracy gains, particularly in situations where the initial feature set is less than optimal. This underscores its ability to refine model performance under such conditions.

Overall, across all four cases, the DBFSO method consistently improved both accuracy and training time. On average, the accuracy improved by 25.8%, while the training time was reduced by 71.2%. It should be noted that the reduced training time primarily results from the smaller number of training samples after density-based filtering. The key contribution of DBFSO lies in simultaneously compressing the dataset by removing low-density samples and enhancing the separability of the remaining feature space, which together lead to improved classification performance.

4.2. Noise-Resilience Analysis

To simulate real-world conditions and evaluate the robustness of tension data analysis, Gaussian noise was selected as the noise model owing to its widespread utilization in signal processing and its ability to closely mimic random measurement errors. Gaussian noise, characterized by a normal distribution, properly aligns with the natural randomness observed in experimental data [53,54]. In this study, Gaussian noise equivalent to 5% of the signal amplitude was added to the tension data (refer to Figure S8 in the Supplementary Material). This realistic yet controlled simulation of noise was used to evaluate data robustness and the effectiveness of subsequent analytical methods.

The introduction of noise into the feature space (Case 1-2) resulted in significant alterations to the outer shape of the data compared with the noise-free scenario (Case 1-1). Specifically, the inclusion of noise increased the number of low-density points, highlighting the light-colored high-density points in the feature space. This effect is illustrated in Figure 7b, in which the boundary of the feature distribution demonstrates notable differences relative to that in Figure 7a. These discrepancies can be attributed to the increased number of outliers (low-density points) influencing the density calculations.

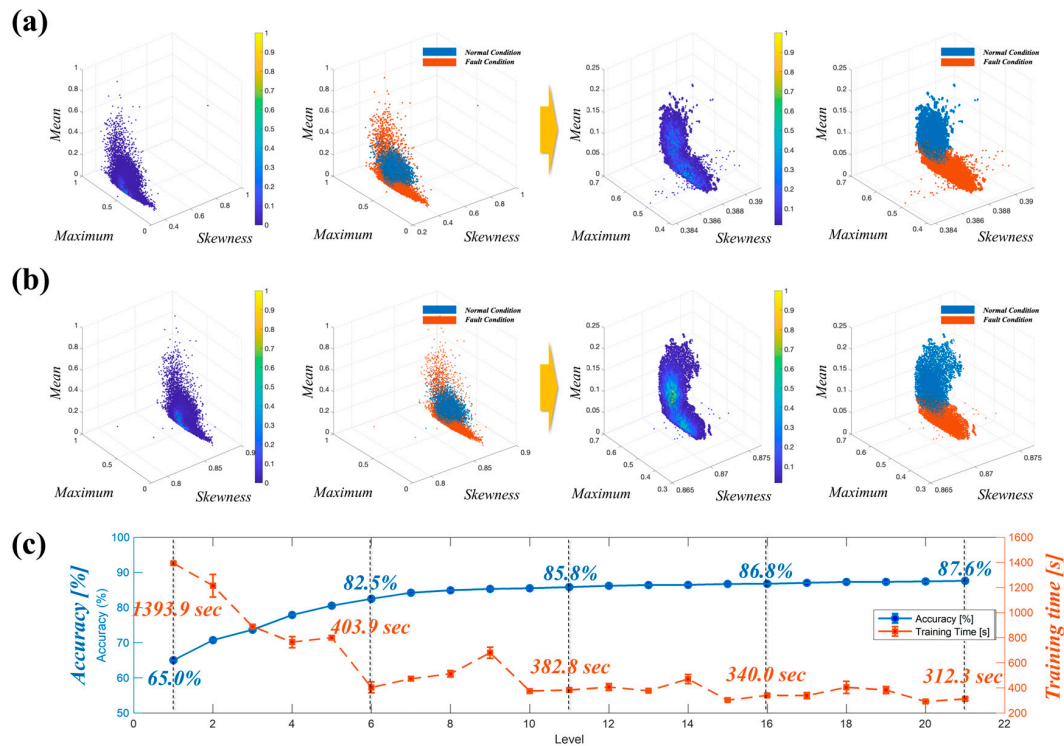


Figure 7. Visualization of the feature space for (a) Case 1-1, comparing raw data and the 95% filtered dataset, (b) Case 1-2, demonstrating raw data and the corresponding 95% filtered dataset, and (c) Case 1-2, with results categorized by DBFSO levels.

Despite these modifications, the classification accuracy exhibited a trend similar to that observed in Case 1-1, but with significant gains as the filtering level increased. Specifically, the accuracy increased by $34.8 \pm 0.3\%$, from $65.0 \pm 0.2\%$ to $87.6 \pm 0.1\%$, as shown in Table 3. Beyond improvements in classification performance, DBFSO also substantially reduces computational demands and data requirements. As indicated in Table 3, the training time decreased from 1393.3 ± 10.5 s to 312.3 ± 15.8 s with increasing filtering, while the dataset size was reduced from 206 Mb to 101 Mb, corresponding to a 51.0% reduction. This exceeds the 47.2% reduction observed in Case 1-1, which is attributable to the removal of added noise. Overall, the reductions—over 75% in training time and more than 50% in storage requirements—highlight the lightweight nature of the proposed framework. As it minimizes both computational and memory burdens without compromising diagnostic accuracy, DBFSO is particularly well suited for implementation in resource-constrained industrial monitoring systems. These results further demonstrate the method's effectiveness in suppressing noise while preserving the most informative data points.

Furthermore, the training time displayed distinct patterns. Although it decreased substantially overall by 77.6% (from 1393.9 ± 10.5 s to 312.3 ± 12.3 s), a noticeable increase in training time was observed at the 9th filtering level. This anomaly may be attributed to the increased computational intensity required to process outliers introduced by noise. Moreover, as additional low-density points were filtered, brighter (high-density) points

became more prominent in the feature space, further emphasizing the core structure of the data (Figure 7c).

Table 3. Classification results of Case 1-2 with noise by filtering level: optimal combination.

Optimal Combination	Parameter	DBFSO by Percentage				
		100.0%	98.75%	97.5%	96.25%	95.0%
Mean Maximum Skewness	Accuracy [%]	65.0	82.5	85.8	86.8	87.6
	Training time [s]	1393.9	403.9	382.8	340.0	312.3
	Precision [%]	58.2	75.8	78.1	79.5	80.6
	Recall [%]	65.0	82.5	85.8	86.8	87.6
	F1 score [%]	61.41	79.0	81.8	83.0	83.9
	Data size [Mb]	206	131	125	114	101

The results of the four cases from the perspectives of accuracy and training time are shown in Figure 8. The y -axis range was standardized across all plots by ensuring a uniform range difference (such as 50% and 2500 s). In the presence of noise, the proposed filtering method demonstrated a substantial improvement in both classification accuracy and computational efficiency across all the cases (detailed results in Figures S9–S11 and Tables S6–S8, Supplementary Material). In Case 1-2, the accuracy increased by $34.8 \pm 0.3\%$ (from $65.0 \pm 0.2\%$ to $87.6 \pm 0.1\%$), whereas the training time was reduced by 77.6% (from 1393.9 ± 10.5 s to 312.3 ± 15.8 s). Similarly, Case 3-2 demonstrated a $36.4 \pm 0.3\%$ increase in accuracy (from $60.1 \pm 0.1\%$ to $82.0 \pm 0.2\%$) and a 55.4% decrease in training time (from 1096.2 ± 98.2 s to 488.7 ± 32.1 s). In Case 2-2, the accuracy increased by $42.9 \pm 0.2\%$ (from $52.9 \pm 0.1\%$ to $75.6 \pm 0.1\%$), accompanied by an 83.3% decrease in training time (from 2344.6 ± 12.8 s to 391.9 ± 10.9 s). Finally, Case 4-2 demonstrated a $38.6 \pm 0.4\%$ improvement in accuracy (from $49.8 \pm 0.2\%$ to $69.0 \pm 0.2\%$) and an 80.5% decrease in training time (from 2328.0 ± 10.8 s to 453.7 ± 15.2 s).

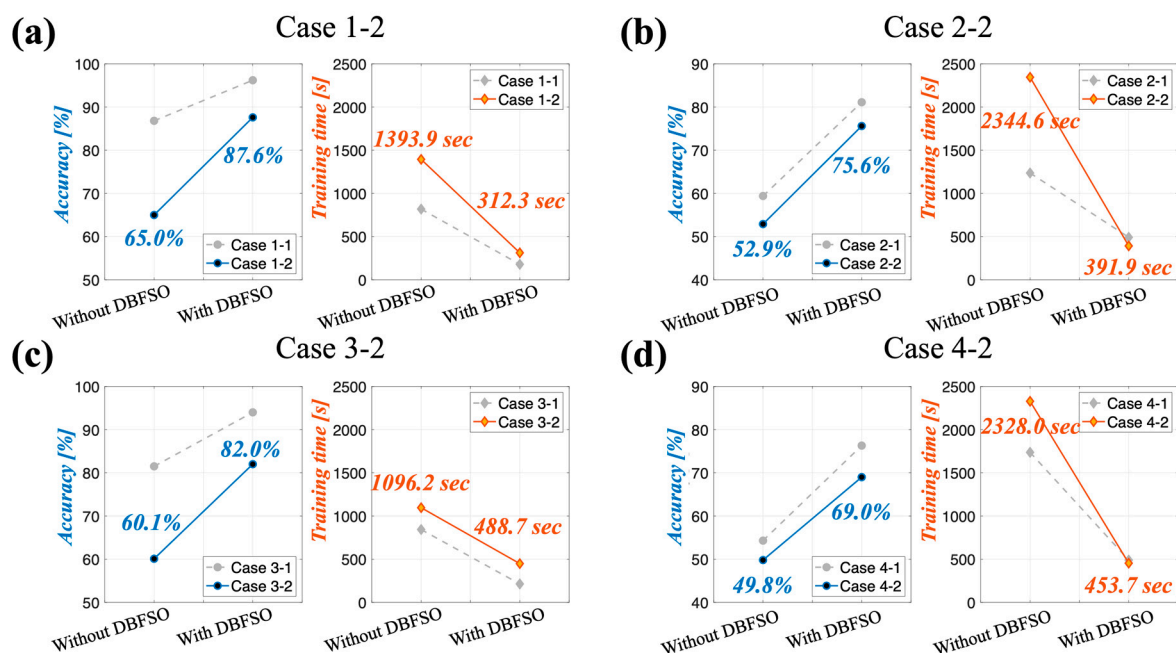


Figure 8. Results of improvement in accuracy and training time with and without the DBFSO method (100% and 95% CI): (a) Case 1-2, (b) Case 2-2, (c) Case 3-2, and (d) Case 4-2.

The average improvements in accuracy and training time were calculated for Cases 1-2 through 4-2 to assess the overall performance of the proposed filtering method. Across these cases, the accuracy increased by an average of 38.2%, whereas the training time decreased by an average of 74.2%. Compared with the previous results for Cases 1-1 to 4-1, which demonstrated an average accuracy improvement of 25.8% and training time reduction of 71.2%, the enhancements achieved with the filtering method were more significant.

This analysis underscores the robustness of the DBFSO method in managing noisy data, yielding consistent improvements in both classification accuracy and computational efficiency. The significant performance gains observed in Cases 1-2 to 4-2 indicate that DBFSO effectively refined the feature space, particularly in challenging conditions with high noise levels and complex data distributions. These findings underscore the potential of this method for optimizing classification models while reducing computational overhead, rendering it applicable across diverse fault diagnosis scenarios.

4.3. Additional Classifier-Based Validation

To isolate the performance improvement attributable solely to the DBFSO filtering mechanism, the primary validation in this study employed a unified classifier architecture based on an SVM model. This ensured that differences in classification performance were not confounded by model-dependent behaviors. However, to further examine the influence of classifier choice and verify whether the benefit of DBFSO is model-specific or classifier-agnostic, additional analyses were conducted using k-NN, decision tree, and ensemble bagged tree classifiers.

For consistency, DBFSO was applied with the same 5% filtering ratio used in the previous experiments. As summarized in Table 4, the baseline accuracy differed across the classifiers, which is expected due to their inherent decision mechanisms. Nevertheless, an important observation is that all classifiers showed consistent accuracy improvements after applying DBFSO filtering. Specifically, accuracy gains of 9.4%, 8.4%, 11.2%, and 9.7% were observed for SVM, k-NN, decision tree, and ensemble bagged trees, respectively.

Table 4. Classification accuracy with and without DBFSO for different classifiers (Case 1-1).

Classifier	Without DBFSO [%]	With DBFSO [%]	Improvement [%]
SVM	86.8	96.2	+9.4
k-NN (k = 5)	84.2	92.6	+8.4
Decision Tree	78.5	89.7	+11.2
Ensemble Bagged Trees	86.1	95.8	+9.7

These results confirm that DBFSO is not tied to a particular classification model; rather, it systematically improves the distributional quality of the training samples regardless of the classifier. This reinforces the conclusion that DBFSO serves as a robust, classifier-agnostic feature space refinement strategy.

4.4. Generalization of DBFSO to Bearing Fault Diagnosis

To assess the generalization capability of the DBFSO method, a bearing fault dataset obtained from a 7.5 kW servomotor (Hyosung Power & Industrial Systems) was utilized. The dataset includes three fault conditions: normal operation, bearing cage fault, and unbalanced fault. Detailed information on the data acquisition settings, experimental conditions, and visualization of the acquired data is provided in Table S9 and Figure S12 (Supplementary Material). The vibration signals were recorded using a commercial vibration measurement module at a sampling rate of 5000 Hz, with each condition tested in three independent trials to ensure reliability. The three fault scenarios investigated were as

follows: Case A (normal vs. bearing cage fault), Case B (normal vs. unbalanced fault), and Case C (bearing cage fault vs. unbalanced fault).

Figure 9 shows the accuracy and training time for bearing fault diagnosis, with and without the DBFSO method. The x -axis data of the acquired dataset were processed to facilitate comparison. For a unified comparison, the optimal and suboptimal feature combinations were selected for each case based on the FDC algorithm, and the entire process, including the 70/15/15 training/validation/test data split and consistent hyperparameter tuning via a grid search with fivefold cross-validation, was set identically to the methodology described in the previous sections.

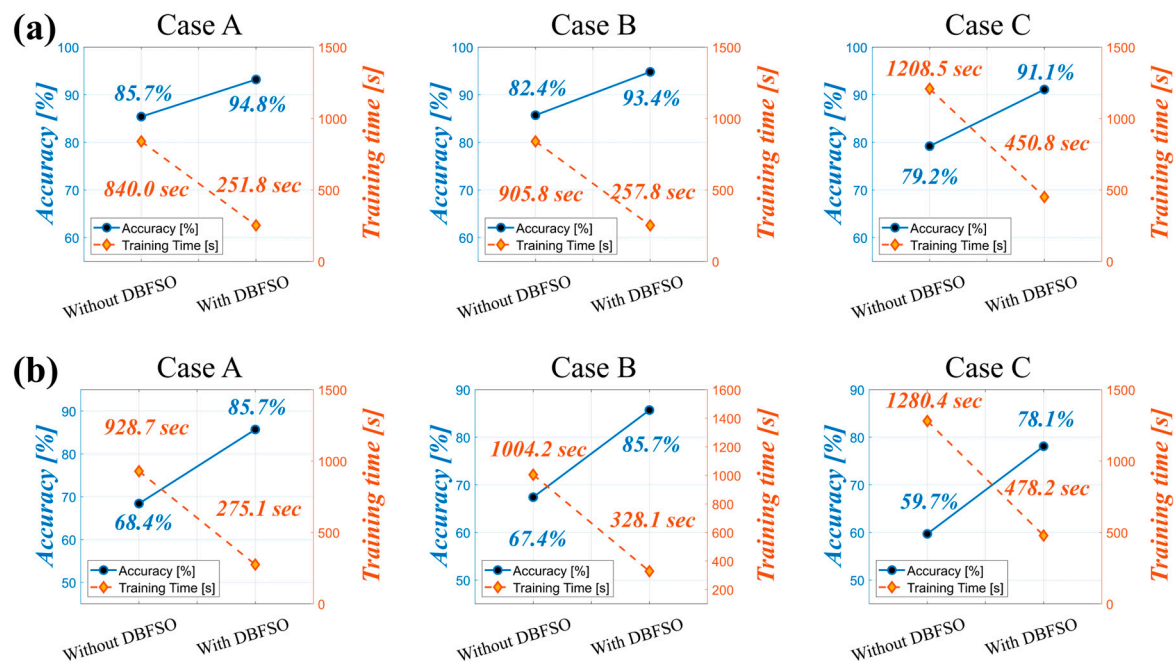


Figure 9. Improvements in accuracy and training time with and without the DBFSO method (100% and 95%) for bearing fault diagnosis: (a) optimal feature combination and (b) suboptimal feature combination.

As shown in Figure 9a, the results from the optimal feature combination consistently demonstrated high baseline accuracy, which the DBFSO method further enhanced. Specifically, the accuracy increased from $85.7 \pm 0.2\%$ and $94.8 \pm 0.1\%$, $82.4 \pm 0.1\%$, to $93.4 \pm 0.2\%$, and from $79.2 \pm 0.2\%$, to $91.1 \pm 0.1\%$, whereas the training time decreased from 840 ± 12.8 s to 251.8 ± 20.8 s, 905.8 ± 35.4 s to 257.8 ± 19.8 s, and from 1208.5 ± 80.4 s to 450.7 ± 17.2 s, respectively.

For the suboptimal feature combinations, Figure 9b illustrates a greater impact from the DBFSO method. Here, the initial accuracy was notably lower, but the framework successfully improved it. Specifically, the accuracy increased from $68.4 \pm 0.2\%$ to $85.7 \pm 0.1\%$, and the training time decreased from 928.7 ± 20.4 s to 275.1 ± 12.4 s. Similarly, in the other suboptimal cases (Cases B and C), the accuracy increased from $67.4 \pm 0.1\%$ to $85.7 \pm 0.3\%$ and from $59.7 \pm 0.2\%$ to $78.1 \pm 0.1\%$, whereas the training time decreased from 1004.2 ± 30.7 s to 328.1 ± 23.8 s and from 1208.4 ± 49.8 s to 478.2 ± 21.6 s, respectively. These significant gains in accuracy for suboptimal cases highlight the particular effectiveness of the DBFSO method in challenging scenarios.

The detailed confusion matrices, presented in Figure S13 (Supplementary Material), provide a deeper analysis of our results. These matrices confirm that the DBFSO method does not merely improve the majority class but effectively reduces misclassifications across all classes, particularly for more challenging fault scenarios.

The application of the DBFSO method in bearing fault diagnosis has demonstrated significant improvements in classification accuracy and computational efficiency. Specifically, the average accuracy increment was 20.4%, whereas the training time decreased by 67.5% across various fault scenarios. These enhancements underscore the robustness and adaptability of the DBFSO method, highlighting its potential for widespread application in diverse industrial contexts beyond R2R systems, where efficient and accurate fault diagnosis is essential.

4.5. Comparison with Existing Filtering Techniques

Although DBFSO has demonstrated significant improvements in classification accuracy and computational efficiency, it is important to compare its effectiveness with that of commonly used baseline filtering techniques, such as the median filter [55,56] and Isolation Forest [57]. These methods were selected to represent both conventional data preprocessing and recently suggested outlier detection techniques.

The median filter operates at the raw data level, replacing each data point with the median of its neighboring values to effectively mitigate impulsive noise. Isolation Forest is an unsupervised learning algorithm specifically designed for anomaly detection. It works by directly isolating outliers, which are defined as points that are few and different, by building an ensemble of decision trees. Both of these approaches impact classification accuracy indirectly: the median filter improves the quality of the raw sensor data, and Isolation Forest by removing outliers prior to model training.

By contrast, DBFSO directly optimizes the feature space by filtering the extracted features in a structured three-dimensional space rather than preprocessing raw sensor signals. This distinction is critical, as DBFSO selectively eliminates low-density points from a coordinated feature set, ensuring that only the most representative data remain for classification. DBFSO significantly improves the distinguishability between normal and faulty conditions by refining the input feature space itself rather than solely focusing on the raw signals. This enhancement has a profound effect on the performance of classification models. The targeted filtering process employed by DBFSO effectively reduces the impact of outliers and sparsely distributed data points, which are challenges that conventional methods and even newer algorithms such as Isolation Forest often struggle to address effectively in this context.

The comparison results between DBFSO and the selected baseline filtering techniques are shown in Figure 10. The resulting feature spaces from each filtering technique—median and Isolation Forest—are presented to visualize their impact. Although the median filter shows minor refinement of the raw data and Isolation Forest removes some outliers, its ability to create a clearly separated feature space remains limited. By contrast, the DBFSO method yields a noticeably separated feature space, effectively enhancing the distinction between the two data classes.

The filtering processes are represented in steps, as shown in the flowchart, and the average classification accuracy values for Case 1-2 to Case 4-2 are illustrated in Figure 10. DBFSO consistently outperformed both the median filter and Isolation Forest in enhancing classification performance. Although these conventional and suggested techniques provide minor improvements by reducing raw signal noise or removing outliers, their impact remains limited, as they do not account for structural relationships within the feature space. By contrast, DBFSO directly optimizes the constructed feature space by filtering out low-density, less informative data points, resulting in a substantial improvement in class separability. To the best of our knowledge, no existing method has applied density-based filtering after the construction of a multidimensional feature space in the context of fault diagnosis. This highlights the novelty of DBFSO and its distinct advantages in

reducing noise and refining the classification-ready data representation, ultimately leading to significantly improved model performance in practical diagnostic tasks.

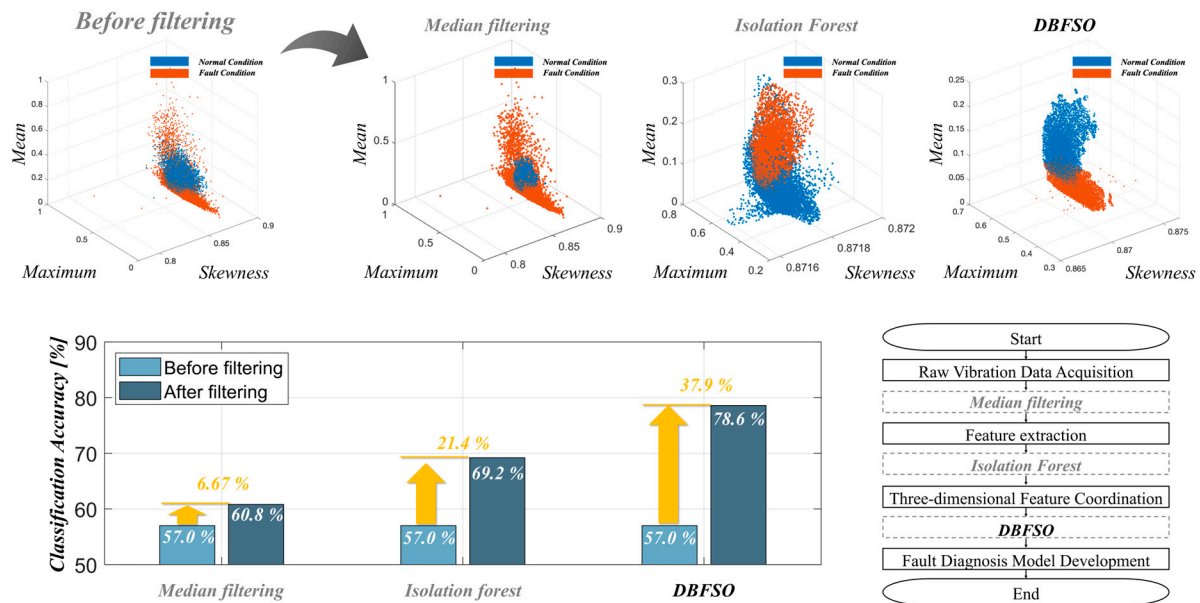


Figure 10. Comparison with existing filtering methods.

5. Conclusions and Future Work

5.1. Summary of Major Findings

Rapid advancements in smart manufacturing underscore the critical importance of precise fault diagnosis and robust classification models to maintain the efficiency and reliability of industrial systems. In this context, the selection of optimal features and effective data filtering are crucial for achieving top-tier diagnostic performance, particularly in challenging environments with noisy or sparsely distributed data.

This study introduced a novel method known as DBFSO that enhances classification accuracy by addressing the key limitations of conventional feature selection approaches. By constructing a three-dimensional feature space that incorporates data from normal and abnormal conditions, the proposed method adeptly captured the underlying fault patterns. Leveraging the k-NN algorithm, the density of each feature point was calculated, enabling the identification of low-density points that exerted minimal influence on classification performance. These low-density points were subsequently filtered based on a volumetric percentile threshold, refining the feature space to focus on the most critical data.

The proposed DBFSO method demonstrated significant advantages over existing approaches. It enhanced classification performance under optimal feature configurations and also exhibited superior robustness in suboptimal and noisy environments where conventional methods often struggle. By dynamically refining the feature space, the method improved the reliability of fault detection even when samples were sparsely distributed or contaminated by noise. To validate these improvements, a statistical analysis using the Wilcoxon signed-rank test confirmed that the gains in both accuracy and computational efficiency were consistently significant across all experimental scenarios, with p -values below 0.05. This consistent statistical significance, combined with the performance enhancements observed in multiple classifier architectures, reinforces the conclusion that DBFSO provides a stable and reproducible improvement in diagnostic performance.

DBFSO yielded substantial performance gains across the evaluated datasets, achieving up to a 38.2% increase in classification accuracy and a 74.2% reduction in training time. The additional evaluations using SVM, k-NN, decision tree, and ensemble models further

confirmed that the improvements consistently appeared across diverse classifier architectures, indicating that the effect of DBFSO originates from the refinement of the feature space rather than any model-specific behavior.

Furthermore, the method consistently maintained its effectiveness even when the data were noisy, sparse, or sampled from suboptimal operating conditions—a scenario in which conventional classification pipelines typically experience significant degradation. By filtering low-density, low-informative samples, DBFSO enhanced class separability and stabilized decision boundaries, thereby improving diagnostic reliability in environments characterized by fluctuating data quality, such as real industrial systems.

The performance gains of the DBFSO framework can be theoretically attributed to its selective removal of low-density samples in the feature space. Low-density regions often correspond to noise, outliers, or ambiguous borderline points that compromise classifier robustness [28,58]. By systematically filtering out such samples based on k-NN-derived density scores, DBFSO effectively enhances feature space purity and class separability. Prior studies in supervised learning and fault diagnosis have demonstrated that density-based instance selection not only reduces the distortion of decision boundaries but also improves generalization capability [59,60]. The improvements observed in our R2R and bearing fault datasets—up to 38.2% average accuracy gain and 74.2% reduction in training time—underscore the practical benefits of this mechanism. This evidence confirms that optimizing input data quality through density-based strategies is a key factor for achieving robust diagnostic performance, aligning with broader principles of noise reduction and data preprocessing in intelligent fault diagnosis.

Beyond the density-based explanation discussed above, it is also important to clarify how DBFSO differs from conventional feature space optimization methods. Feature selection and dimensionality reduction techniques optimize the feature space by removing irrelevant variables or constructing low-dimensional representations through transformations such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA). These approaches operate at the feature level, altering the structure or dimensionality of the input variables.

In contrast, DBFSO performs optimization at the instance level by removing low-density samples that may represent noise, outliers, or ambiguous boundary instances. Rather than modifying feature variables, DBFSO improves the distributional quality of samples within the existing feature space, thereby enhancing class separability and reducing decision-boundary distortion. This distinction indicates that DBFSO addresses a different aspect of feature space degradation compared to feature-level techniques and can therefore serve as a complementary strategy rather than a direct alternative.

This clarification positions DBFSO within the broader landscape of feature space optimization methods and demonstrates that its contribution lies in improving sample quality, not altering feature dimensionality. Thus, DBFSO provides a distinct yet compatible mechanism for strengthening the robustness of classification in industrial fault diagnosis.

5.2. Implications for Industrial Applications

The proposed DBFSO method shows significant promise for improving diagnostic capabilities in various industrial applications. By effectively addressing challenges related to noise and sparsely distributed data, this method enhances the reliability and accuracy of fault detection systems. Industries relying on rotary components, such as wind energy, automotive manufacturing, and production systems, benefit considerably from this approach. Additionally, the scalability of DBFSO ensures its applicability to large datasets commonly encountered in industrial environments, offering solutions to minimize downtime, reduce maintenance costs, and improve overall productivity.

The lightweight nature is also a key advantage of this approach. Unlike deep learning-based models, which require large-scale datasets, high-dimensional inputs, and substantial computational resources, DBFSO operates directly in a three-dimensional feature space using simple k-NN density estimation. This results in significant reductions in training time (up to 78%) and enables real-time applicability in production environments with limited computing resources. These characteristics are particularly valuable for embedded monitoring systems and online diagnostics.

Furthermore, although this study focused on rotary components in R2R systems, the principles of the DBFSO method can be extended to other sectors. The application to bearing fault datasets highlights its broader applicability to non-rotary systems, including structural health monitoring, medical diagnostics, and financial data analysis—areas where noise and data sparsity present similar challenges to classification accuracy.

5.3. Theoretical Considerations and Potential Bias

From a theoretical perspective, DBFSO builds on the assumption that the local point density in the feature space reflects the informativeness of samples. This provides a systematic mechanism for enhancing class separability while also suggesting directions for refinement rather than strict limitations. For example, k-NN-based density estimation is most effective when local neighborhoods are relatively uniform; for datasets with anisotropic distributions or irregularly shaped clusters, adaptive density estimation strategies could be explored to further enrich fault pattern representation.

Although the current study emphasizes three-dimensional feature spaces for computational efficiency and clarity of interpretation, and the paper shows three-dimensional results for better visualization and understanding, the framework is readily extendable to higher-dimensional spaces. This enables DBFSO to capture more complex multidimensional interactions, and our results confirm its generalizability beyond three dimensions.

DBFSO also leverages contrasts between high-density and sparse regions. Although localized noise or irrelevant signals may occasionally appear as dense clusters, complementary filtering criteria can be incorporated to increase resilience. Likewise, when supervised labels are imbalanced or imperfect, hybrid density–label metrics could further strengthen selection robustness. These considerations highlight opportunities to broaden the applicability of DBFSO and ensure its adaptability across diverse industrial datasets.

5.4. Limitations and Future Research Directions

The DBFSO method shows promise in enhancing fault classification by filtering out noisy or sparsely distributed data points, thereby improving feature separability and model robustness. However, application-specific tuning may be required because of variations in data characteristics, machinery types, and environmental conditions. Key parameters, such as the number of nearest neighbors (k) and the density threshold criteria, must currently be selected manually for each dataset, and reliance on heuristic selection—particularly for percentile-based thresholds—may limit generalization and consistency across use cases.

To overcome these limitations, future research should focus on principled, systematic approaches for automated parameter optimization. Strategies such as adaptive methods based on cross-validation or heuristic search algorithms can dynamically adjust thresholds in response to dataset variability and noise levels. Furthermore, more rigorous approaches, including reinforcement learning or Bayesian optimization, hold significant potential for enabling data-driven, adaptive threshold selection, thereby improving the robustness, generalizability, and applicability of DBFSO across diverse industrial environments.

To further enhance practicality in industrial environments, DBFSO can be integrated with existing machine learning frameworks to improve performance under real-world

conditions. For example, combining DBFSO with neural network-based models, such as autoencoders or convolutional neural networks, could help mitigate the impact of noisy inputs by refining the feature space prior to model training. Additionally, transfer learning could be employed to adapt DBFSO-enhanced models to new datasets or operating conditions with minimal retraining, increasing their utility across similar machines or production lines.

Although DBFSO was developed for scenarios with known normal and abnormal labels, its ability to reduce noise and enhance class separability indirectly supports generalization when training data are limited. This makes it a promising option for practical applications where labeled data are scarce or expensive to collect. Nonetheless, current limitations such as the handling of unknown fault types and cross-system variation present opportunities for future extensions. Overall, DBFSO provides a lightweight and interpretable foundation that can be further refined to meet the demands of scalable, reliable fault diagnosis in industrial settings.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/math13243984/s1>: Figure S1. Classification results based on SVM algorithm with varying k parameter (5, 10, 100, $\sqrt{N}/2$, $\sqrt{N}$, $2\sqrt{N}$): (a) accuracy before and after 95% DBFSO filtering (b) enhancement of accuracy, showing 9.5% and 22.5% enhancement of accuracy, and less than 0.5% standard deviation across k parameters. Figure S2. Acquired tension data: (a) normal, (b) eccentricity in Span 2, and (c) eccentricity in Span 3. Figure S3. Case 1-1 results categorized by DBFSO level: 95.0%, 93.75%, 92.5%, 91.25%, and 90.0%. Figure S4. Case 1-1 results categorized by DBFSO level: from 100% to 90%. Figure S5. Case 2-1 results categorized by DBFSO level: (top) density-based visualization, (middle) accuracy and training time, and (bottom) condition-based visualization. Figure S6. Case 3-1 results categorized by DBFSO level: (top) density-based visualization, (middle) accuracy and training time, and (bottom) condition-based visualization. Figure S7. Case 4-1 results categorized by DBFSO level: (top) density-based visualization, (middle) accuracy and training time plots, and (bottom) condition-based visualization. Figure S8. Tension data before and after simulated noise: (a) normal, (b) eccentricity in Span 2, and (c) eccentricity in Span 3. Figure S9. Visualization of the feature space for (a) Case 2-1, which compares raw data and the 90% filtered dataset; (b) Case 2-2, which illustrates raw data and the corresponding 90% filtered dataset; and (c) Case 2-2, with results categorized according to DBFSO levels. Figure S10. Visualization of the feature space for (a) Case 3-1, comparing raw data and 90% filtered dataset; (b) Case 3-2, indicating raw data and corresponding 90% filtered dataset; and (c) Case 3-2, with results categorized based on DBFSO levels. Figure S11. Visualization of the feature space for (a) Case 4-1, comparing raw data and 90% filtered dataset; (b) Case 4-2, illustrating raw data and corresponding 90% filtered dataset; and (c) Case 4-2, with results categorized according to DBFSO levels. Figure S12. Acquired vibration data: (a) normal, (b) bearing cage fault, and (c) unbalanced fault. Figure S13. Confusion matrix for bearing fault classification results: (a) optimal feature combination and (b) suboptimal feature combination. Table S1. Statistical feature variables and extraction methods. Table S2. Specifications of loadcell used in data acquisition. Table S3. Classification results of Case 2-1 by filtering level: Suboptimal combination. Table S4. Classification results for Case 3-1 by filtering level: Optimal combination. Table S5. Classification results of Case 4-1 by filtering level: Suboptimal combination. Table S6. Classification results for Case 2-2 with noise by filtering level: Optimal combination. Table S7. Classification results for Case 3-2 with noise based on filtering level: Optimal combination. Table S8. Classification results for Case 4-2 with noise based on filtering level: Optimal combination. Table S9. Specifications of the motor and accelerometer.

Author Contributions: J.Y.: Conceptualization, Data curation, Methodology, Software, Visualization, Writing—original draft. K.-C.C.: Software, Visualization. W.K.: Validation, Methodology. C.K.: Software, Visualization. H.S.K.: Methodology, Validation. C.L.: Funding acquisition, Resources, Project administration, Supervision, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Industrial Fundamental Technology Development Program (No. RS-2024-00507660, “Development of autonomous operating system technology to improve productivity in synthetic leather manufacturing processes”) funded by the Ministry of Trade, Industry & Energy (MOTIE) of Korea and the Korean Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (RS-2024-00435502, Human Resource Development Program for Industrial Innovation (Global)).

Data Availability Statement: The original contributions presented in this study are included in the Supplementary Material. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

DBFSO	Density-based feature space optimization
DPC	Density peak clustering
FDC	Feature variable dimensional coordination
LOF	Local outlier factor
RUL	Remaining useful life
SVM	Support vector machine
PCA	Principal component analysis
LDA	Linear discriminant analysis

References

1. Aldrini, J.; Chihi, I.; Sidhom, L. Fault diagnosis and self-healing for smart manufacturing: A review. *J. Intell. Manuf.* **2024**, *35*, 2441–2473. [\[CrossRef\]](#)
2. Vogl, G.W.; Weiss, B.A.; Helu, M. A review of diagnostic and prognostic capabilities and best practices for manufacturing. *J. Intell. Manuf.* **2019**, *30*, 79–95. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent manufacturing in the context of Industry 4.0: A review. *Engineering* **2017**, *3*, 616–630. [\[CrossRef\]](#)
4. Chen, W.; Qiu, Y.; Feng, Y.; Li, Y.; Kusiak, A. Diagnosis of wind turbine faults with transfer learning algorithms. *Renew. Energy* **2021**, *163*, 2053–2067. [\[CrossRef\]](#)
5. Paya, B.A.; Esat, I.I.; Badi, M.N.M. Artificial neural network-based fault diagnostics of rotating machinery using wavelet transforms as a preprocessor. *Mech. Syst. Signal Process.* **1997**, *11*, 751–765. [\[CrossRef\]](#)
6. Venkatesan, S.; Manickavasagam, K.; Tengenai, N.; Vijayalakshmi, N. Health monitoring and prognosis of electric vehicle motor using intelligent-digital twin. *IET Electric Power Appl.* **2019**, *13*, 1328–1335. [\[CrossRef\]](#)
7. Wang, X.; Lu, S.; Chen, K.; Wang, Q.; Zhang, S. Bearing fault diagnosis of switched reluctance motor in electric vehicle powertrain via multisensor data fusion. *IEEE Trans. Ind. Inform.* **2022**, *18*, 2452–2464. [\[CrossRef\]](#)
8. Yun, J.; Lee, S.; Jang, J.; Kim, M.; Kim, C.; Lee, C. Sensor-Efficient Estimation of Roll Misalignment via Side-to-Side Tension Differences in Roll-to-Roll Polymer Film Processing. *Polymers* **2025**, *17*, 2907. [\[CrossRef\]](#)
9. Kim, M.; Noh, J.; Lee, S.; Yun, J.; Lee, C. Mitigating strain deviation by investigating tension transfer mechanisms dependent on anode shape in roll-to-roll electrode fabrication. *Struct. Multidiscip. Optim.* **2025**, *68*, 184. [\[CrossRef\]](#)
10. Yun, J.; Lee, Y.; Kim, M.; Jo, J.; Lee, C. Thermal wrinkling in anode production: Defect analysis and mitigation strategies. *J. Manuf. Process.* **2025**, *141*, 81–92. [\[CrossRef\]](#)
11. Kooy, N.; Mohamed, K.; Pin, L.T.; Guan, O.S. A review of roll-to-roll nanoimprint lithography. *Nanoscale Res. Lett.* **2014**, *9*, 320. [\[CrossRef\]](#)
12. Maize, K.; Mi, Y.; Cakmak, M.; Shakouri, A. Real-time metrology for roll-to-roll and advanced inline manufacturing: A review. *Adv. Mater. Technol.* **2023**, *8*, 2200173. [\[CrossRef\]](#)
13. Noh, J.; Jo, M.; Kim, M.; Lee, C. Advanced taper tension profile for improved control performance in roll-to-roll winding processes. *Alex. Eng. J.* **2024**, *104*, 279–291. [\[CrossRef\]](#)
14. García, D.F.; López, J.M.; Suarez, F.J.; Garcia, J.; Obeso, F.; Gonzalez, J.A. A novel real-time fuzzy-based diagnostic system of roll eccentricity influence in finishing hot strip mills. *IEEE Trans. Ind. Appl.* **1998**, *34*, 1342–1350. [\[CrossRef\]](#)

15. Shin, K.H.; Jang, J.I.; Kang, H.K.; Song, S.H. Compensation method for tension disturbance due to an unknown roll shape in a web transport system. *IEEE Trans. Ind. Appl.* **2003**, *39*, 1422–1428. [\[CrossRef\]](#)
16. Fink, O.; Wang, Q.; Svensén, M.; Dersin, P.; Lee, W.J.; Ducoffe, M. Potential, challenges and future directions for deep learning in prognostics and health management applications. *Eng. Appl. Artif. Intell.* **2020**, *92*, 103678. [\[CrossRef\]](#)
17. Han, X.; Wang, Z.; Xie, M.; He, Y.; Li, Y.; Wang, W. Remaining useful life prediction and predictive maintenance strategies for multi-state manufacturing systems considering functional dependence. *Reliab. Eng. Syst. Saf.* **2021**, *210*, 107560. [\[CrossRef\]](#)
18. Liu, Q.; Dai, Z.; Lai, H.; Chen, M.; Huang, H.; Fu, J.; Hou, M.; Xu, X.; Wang, G.; Yan, J. A novel RUL prediction method for rolling bearing: TcLstmNet-CBAM. *Sci. Rep.* **2025**, *15*, 14055. [\[CrossRef\]](#)
19. Zraibi, B.; Mansouri, M.; Loukili, S.E. RUL prediction for lithium-ion batteries using improved-CGD hybrid model. *Int. J. Dyn. Control* **2025**, *13*, 237. [\[CrossRef\]](#)
20. Toothman, M.; Braun, B.; Bury, S.J.; Moyne, J.; Tilbury, D.M.; Ye, Y.; Barton, K. A digital twin framework for prognostics and health management. *Comput. Ind.* **2023**, *150*, 103948. [\[CrossRef\]](#)
21. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Universal domain adaptation in fault diagnostics with hybrid weighted deep adversarial learning. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7957–7967. [\[CrossRef\]](#)
22. Yuan, W.; Liu, F.; Gu, H.; Miao, F.; Zhang, F.; Jiang, M. Accuracy-Improved Fault Diagnosis Method for Rolling Bearing Based on Enhanced ESGMD-CC and BA-ELM Model. *Shock. Vib.* **2024**, *2024*, 8026402. [\[CrossRef\]](#)
23. Saeed, F.; Al-Sarem, M.; Al-Mohaimed, M.; Emara, A.; Boulila, W.; Alasli, M.; Ghabban, F. Enhancing Parkinson’s disease prediction using machine learning and feature selection methods. *Comput. Mater. Contin.* **2022**, *71*, 5639–5658. [\[CrossRef\]](#)
24. Sasikala, S.; Renuka, D.D.; Kumar, R. Feature selection methods and algorithms. In *Research Practitioner’s Handbook on Big Data Analytics*; Apple Academic Press: New York, NY, USA, 2023; pp. 71–111. [\[CrossRef\]](#)
25. Takci, H. Improvement of heart attack prediction by the feature selection methods. *Turk. J. Electr. Eng. Comput. Sci.* **2018**, *26*, 1–10. [\[CrossRef\]](#)
26. García-Gil, D.; Luengo, J.; García, S.; Herrera, F. Enabling Smart Data: Noise filtering in Big Data classification. *Inf. Sci.* **2019**, *479*, 135–152. [\[CrossRef\]](#)
27. Koziarski, M.; Krawczyk, B.; Woźniak, M. Radial-Based oversampling for noisy imbalanced data classification. *Neurocomputing* **2019**, *343*, 19–33. [\[CrossRef\]](#)
28. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. *ACM SIGMOD Rec.* **2000**, *29*, 93–104. [\[CrossRef\]](#)
29. Du, M.; Ding, S.; Jia, H. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowl.-Based Syst.* **2016**, *99*, 135–145. [\[CrossRef\]](#)
30. Guo, B.; Lu, M.; Fan, Y.; Wu, H.; Yang, Y.; Wang, C. A novel remote sensing monitoring index of salinization based on three-dimensional feature space model and its application in the Yellow River Delta of China. *Geom. Nat. Hazards Risk* **2023**, *14*, 95–116. [\[CrossRef\]](#)
31. Ma, Y.; Tashpolat, N. Remote sensing monitoring of soil salinity in Weigan River–Kuqa River Delta oasis based on two-dimensional feature space. *Water* **2023**, *15*, 1694. [\[CrossRef\]](#)
32. Yang, Y.; Wang, W.; Fu, H.; Kuo, C.-C.J. On supervised feature selection from high dimensional feature spaces. *APSIPA Trans. Signal Inf. Process.* **2022**, *11*, e31. [\[CrossRef\]](#)
33. Chen, Y.; Hu, X.; Fan, W.; Shen, L.; Zhang, Z.; Liu, X.; Du, J.; Li, H.; Chen, Y.; Li, H. Fast density peak clustering for large scale data based on kNN. *Knowl.-Based Syst.* **2020**, *187*, 104824. [\[CrossRef\]](#)
34. Cover, T.M.; Hart, P.E. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [\[CrossRef\]](#)
35. Shi, B.; Han, L.; Yan, H. Adaptive clustering algorithm based on kNN and density. *Pattern Recognit. Lett.* **2018**, *104*, 37–44. [\[CrossRef\]](#)
36. Tran, T.N.; Wehrens, R.; Buydens, L.M.C. KNN-kernel density-based clustering for high-dimensional multivariate data. *Comput. Stat. Data Anal.* **2006**, *51*, 513–525. [\[CrossRef\]](#)
37. Xing, W.; Bei, Y. Medical health big data classification based on KNN classification algorithm. *IEEE Access* **2020**, *8*, 28808–28819. [\[CrossRef\]](#)
38. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, KDD 1996, Portland, OR, USA, 2–4 August 1996*; AAAI Press: Washington, DC, USA, 1996; pp. 226–231.
39. Li, X.; Wang, L.; Wang, M.; Wen, C.; Fang, Y. DANCE-NET: Density-aware convolution networks with context encoding for airborne LiDAR point cloud classification. *ISPRS J. Photogramm. Remote Sens.* **2020**, *166*, 128–139. [\[CrossRef\]](#)
40. Tang, J.; Chen, Z.; Fu, A.W.; Cheung, D.W. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in Knowledge Discovery and Data Mining*; Chen, M.S., Yu, P.S., Liu, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 535–548. [\[CrossRef\]](#)

41. Wang, L.; Shao, Y. Fault feature extraction of rotating machinery using a reweighted complete ensemble empirical mode decomposition with adaptive noise and demodulation analysis. *Mech. Syst. Signal Process.* **2020**, *138*, 106545. [\[CrossRef\]](#)
42. Barandas, M.; Folgado, D.; Fernandes, L.; Santos, S.; Abreu, M.; Bota, P.; Liu, H.; Schultz, T.; Gamboa, H. TSFEL: Time series feature extraction library. *SoftwareX* **2020**, *11*, 100456. [\[CrossRef\]](#)
43. Khaire, U.M.; Dhanalakshmi, R. Stability of feature selection algorithm: A review. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 1060–1073. [\[CrossRef\]](#)
44. Lee, Y.; Park, B.; Jo, M.; Lee, J.; Lee, C. A quantitative diagnostic method of feature coordination for machine learning model with massive data from rotary machine. *Expert Syst. Appl.* **2023**, *214*, 119117. [\[CrossRef\]](#)
45. Ebrahimi, M.A.; Khoshtaghaza, M.H.; Minaei, S.; Jamshidi, B. Vision-based pest detection based on SVM classification method. *Comput. Electron. Agric.* **2017**, *137*, 52–58. [\[CrossRef\]](#)
46. Mustafa Abdullah, D.; Mohsin Abdulazeez, A. Machine Learning Applications based on SVM Classification: A review. *Qubahan Acad. J.* **2021**, *1*, 81–90. [\[CrossRef\]](#)
47. Kammoun, A.; Alouinifellow, M.S. On the precise error analysis of support vector machines. *IEEE Open J. Signal Process.* **2021**, *2*, 99–118. [\[CrossRef\]](#)
48. Prasetyo, R.T. Genetic algorithm to optimize k-nearest neighbor parameter for benchmarked medical datasets classification. *J. Online Inform.* **2020**, *5*, 153–160. [\[CrossRef\]](#)
49. Zhang, S.; Cheng, D.; Deng, Z.; Zong, M.; Deng, X. A novel k NN algorithm with data-driven k parameter computation. *Pattern Recognit. Lett.* **2018**, *109*, 44–54. [\[CrossRef\]](#)
50. Hassanat, A.B.; Abbadi, M.A.; Altarawneh, G.A.; Alhasanat, A.A. Optimal K parameter for KNN classifier with square root. *Int. J. Comput. Sci. Inf. Secur.* **2014**, *12*, 33–39.
51. Juba, B.; Le, H.S. Precision-Recall versus accuracy and the role of large data sets. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 4039–4048. [\[CrossRef\]](#)
52. Setiyadi, D.; Alizah, M.D.; Dharsono, Y.P.; Sautomo, S. Sfenrianto Accuracy, recall, precision of SVM kernels in predicting autistic spectrum disorder in adults. *Int. J. Recent Technol. Eng.* **2020**, *8*, 2215–2218. [\[CrossRef\]](#)
53. Lee, D.U.; Luk, W.; Villasenor, J.D.; Cheung, P.Y.K. A Gaussian noise generator for hardware-based simulations. *IEEE Trans. Comput.* **2004**, *53*, 1523–1534. [\[CrossRef\]](#)
54. Kumar, A.; Sodhi, S.S. Comparative analysis of Gaussian filter, median filter and denoise autoencoder. In Proceedings of the 7th International Conference on Computing for Sustainable Global Development. 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 12–14 March 2020; pp. 45–51. [\[CrossRef\]](#)
55. Noor, A.; Zhao, Y.; Khan, R.; Wu, L.; Abdalla, F.Y.O. Median filters combined with denoising convolutional neural network for Gaussian and impulse noises. *Multimed. Tools Appl.* **2020**, *79*, 18553–18568. [\[CrossRef\]](#)
56. Otair, M.; Abualigah, R.; Absaleem, A.; Tarawne, O.H. Exclusive median filter (EMF) for enhancing image quality. *AIP Conf. Proc.* **2023**, *2979*, 020005. [\[CrossRef\]](#)
57. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation Forest. In Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM), Pisa, Italy, 15–19 December 2008; IEEE: New York, NY, USA, 2008; pp. 413–422. [\[CrossRef\]](#)
58. Li, W.; Mo, W.; Zhang, X.; Squiers, J.J.; Lu, Y.; Sellke, E.W.; Fan, W.; DiMaio, J.M.; Thatcher, J.E. Outlier detection and removal improves accuracy of machine learning approach to multispectral burn diagnostic imaging. *J. Biomed. Opt.* **2015**, *20*, 121305. [\[CrossRef\]](#) [\[PubMed\]](#)
59. Plant, C.; Böhm, C.; Tilg, B.; Baumgartner, C. Enhancing instance-based classification with local density: A new algorithm for classifying unbalanced biomedical data. *Bioinformatics* **2006**, *22*, 981–988. [\[CrossRef\]](#)
60. Czarnowski, I. Cluster-based instance selection for machine classification. *Knowl. Inf. Syst.* **2012**, *30*, 113–133. [\[CrossRef\]](#)

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