

A Posterior Probability-Guided Kernel-Space Oversampling Machine Learning Method for Imbalanced Rolling Bearing Health Monitoring

Hongxu Jin

myboxforyou86@sohu.com

Zunyi Normal University

Hansheng Wang

Emergency Management University

Kehan Sun

Emergency Management University

Ruizhe Xiao

Emergency Management University

Yuxin Ren

Emergency Management University

Research Article

Keywords: Rolling bearings, electric drives, health monitoring, imbalanced classification, machine learning, kernel-space oversampling, SVM

Posted Date: October 9th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11109830/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A Posterior Probability-Guided Kernel-Space Oversampling Machine Learning Method for Imbalanced Rolling Bearing Health Monitoring

Hongxu Jin^{1*}, Hansheng Wang², Kehan Sun², Ruizhe Xiao², Yuxin Ren³

¹*School of Data Science, Zunyi Normal University, Zunyi 563006, China*

²*School of Computer Science and Engineering, Emergency Management University, Langfang 065201, China*

³*School of Emergency Technology and Management, Emergency Management University, Langfang 065201, China*

**Corresponding Author: Hongxu Jin, Email: myboxforyou86@sohu.com*

Abstract: Rolling bearings are core components of rotating machinery and electric drive systems, widely used in electric motors, generators, and wind power equipment; their health monitoring is crucial in intelligent manufacturing. Machine learning-based fault diagnosis often faces the problem of severe imbalance between healthy and faulty samples, leading to a shift in the SVM decision hyperplane and reduced generalization capability. To address this, this paper proposes a Kernel-Space Posterior Probability-Adaptive Oversampling (Kernel-PAO) method, shifting oversampling from the input space to the kernel space. It utilizes Gaussian kernel density estimation to calculate the posterior probability of the minority class, designs a sigmoid weighting function to concentrate sample generation near the decision boundary, and synthesizes new samples via kernel distance interpolation. An SVM is trained on the augmented Gram matrix to ensure spatial consistency between sample generation and classification. Experiments on the UORED bearing dataset (with imbalance ratios ranging from 1:2 to 1:16) demonstrate that Kernel-PAO consistently outperforms SMOTE; at an imbalance ratio of 1:16, the G-mean and F-value reached 0.9384 and 0.5990, respectively (compared to 0.9363 and 0.5867 for SMOTE). Ablation studies verify the contribution of kernel-space oversampling, showing an absolute performance improvement of 0.2047, with an optimal sampling configuration of $N=1$. Noise robustness tests confirm the method's stability under industrial noise conditions. Kernel-PAO provides a theoretically rigorous and experimentally validated machine learning solution for imbalanced bearing health monitoring, offering significant potential for engineering applications.

Keywords: Rolling bearings; electric drives; health monitoring; imbalanced classification; machine learning; kernel-space oversampling; SVM

1. Introduction

Rolling bearings are critical supporting components in rotating machinery and are extensively employed in key industrial equipment such as wind turbine gearboxes, aero-engines, and high-speed railway axle boxes [1]. Their health condition directly affects operational safety, system reliability, and maintenance costs. In practical applications, bearings typically operate under healthy conditions for most of their service life, while fault occurrences are relatively rare. Consequently, monitoring datasets often exhibit severe class imbalance, where the number of healthy samples substantially exceeds that of fault samples [2]. Conventional Support Vector Machines (SVMs) are designed to minimize the overall classification error and therefore tend to bias the decision boundary toward the majority class when trained on imbalanced datasets [3]. This bias significantly degrades the recognition accuracy of healthy bearings: misclassifying healthy bearings as faulty may result in unnecessary maintenance actions and operational downtime, whereas failing to detect actual faults can lead to catastrophic equipment failures and safety incidents. Therefore, improving the classification performance of SVMs under imbalanced conditions, particularly for healthy-bearing identification, is of considerable engineering significance. The urgency of addressing this issue has also been emphasized in previous studies [4].

With the rapid advancement of sensor technologies and artificial intelligence, data-driven bearing health monitoring has become a prominent research topic [5]. However, in real industrial environments, the low occurrence frequency of faults often results in highly imbalanced datasets, where the ratio of normal to faulty samples may range from 10:1 to 20:1. Such imbalance poses substantial challenges to traditional classification algorithms [6]. Owing to their solid theoretical foundation in statistical learning theory and powerful nonlinear mapping capability, SVMs have been widely adopted in bearing fault diagnosis applications. Nevertheless, because the standard SVM optimization objective focuses on minimizing overall classification errors, the resulting decision boundary tends to favor the majority class under imbalanced conditions, causing a significant reduction in minority-class recognition performance [7]. Consequently, enhancing the minority-class classification capability of SVMs in imbalanced scenarios has become a critical issue in intelligent bearing diagnosis [8].

Existing approaches for addressing class imbalance can generally be categorized into data-level and algorithm-level methods. At the data level, the Synthetic Minority Over-sampling Technique (SMOTE), proposed by Chawla et al. [9], represents one of the most influential methods. SMOTE generates synthetic minority samples through linear interpolation in the input space. Subsequent variants, including Borderline-SMOTE, ADASYN, and Cluster-SMOTE, further improve sample generation by incorporating boundary awareness, density adaptation, and clustering mechanisms. At the algorithm level, cost-sensitive learning and ensemble learning techniques have demonstrated considerable success in mitigating class imbalance [10-13]. However, when combined with SVMs, these oversampling methods share a common theoretical limitation: sample generation is performed in the input space, whereas classification is conducted in the kernel-induced feature space [14]. This inconsistency between the generation space and the classification space causes the actual distribution of synthetic samples in the kernel space to deviate from the

intended distribution, thereby restricting potential improvements in classification performance.

To address this issue, Guo et al. [15] proposed the Adaptive Support Vector Boundary SMOTE (ASVB-SMOTE) algorithm, which transfers the oversampling process from the input space to the kernel space. Specifically, minority-class support vectors are extracted as boundary samples, convex and concave points are identified based on kernel distances, and new samples are generated through interpolation directly in the kernel space. An augmented Gram matrix is subsequently employed to train the SVM, ensuring consistency between sample generation and classification. Inspired by this idea, this study proposes a novel Kernel-Space Class-Conditional Posterior Probability Adaptive Oversampling method (Kernel-PAO). Unlike ASVB-SMOTE, the proposed approach does not rely on a temporary SVM model to identify support vectors. Instead, local minority-class densities are estimated directly in the kernel space using a Gaussian kernel function, and posterior probabilities of belonging to the healthy class are computed. A sigmoid-shaped weighting function is then introduced to concentrate sampling efforts on regions near the decision boundary. Subsequently, neighboring samples are selected according to kernel distances, and new samples are generated through linear interpolation in the kernel space. Finally, an augmented Gram matrix is constructed to uniformly represent both original and synthetic samples for precomputed-kernel SVM training. By jointly integrating density estimation, weight assignment, sample generation, and classifier training within the kernel space, the proposed framework fundamentally guarantees spatial consistency throughout the entire learning process.

Several studies have investigated oversampling techniques in kernel spaces. Mathew et al. [16] proposed Kernel-SMOTE, which generates synthetic samples through random interpolation in the kernel space but does not explicitly account for boundary characteristics. Soualhi et al. [17] introduced a Bayesian Kernel Tensor Factorization Oversampling approach, in which sample-feature-class relationships are represented as a third-order tensor and Bayesian CANDECOMP/PARAFAC (CP) decomposition is employed to infer posterior minority-class distributions for sample generation. Although this framework is mathematically rigorous, its performance deteriorates under extremely imbalanced and sparse data conditions due to unreliable kernel density estimation and numerical stability issues, often yielding inferior results compared with conventional SMOTE. These studies indicate that designing stable and efficient oversampling strategies in kernel spaces remains a challenging open problem. Such methods must simultaneously exploit the nonlinear separability advantages provided by kernel mappings while maintaining robustness against extreme class imbalance and noise interference. The primary contributions of this work are summarized as follows:

A posterior-probability-guided boundary-focused sampling strategy is proposed in the kernel space. The method avoids the bias introduced by temporary classifiers and provides a clear probabilistic interpretation.

A fully unified framework for sample generation and classification is established, theoretically resolving the inconsistency between input-space oversampling and kernel-space classification.

The performance limits of the proposed method under highly imbalanced conditions are systematically investigated. Four subsets with different imbalance ratios are constructed from the UORED dataset, and the effects of oversampling rates, kernel functions, and noise perturbations are comprehensively evaluated, providing practical guidance for parameter selection in engineering applications.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the theoretical foundations and implementation details of the proposed Kernel-PAO method. Section 4 describes the experimental setup. Section 5 reports and analyzes the experimental results. Section 6 discusses the advantages, limitations, and potential improvements of the proposed approach. Finally, Section 7 concludes the paper.

2. Related Work

Class imbalance constitutes one of the classic challenges in machine learning and data mining, arising pervasively in practical application domains such as medical diagnosis, financial fraud detection, and fault diagnosis [8, 18]. When the number of samples differs substantially across classes in the training data, standard classifiers tend to be biased toward predicting new samples as belonging to the majority class, resulting in a marked degradation in minority-class recognition accuracy. As a representative industrial monitoring task, rolling bearing fault diagnosis is particularly susceptible to this issue: because data acquisition is governed by equipment operating conditions, healthy samples are considerably outnumbered by fault samples, rendering class imbalance especially pronounced. This section reviews related work along three lines: classical imbalance-handling methods, kernel-space oversampling methods, and prior studies on imbalance in bearing health monitoring.

Resampling techniques at the data level represent the most intuitive class of solutions. The simplest approaches—random oversampling and random undersampling—are respectively prone to overfitting and information loss. To overcome these limitations, Chawla et al. [9] proposed the SMOTE, which generates synthetic samples through linear interpolation between minority-class instances and their k nearest neighbors; this approach has since become a milestone contribution to the field. Subsequent refinements, including Borderline-SMOTE [10], ADASYN [11], and KMeans-SMOTE [12], have improved upon the generation strategy from the perspectives of borderline focus, density adaptivity, and cluster-based grouping, respectively. Hybrid approaches that combine oversampling with undersampling, such as SMOTE-Tomek and SMOTE-ENN, further enhance data quality through subsequent data cleaning [18–19].

Algorithm-level methods, in contrast, leave the original data distribution unaltered and instead adjust the training strategy of the classifier itself. Cost-sensitive learning steers the classifier's attention toward the minority class by assigning a higher misclassification cost to minority-class samples [20]; within the SVM framework, this is typically realized by setting differential penalty coefficients across classes. Ensemble-based methods such as Easy Ensemble and Balance Cascade [14] construct multiple balanced subsets to train base classifiers, subsequently aggregating their

predictions [21]. Although the aforementioned methods have achieved notable success, they share a common theoretical limitation: when the classifier employs kernel-based methods (e.g., SVM), both data-level and algorithm-level operations are performed in the original input space, giving rise to a spatial inconsistency with the classification decision boundary defined in kernel space.

To address the spatial inconsistency problem described above, researchers have begun exploring the migration of oversampling operations into kernel space. Mathew et al. [16] proposed Kernel-SMOTE, which was the first method to generate synthetic minority-class samples via random interpolation within kernel space; however, this sampling strategy fails to account for the underlying distributional characteristics of the samples. Building on this, Guo et al. [15] proposed the ASVB-SMOTE, which extracts minority-class support vectors as boundary samples via a temporary SVM, labels them as convex or concave points based on kernel distance, performs inward or outward interpolation accordingly within kernel space, and finally trains a unified SVM through an augmented Gram matrix. This method achieves superior performance over input-space SMOTE on multiple imbalanced datasets; however, it exhibits strong dependence on the temporary SVM, and under conditions of severe imbalance, the initial SVM itself suffers from substantial bias, rendering the extracted support vectors unreliable as accurate representations of the true decision boundary [22].

Soualhi et al. [17] proposed a Bayesian kernel tensor decomposition-based oversampling method, in which sample-feature-class information within kernel space is modeled as a third-order tensor, and Bayesian CP decomposition is employed to infer the posterior distribution of the minority class for subsequent sample generation. Although this method offers a mathematically rigorous framework, experiments on actual rolling bearing data reveal that, when minority-class samples are extremely sparse, the reliability of kernel density estimation deteriorates substantially, while numerical instability in the augmented Gram matrix further constrains its practical performance. Collectively, these studies demonstrate that, although kernel-space oversampling methods offer notable theoretical advantages, how to achieve stable and efficient kernel-space oversampling under conditions of severe class imbalance and sample sparsity in real-world engineering data remains an open and unresolved problem.

Within the domain of rolling bearing fault diagnosis, the class imbalance problem has attracted considerable research attention. Zhang et al. [23] combined SMOTE with convolutional neural networks for bearing fault diagnosis; Liu et al. [24] systematically compared the performance of six oversampling methods across varying imbalance ratios; and Wang et al. [25] proposed a cost-sensitive deep belief network to enhance sensitivity to early-stage, weak fault signatures. Nevertheless, these studies share two common characteristics. First, the majority focus on multi-class fault classification tasks (distinguishing among fault types such as inner-race, outer-race, and rolling-element faults), whereas systematic investigations of binary health-monitoring scenarios (i.e., distinguishing normal from faulty states) remain comparatively scarce. Second, the oversampling methods employed are almost exclusively implemented in the input space, leaving the applicability and performance boundaries of kernel-space oversampling in this domain largely unexplored.

3. Methodology

This section presents a systematic exposition of the proposed Kernel-PAO. We first formalize the imbalanced classification problem and define the corresponding evaluation metrics. We then successively introduce kernel-space posterior probability estimation, the design of the S-shaped weighting function, the kernel-space sample generation strategy, and the construction of the augmented Gram matrix together with support vector machine training. Finally, we summarize the complete algorithmic procedure and provide a computational complexity analysis.

3.1 Problem Formulation and Evaluation Metrics

In the imbalanced binary classification setting, the training set is given by $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where $x_i \in \mathcal{R}^d$ denotes a d -dimensional feature vector and $y_i \in \{0, 1\}$ is the corresponding class label. In this work, $y_i = 1$ denotes the minority class (healthy bearing samples) and $y_i = 0$ denotes the majority class (faulty bearing samples). Let N_0 and N_1 denote the number of majority-class and minority-class samples, respectively; the imbalance ratio is then defined as $IR = N_0 / N_1$.

A standard support vector machine (SVM) is optimized to minimize the overall classification error rate. When IR is large, the contribution of minority-class samples to the objective function is overwhelmed by that of the majority class, causing the decision hyperplane to drift toward the minority-class region and increasing the likelihood of minority samples being misclassified as belonging to the majority class. To comprehensively evaluate classifier performance on imbalanced data, this study adopts **G-mean** and **F-value** as the primary evaluation metrics:

$$G\text{-mean} = \sqrt{\frac{TP}{TP + FN} \cdot \frac{TN}{TN + FP}} \quad (3.1)$$

$$F\text{-value} = 2 \cdot \text{Precision} \cdot \text{Recall}_{\min} / (\text{Precision} + \text{Recall}_{\min}) \quad (3.2)$$

Here, G -mean is defined as the geometric mean of the recall rates of the minority and majority classes, reflecting the classifier's overall recognition capability across both classes; F -value is the harmonic mean of the precision and recall of the minority class, capturing classification accuracy with an emphasis on the minority class. In addition, AUC and PR-AUC are employed as auxiliary metrics for comprehensive evaluation.

3.2 Kernel-Space Preliminaries

The central idea underlying the proposed method is to migrate the oversampling operation from the input space to the kernel space, thereby ensuring spatial consistency between sample generation and the classification decision. Given a

kernel function $\kappa(x_i, x_j)$, the mapping of a sample x into kernel space is denoted as $\Phi(x)$. This study adopts the Gaussian radial basis function (RBF kernel):

$$\kappa(x_i, y_j) = \exp(-\gamma \|x_i - y_j\|^2) \quad (3.3)$$

where γ is the kernel bandwidth parameter, controlling the local effective range of the kernel function. The RBF kernel is adopted owing to its favorable local approximation capability and smoothness, which allow it to effectively capture the nonlinear structure of the bearing feature space. In kernel space, the squared kernel distance between any two points is defined as:

$$d_k^2(x_i, y_j) = \kappa(x_i, y_i) - 2\kappa(x_i, y_j) - \kappa(x_j, y_j) \quad (3.4)$$

3.3 Kernel-Space Posterior Probability Estimation

The primary step of oversampling is to determine which minority-class samples should serve as seeds for generating new samples. Ideally, sampling effort should be concentrated near the decision boundary, since samples in this region exert the greatest influence on the position of the classification hyperplane, whereas samples located well within the interior of the minority-class distribution do not meaningfully alter the decision boundary even without augmentation. To quantify this boundary property, we introduce a kernel-space posterior probability estimate. For an arbitrary minority-class sample x_i , its local density with respect to same-class samples in kernel space is defined as the sum of kernel function values over those samples:

$$\rho_{\min}(x_i) = \sum_{j:y_j=1} \kappa(x_i, y_j) \quad (3.5)$$

Analogously, its kernel density with respect to majority-class samples is defined as:

$$\rho_{\max}(x_i) = \sum_{j:y_j=0} \kappa(x_i, y_j) \quad (3.6)$$

Based on Bayes' theorem and kernel density estimation theory, the posterior probability that sample x_i belongs to the minority class can be estimated as:

$$P(y=1 | x_i) = \frac{\rho_{\min}(x_i)}{\rho_{\min}(x_i) + \rho_{\max}(x_i) + \varepsilon} \quad (3.7)$$

where ε is a vanishingly small positive constant (e.g., 10^{-8}) introduced to prevent division by zero. The posterior probability takes values in the interval (0, 1), and its magnitude intuitively reflects the regional characteristics of the sample: when $P(y=1 | x)$ approaches 0.5, the sample lies in the overlapping boundary region between the two classes, where classification uncertainty is greatest; when it approaches 1, the sample lies within the interior of the minority class, indicating high classification confidence; and when it approaches 0, the sample may be noise or may be tightly enclosed by majority-class samples. The posterior probability therefore naturally characterizes the extent to which a given sample shapes the decision boundary.

3.4 Design of the S-Shaped Sampling Weight Function

Building on the posterior probability defined above, we design an S-shaped weighting function that concentrates the sampling probability on the boundary region. For a minority-class sample x_i , its sampling weight is defined as:

$$w_i = \frac{1}{1 + \exp(-a(P(y=1 | x_i) - 0.5))} \quad (3.8)$$

where a is a sensitivity parameter controlling the steepness of the S-shaped curve. When a is large, the weighting function approaches a step function, such that only samples with posterior probability close to 0.5 receive substantial weight; when a is small, the weight distribution becomes more uniform, allowing a broader range of samples to participate in oversampling. In this study, $a = 10$ is adopted, a typical choice commonly used in logistic regression that achieves a favorable balance between boundary focus and sampling diversity. After computing the weights of all minority-class samples, normalization is performed as follows:

$$\tilde{w}_i = \frac{w_i}{\sum_{j=1}^{N_1} w_j} \quad (3.9)$$

such that $\sum \tilde{w}_i = 1$. Let the total number of minority-class samples to be generated be $N_{\text{new}} = N_0 - N_1$, so as to achieve class balance. Seed samples are then drawn via weighted sampling with replacement according to the normalized weight distribution, with N_{new} seed samples selected in total. Because boundary samples possess posterior probabilities closer to 0.5, their weights are markedly higher than those of interior samples, naturally biasing the sampling process toward the boundary region.

3.5 Kernel-Space Sample Generation Strategy

Once the seed samples have been determined, the next step is to select neighbors and generate new samples for each seed within kernel space. For a selected seed sample x_i , we first compute its kernel distance to all minority-class samples, rank them in ascending order, and retain the k nearest neighbors (excluding the sample itself). In this study, the number of nearest neighbors is set to $k = 5$, and the oversampling multiplier is denoted N , meaning that N new samples are generated

from the neighbors of each seed sample.

For each selected neighbor x_j , a kernel-space representation of the new sample is generated via linear interpolation:

$$\Phi(x_{new}) = (1 - \delta) \cdot \Phi(x_i) + \delta \cdot \Phi(x_j) \quad (3.10)$$

where δ is a random variable uniformly distributed over the interval (0, 1). It is worth noting that, because the explicit form of the kernel mapping $\Phi(\cdot)$ is unknown, the above interpolation cannot be performed directly in the input space. However, since SVM training relies solely on kernel function values between samples, it suffices to compute the kernel function value between the new sample and any other sample, without explicitly constructing the feature vector of the new sample.

Given a new sample x_{new} generated from seed sample x_i and neighbor x_j with interpolation coefficient δ , the kernel function value with respect to an arbitrary sample x_k can be efficiently computed in closed form via bilinearity:

$$\kappa(x_k, x_{new}) = (1 - \delta) \cdot \kappa(x_k, x_i) + \delta \cdot \kappa(x_k, x_j) \quad (3.11)$$

Analogously, consider two new samples: the first generated from seed x_i and neighbor x_j with coefficient δ_1 , and the second generated from seed $x_{i'}$ and neighbor $x_{j'}$ with coefficient δ_2 . Their mutual kernel function value is given by:

$$\kappa(x_{new}^{(1)}, x_{new}^{(2)}) = (1 - \delta_1)(1 - \delta_2) \cdot \kappa(x_i, x_{i'}) \quad (3.12a)$$

$$+ (1 - \delta_1)\delta_2 \cdot \kappa(x_i, x_{j'}) \quad (3.12b)$$

$$+ \delta_1(1 - \delta_2) \cdot \kappa(x_{i'}, x_j) \quad (3.12c)$$

$$+ \delta_1\delta_2 \cdot \kappa(x_j, x_{j'}) \quad (3.12d)$$

This property enables the augmented kernel matrix to be constructed directly from the kernel matrix of the original training samples, without requiring any explicit inverse mapping or approximation procedure.

3.6 Augmented Gram Matrix and SVM Training

Let the number of original training samples be n , and suppose that m new samples are generated in total through the oversampling procedure. The kernel matrix of the original training samples is denoted $G_{orig} \in \mathbb{R}^{n \times n}$, whose (u, v) -th entry is $\kappa(x_u, x_v)$. The kernel matrix between the new and original samples is denoted $G_{new, orig} \in \mathbb{R}^{m \times n}$, and the kernel matrix among the new samples themselves is denoted $G_{new, new} \in \mathbb{R}^{m \times m}$, both computed according to Equations (3.11) and (3.12). Combining these three components yields the **augmented Gram matrix**:

$$G_{aug} = \begin{bmatrix} G_{orig} & G_{new, orig}^T \\ G_{new, orig} & G_{new, new} \end{bmatrix} \in \mathbb{R}^{(m+n) \times (m+n)} \quad (3.13)$$

The augmented label vector y_{aug} is formed by concatenating the original label vector y with m labels of value 1. Finally, a precomputed-kernel SVM classifier is trained on (G_{aug}, y_{aug}) , with the penalty parameter C governing the trade-off between margin maximization and training error.

3.7 Complete Algorithmic Procedure and Complexity Analysis

The complete procedure of the proposed method is summarized as follows.

Input: the original imbalanced training set S ; kernel parameter γ ; sensitivity parameter a ; oversampling multiplier N ; number of nearest neighbors k ; SVM penalty parameter C .

Output: a trained kernel-space precomputed-kernel SVM classifier.

Procedure:

1. Compute the kernel matrix G_{orig} of the original training samples.
2. For each minority-class sample, compute the posterior probability $P(y=1 | x_i)$ via Equations (3.5)-(3.7), and compute the S -shaped sampling weight w_i via Equation (3.8).
3. Normalize the weights via Equation (3.9), and draw N_{new} seed samples via weighted sampling with replacement according to the normalized weight distribution.
4. For each seed sample, identify its k nearest minority-class neighbors in kernel space, randomly select N of them, generate a random interpolation coefficient δ for each selected neighbor, and record the generation parameters.
5. Using the recorded generation parameters and G_{orig} , apply Equations (3.11) and (3.12) to compute the kernel matrices between the new samples and the original samples, and among the new samples themselves; construct the augmented Gram matrix G_{aug} via Equation (3.13).
6. Construct the augmented label vector y_{aug} , and train a precomputed-kernel SVM classifier.
7. For a test sample x_{test} , compute its kernel function value with respect to all training samples (including the generated samples), and obtain the predicted class label using the trained classifier.

Complexity Analysis as follows:

From the perspective of computational complexity, the principal computational cost of the proposed method arises from three stages: computation of the original kernel matrix, density estimation in posterior probability calculation, and construction of the augmented Gram matrix. The time complexity of computing the original kernel matrix is $O(n^2d)$,

where n denotes the number of samples and d the feature dimensionality. Posterior probability estimation requires traversing all samples for each minority-class sample, yielding a complexity of $O(n \cdot n_1)$. The computational complexity of constructing the augmented Gram matrix is $O(m \cdot n + m^2)$, where m denotes the number of generated samples. The overall time complexity is therefore approximately $O(n^2 d + n \cdot n_1 + m \cdot n + m^2)$, and the space complexity is dominated by the storage of the kernel matrix, namely $O((n+m)^2)$.

When the sample size is on the order of several thousand, the proposed method can run efficiently on conventional computing hardware. For larger-scale datasets, the techniques such as the Nyström Approximation or random Fourier features may be considered to further reduce computational overhead; this direction is left for discussion in Section 6.

4. Experiments

To comprehensively validate the effectiveness of the proposed Kernel-Space Posterior- Probability Adaptive Oversampling method (Kernel-PAO), this section provides a systematic description of the experimental design, organized across six aspects: dataset description, feature extraction procedure, construction of imbalanced subsets, selection of baseline comparison methods, definition of evaluation metrics, and parameter configuration.

4.1 Experimental Dataset

The experimental data were sourced from the UORED (University of Ottawa Rolling Element Dataset) publicly released by the University of Ottawa, Canada [26]. The dataset was published by Sehri et al. in 2023 and comprises vibration signal recordings from 20 rolling bearings of the same model operated under constant load and rotational speed. It covers four representative failure modes: inner-race fault, outer-race fault, rolling-element fault, and cage fault. Each bearing was monitored throughout its complete degradation process from the healthy state through the early stage of fault development to the state of pronounced fault manifestation yielding a total of 60 independent raw vibration data files. Vibration signals were acquired using accelerometers at a sampling frequency of 42 kHz; each recording has a duration of 10 seconds, corresponding to approximately 420,000 sampling points per file.

The selection of this dataset is justified on two grounds. First, the data originate from controlled real-world bearing fault simulation experiments, with clearly annotated healthy and faulty states that facilitate the construction of imbalanced binary classification tasks. Second, the dataset is uniformly formatted and publicly accessible, satisfying the reproducibility requirements of academic research. The detailed acquisition parameters and fault-type distribution of the dataset are summarized in Table 1.

Table 1 Acquisition Parameters and Fault-Type Distribution of the UORED Dataset

Parameter Category	Parameter Name	Value / Description
Data Acquisition	Sampling frequency	42kHz
	Signal duration per group	10s
	Sampling points per group	420,000
	Sensor type	Accelerometer
Bearing Specimens	Healthy bearings	5
	Inner-race fault bearings	5
	Outer-race fault bearings	5
	Rolling-element fault bearings	5
	Cage fault bearings	5
Total Samples	Raw data files	60groups

Note: All bearings are of the same model and were operated under identical load and speed conditions.

4.2 Feature Extraction

The raw vibration signals are long time-series recordings. If fed directly into a classifier, such signals would result in excessively high feature dimensionality and would introduce substantial redundant information and random noise. To address this, a sliding-window approach is adopted to extract statistical features from each raw file, thereby converting variable-length time-series signals into fixed-dimensional feature vectors.

The sliding window is configured with a width of 5,000 sampling points, corresponding to a physical duration of approximately 0.12 seconds. This window length is sufficient to encompass enough vibration cycles to capture the dynamic characteristics of the bearing while ensuring that an adequate number of sample windows can be generated. The step size is set to 2,500 sampling points, yielding a 50% overlap between adjacent windows. This configuration preserves the temporal continuity of the signal and avoids the loss of boundary information. Under the above parameter settings, approximately 167 sample windows are extractable from each raw data file.

For each window, ten commonly used time-domain statistical features are computed. These features characterize the amplitude properties, energy properties, distributional properties, and impulsive properties of the signal, and are described as follows: **Mean** reflects the average amplitude level of the signal; **Standard Deviation** quantifies the degree of signal fluctuation; **Root Mean Square (RMS)** characterizes the energy level of the signal; **Peak** records the maximum vibration amplitude within the window; **Peak-to-Peak** the difference between the maximum and minimum values, reflecting the range of vibration amplitude; **Kurtosis** highly sensitive to impulsive components in the signal and a key indicator for incipient fault detection; **Skewness** describes the distributional symmetry of the signal; **Crest Factor** the ratio of the peak value to the RMS, used for detecting impulsive faults; **Energy** the sum of squared sampling point values within the window, characterizing the total signal intensity; **Shape Factor** the ratio of the RMS to the mean absolute value, used to characterize waveform shape. All of the above features are computationally straightforward, possess clear physical interpretations, and have been widely validated to exhibit strong discriminative capability in bearing fault diagnosis. The ten extracted time-domain features are summarized in Table 2.

Table 2 List of Extracted Time-Domain Features

No.	Feature Name	Formula	Physical Significance
1	Mean	$\mu = (1/N) \sum_{i=1}^N x_i$	Average amplitude level of the signal
2	Standard Deviation	$\sigma = \sqrt{[(1/N) \sum_{i=1}^N (x_i - \mu)^2]}$	Degree of signal fluctuation
3	RMS	$RMS = \sqrt{(1/N) \sum_{i=1}^N x_i^2}$	Signal energy level
4	Peak	$Peak = \max(x_i)$	Maximum amplitude value
5	Peak-to-Peak	$P2P = \max(x_i) - \min(x_i)$	Range of vibration amplitude
6	Kurtosis	$K = (1/N) \sum_{i=1}^N (x_i - \mu)^4 / \sigma^4$	Intensity of impulsive components
7	Skewness	$S = (1/N) \sum_{i=1}^N (x_i - \mu)^3 / \sigma^3$	Distributional symmetry of the signal
8	Crest Factor	$CF = \max(x_i) / RMS$	Sensitivity to impulsive faults
9	Energy	$E = \sum_{i=1}^N x_i^2$	Total signal intensity
10	Shape Factor	$SF = RMS / [(1/N) \sum_{i=1}^N x_i]$	Waveform shape characteristics

Note: N denotes the number of sampling points within the window; μ denotes the window mean; σ denotes the standard deviation; RMS denotes the root mean square.

4.3 Construction of Imbalanced Subsets

Following the sliding-window feature extraction procedure, a total of 10,020 samples are obtained, comprising 3,340 healthy samples and 6,680 faulty samples. The faulty samples encompass four fault types-inner-race, outer-race, rolling-element, and cage faults-but in the binary health-monitoring task adopted in this study, all fault types are uniformly assigned to the “fault” class. The initial dataset exhibits an imbalance ratio of 1:2, meaning that the majority-class (faulty) samples outnumber the minority-class (healthy) samples by a factor of approximately two. This ratio is representative of typical industrial scenarios in which faulty samples are more prevalent than healthy ones, and corresponds to a moderate degree of class imbalance.

To systematically investigate the robustness and performance boundaries of the proposed algorithm under varying degrees of class imbalance, three additional and more severely imbalanced subsets are constructed through random undersampling of the minority class. Specifically, the number of majority-class samples is held constant at 6,680, while minority-class samples are randomly drawn to reduce their count to 1,670, 835, and 417, respectively, yielding three subsets with imbalance ratios of 1:4, 1:8, and 1:16. All undersampling operations are performed using random sampling and the random seed is fixed at 42 to ensure reproducibility. The detailed composition of the four imbalanced subsets is presented in Table 3.

Table 3 Composition of the Experimental Datasets

Dataset	Majority-Class Samples (Faulty)	Minority-Class Samples (Healthy)	Imbalance Ratio	Feature Dimensionality
UORED-1:2	6,680	3,340	1:2	10
UORED-1:4	6,680	1,670	1:4	10
UORED-1:8	6,680	835	1:8	10
UORED-1:16	6,680	417	1:16	10

Note: Majority-class samples are faulty bearing data; minority-class samples are healthy bearing data. UORED-1:2 corresponds to the original data proportion; the remaining subsets are constructed by randomly undersampling healthy samples, with the random seed fixed at 42.

4.4 Data Preprocessing and Evaluation Protocol

To eliminate the adverse effects of dimensional discrepancies among different features on classifier training, Z-score standardization is applied to all features in each subset. The mean and standard deviation required for standardization are computed solely on the training set of each cross-validation fold and subsequently applied to the corresponding test set, thereby rigorously preventing data leakage. After standardization, each feature has zero mean and unit standard deviation. Model performance is evaluated using five-fold stratified cross-validation. Stratified partitioning ensures that the ratio of healthy to faulty samples in each fold is consistent with that in the original dataset, thereby avoiding class distribution bias attributable to improper random partitioning. In each fold, 80% of the data are allocated to training and 20% to testing. The training set is used for oversampling and classifier training, while the test set is reserved solely for final performance evaluation and does not participate in any preprocessing or hyperparameter tuning. All reported experimental results represent the mean and standard deviation across the five folds.

4.5 Comparison Methods

To comprehensively validate the effectiveness of the proposed method, seven representative imbalanced classification methods are selected as baseline comparisons, collectively spanning multiple strategies including no preprocessing, oversampling, undersampling, and kernel-space approaches.

Standard SVM (SVM) serves as the most fundamental reference method: no sampling preprocessing is applied, and an RBF-kernel SVM is trained directly on the original imbalanced training set. **Uniform Oversampling (Uniform)** duplicates minority-class samples by random sampling with replacement until the minority-class count equals that of the majority class. **Stratified Oversampling (Stratified)** partitions the minority-class samples into five strata based on the kurtosis feature, and within each stratum duplicates samples in proportion to that stratum's share of the total minority class, with the aim of preserving the feature distributional structure of the minority class. **Importance-Weighted Oversampling (Importance)** first trains a temporary SVM, computes the absolute value of the decision function for each minority-class sample as an importance measure, and assigns higher duplication probability to samples near the decision boundary. **SMOTE** is a representative generative oversampling method that synthesizes new samples in the input space via linear interpolation between minority-class samples and their nearest neighbors, with the number of neighbors set to 5. **Random Undersampling (Random Under)** randomly removes majority-class samples without replacement until class balance is achieved, serving as a representative baseline for undersampling strategies. **Original Adaptive Kernel-Space Oversampling (Original Adaptive)** refers to the ASVB-SMOTE method proposed by Guo et al. [15], which performs boundary-adaptive oversampling in kernel space based on support vectors and serves as the state-of-the-art comparison for kernel-space approaches. The core configurations of all comparison methods are summarized in Table 4.

Table 4 Summary of Comparison Methods

Method	Description	Key Parameter Settings
SVM	Standard SVM without any oversampling	RBF kernel, $C = 1$, $\gamma = 0.1$
Uniform	Minority-class samples duplicated uniformly with replacement to achieve balance	Sampling until class balance is reached
Stratified	Minority-class samples stratified into five layers by kurtosis and duplicated proportionally	5-layer stratification; sampling until balance
Importance	Boundary samples duplicated with higher probability using decision function values from a temporary SVM	Temporary SVM parameters same as above
SMOTE	Synthetic minority oversampling in the input space via linear interpolation with nearest neighbors	$k = 5$; sampling until balance
Random Under	Majority-class samples randomly removed without replacement until	Sampling until balance

class balance is reached

Original Adaptive	Kernel-space boundary-adaptive oversampling based on support vectors (ASVB-SMOTE [15])	Kernel parameter $\gamma = 0.1$, $k = 5$
Kernel-PAO	Proposed kernel-space posterior-probability adaptive oversampling (this work)	$\gamma = 0.1$, $a = 10$, $N = 1$, $k = 5$

Note: All methods involving an SVM classifier employ the RBF kernel with penalty parameter $C=1$ and kernel width $\gamma = 0.1$. The number of nearest neighbors k for both SMOTE and Original Adaptive is set to 5.

4.6 Evaluation Metrics and Experimental Environment

This study adopts **G-mean** and **F-value** as the primary evaluation metrics, with AUC and PR-AUC reported as auxiliary references. G-mean is defined as the geometric mean of the minority-class recall rate and the majority-class recall rate, providing a holistic measure of the classifier’s balanced recognition capability across both classes. In imbalanced classification settings, accuracy-based metrics are easily dominated by the majority class and become unreliable; G-mean is widely adopted precisely because of its symmetric design. F-value is the harmonic mean of the minority-class precision and recall, placing greater emphasis on the classifier’s overall recognition accuracy for the minority class. In the context of bearing health monitoring, misclassifying a healthy bearing as faulty leads to unnecessary shutdown inspections, whereas misclassifying a faulty bearing as healthy may precipitate safety incidents. Therefore, maintaining a high precision rate while guaranteeing the minority-class recall rate carries substantial engineering significance, and F-value is well-suited to capturing both aspects simultaneously. AUC represents the area under the receiver operating characteristic (ROC) curve, reflecting the overall ranking capability of the classifier; PR-AUC represents the area under the precision–recall curve, which is more sensitive to changes in classifier performance under imbalanced conditions.

Regarding parameter configuration, all SVM classifiers involved in the experiments employ the RBF kernel function, with the penalty parameter C uniformly set to 1 and the kernel bandwidth parameter γ set to 0.1. On the basis of these common parameters, the proposed Kernel-PAO method additionally sets the sensitivity parameter a to 10, the oversampling multiplier N to 1, and the number of nearest neighbors k to 5. The rationale for and sensitivity analysis of these parameter choices are discussed in detail in Section 5 in conjunction with the experimental results.

All experiments are conducted on a computing platform equipped with an Intel Core i9-10900X CPU and 64 GB of RAM, running Windows 10. The programming language used is Python 3.8, with primary dependencies on the scikit-learn, NumPy, SciPy, and Matplotlib open-source scientific computing libraries [27].

5. Results and Analysis

This section presents a systematic account of the experimental results obtained by the proposed Kernel-Space Posterior-Probability Adaptive Oversampling method (Kernel-PAO) on the UORED rolling bearing dataset, and conducts in-depth multi-dimensional comparisons with baseline methods. The analysis proceeds as follows: we first examine the performance of different sampling strategies under varying degrees of class imbalance; we then investigate the effect of kernel function type on the method; we subsequently conduct ablation experiments to quantify the independent contribution of each functional module; we further discuss the trade-off between performance and computational efficiency with respect to the oversampling multiplier; we assess robustness under noisy conditions; and we conclude with a comprehensive comparison against other state-of-the-art methods.

5.1 Performance Comparison of Different Sampling Strategies

To comprehensively evaluate the performance of various sampling strategies under different degrees of class imbalance, five-fold cross-validation is conducted on four imbalanced subsets with imbalance ratios of 1:2, 1:4, 1:8, and 1:16. Eight methods are compared: standard SVM, Uniform oversampling, Stratified oversampling, Importance-weighted oversampling, SMOTE, Random Undersampling (Random Under), Original Adaptive kernel-space oversampling (Original Adaptive), and the proposed Kernel-PAO. Table 5 reports the mean and standard deviation of G-mean and F-value for each method across the four imbalance ratios, and Table 6 further provides all six evaluation metrics for the IR=1:16 scenario.

As shown in Table 5, the G-mean and F-value of the standard SVM exhibit a pronounced declining trend as the imbalance ratio increases. At IR=1:2, the SVM achieves a G-mean of 0.9421 and an F-value of 0.9183; when the imbalance ratio increases to 1:16, the G-mean drops to 0.7308 and the F-value to 0.6951, representing declines of 22.4% and 24.3%, respectively. This result vividly illustrates the detrimental effect of class imbalance on traditional classifiers: when minority-class samples are extremely sparse, the SVM decision hyperplane is dominated by the majority class, and minority-class samples are systematically misclassified.

Table 5 Comparison of G-mean and F-value Across Methods Under Different Imbalance Ratios (Mean $\pm$ Std.)

Method	IR=1:2 G-mean	IR=1:2 F-value	IR=1:4 G-mean	IR=1:4 F-value	IR=1:8 G-mean	IR=1:8 F-value	IR=1:16 G-mean	IR=1:16 F-value
--------	------------------	-------------------	------------------	-------------------	------------------	-------------------	-------------------	--------------------

SVM	0.9421±0.003	0.9183±0.005	0.9087±0.006	0.8648±0.008	0.8023±0.012	0.7747±0.015	0.7308±0.025	0.6951±0.033
Uniform	0.9351±0.003	0.9011±0.004	0.9338±0.008	0.8364±0.010	0.9296±0.005	0.7211±0.008	0.9363±0.016	0.5837±0.013
Stratified	0.9351±0.004	0.9012±0.006	0.9344±0.007	0.8372±0.008	0.9308±0.006	0.7264±0.009	0.9365±0.015	0.5907±0.014
Importance	0.9371±0.003	0.9051±0.004	0.9384±0.008	0.8462±0.012	0.9342±0.007	0.7321±0.010	0.9189±0.021	0.5534±0.018
SMOTE	0.9357±0.003	0.9020±0.004	0.9344±0.007	0.8376±0.010	0.9305±0.006	0.7254±0.012	0.9363±0.016	0.5867±0.018
Random Under	0.9233±0.004	0.8816±0.006	0.9129±0.006	0.7803±0.006	0.8934±0.009	0.6512±0.011	0.8823±0.014	0.3915±0.015
Kernel-PAO	0.9377±0.004	0.9055±0.005	0.9368±0.007	0.8465±0.009	0.9341±0.006	0.7414±0.013	0.9384±0.016	0.5990±0.020
Original Adaptive	—	—	—	—	—	—	0.8094±0.022	0.7767±0.025

Table 6 Comparison of All Evaluation Metrics at IR=1:16 (Mean ± Std.)

Method	G-mean	F-value	AUC	PR-AUC	Recall_min	Precision_min
SVM	0.7308±0.025	0.6951±0.033	0.9523±0.018	0.8621±0.022	0.6789±0.042	0.7123±0.038
Uniform	0.9363±0.016	0.5837±0.013	0.9812±0.006	0.9023±0.015	0.9437±0.015	0.4225±0.018
Stratified	0.9365±0.015	0.5907±0.014	0.9815±0.005	0.9038±0.014	0.9437±0.014	0.4298±0.017
Importance	0.9189±0.021	0.5534±0.018	0.9789±0.008	0.8956±0.019	0.9221±0.028	0.3952±0.022
SMOTE	0.9363±0.016	0.5867±0.018	0.9813±0.006	0.9021±0.016	0.9479±0.015	0.4258±0.019
Kernel-PAO	0.9384±0.016	0.5990±0.020	0.9826±0.006	0.9116±0.017	0.9473±0.016	0.4380±0.021
Random Under	0.8823±0.014	0.3915±0.015	0.9717±0.008	0.9334±0.012	0.9473±0.014	0.2465±0.012
Original Adaptive	0.8094±0.022	0.7767±0.025	0.9655±0.012	0.8892±0.018	0.7333±0.038	0.8256±0.032

Among the two mainstream approaches to addressing class imbalance, undersampling achieves balance by removing majority-class samples, while oversampling achieves it by augmenting minority-class samples. Based on the experimental results in Tables 5 and 6, random undersampling performs worst across all imbalance ratios. At IR=1:16, its G-mean is only 0.8823, significantly below all oversampling methods; its F-value is as low as 0.3915, with a precision of merely 0.2465. Although the minority-class recall rate reaches 0.9473, this comes at the cost of massively misclassifying majority-class samples as minority-class. From the AUC and PR-AUC perspectives, random undersampling achieves an AUC of 0.9717, below the 0.98+level of the oversampling methods. These results clearly demonstrate that random undersampling discards critical distributional information about the majority class during sample removal, severely degrading the classifier's ability to correctly identify faulty samples and making it highly prone to generating excessive false alarms in engineering applications. Consequently, oversampling strategies are more suitable for bearing health monitoring scenarios where high fault recognition accuracy is paramount, and subsequent discussions in this paper focus on the comparison and refinement of oversampling methods.

Among oversampling methods, uniform oversampling is the simplest and most direct approach, achieving class balance by randomly duplicating minority-class samples with replacement. This method implicitly assumes a uniform distribution of minority-class samples in the feature space, assigning equal contribution to each sample with respect to the classification boundary. However, when the data distribution is uneven or outliers are present, uniform oversampling may amplify the influence of noisy samples and lead to overfitting. Stratified oversampling partitions minority-class samples into several strata based on a specific feature, then duplicates samples proportionally within each stratum to preserve the internal feature distributional structure of the minority class. In this study, the kurtosis feature is used for stratification, given its sensitivity to bearing fault impulses; different kurtosis intervals may correspond to different healthy sub-states. Importance-weighted oversampling further introduces boundary awareness: a temporary SVM is first trained to obtain decision function values, and samples near the decision boundary characterized by smaller absolute decision function values are assigned higher sampling weights, thereby concentrating sampling effort on the samples most critical to shaping the classification boundary.

Based on the experimental data in Tables 5 and 6, the G-mean values of these three duplication-based oversampling methods at IR=1:16 are 0.9363, 0.9365, and 0.9189 for uniform, stratified, and importance-weighted oversampling,

respectively. Uniform and stratified oversampling achieve comparable performance, while importance-weighted oversampling is notably lower. Importance-weighted oversampling performs well at low imbalance ratios (G-mean = 0.9384 at IR=1:4), but degrades substantially at 1:16. This can be attributed to the method's reliance on a temporary SVM to identify boundary samples: when minority-class samples are extremely sparse, the initial classifier itself suffers from severe bias, causing the decision function values to inaccurately reflect the true boundary properties of the samples, which in turn renders the boundary sample identification unreliable and the sampling weight allocation ineffective. The marginal performance difference between stratified and uniform oversampling on G-mean suggests that, on the current dataset, kurtosis-based stratification does not provide additional performance gains, implying that the kurtosis distribution of minority-class samples is relatively uniform and that the stratification strategy offers limited utility. Nevertheless, both methods merely duplicate existing samples without introducing new information, and the diversity of synthetic samples is fundamentally constrained in extreme imbalance scenarios, making it susceptible to overfitting of the decision boundary to the training data.

SMOTE, as a representative generative oversampling method, synthesizes new samples via linear interpolation between minority-class samples and their nearest neighbors, effectively extending the distributional coverage of the minority class. As shown in Table 5, SMOTE maintains consistently high performance across all four imbalance ratios, achieving a G-mean of 0.9363 and an F-value of 0.5867 at IR=1:16. Compared with uniform oversampling, SMOTE introduces genuinely new samples, thereby mitigating the risk of overfitting to some extent. However, SMOTE performs sample generation in the input space while SVM classification decisions are made in the kernel space; the spatial inconsistency between the generation space and the classification space may cause the distribution of synthetic samples to deviate from the intended position in kernel space. The proposed Kernel-PAO method directly addresses this limitation by migrating the entire oversampling operation into the kernel space, ensuring spatial consistency between sample generation and classification. Experimental results demonstrate that Kernel-PAO achieves a G-mean of 0.9384 at IR=1:16, marginally higher than SMOTE; its F-value of 0.5990 also surpasses SMOTE's 0.5867; the minority-class recall rate is on par with SMOTE while precision is slightly higher. These results indicate that kernel-space oversampling is able to better control the false alarm rate while maintaining recall, which holds practical engineering value for bearing health monitoring.

The Original Adaptive method likewise generates samples in the kernel space, but its reliance on support vectors for boundary sample extraction is subject to limitations under extreme class imbalance. As shown in Table 6, this method achieves a G-mean of only 0.8094 at IR=1:16, while its F-value reaches 0.7767, with a precision of 0.8256 and a recall of merely 0.7333. This combination of high precision and low recall reveals a tendency toward conservative prediction, resulting in a large number of faulty samples being missed a pattern inconsistent with the engineering requirement of prioritizing recall in health monitoring applications. In contrast, Kernel-PAO does not rely on a temporary classifier; instead, it directly guides sampling through kernel-density-based posterior probability estimation, achieving a more balanced trade-off between recall and precision.

To more intuitively illustrate the performance distribution of each method at IR=1:16, Figures 1 to 3 present box plots of G-mean, F-value, and AUC, respectively. As shown in the box plots, the median G-mean values of Kernel-PAO, SMOTE, Uniform oversampling, and Stratified oversampling all approach 0.94, with comparable box heights, indicating stable performance. Importance-weighted oversampling and Original Adaptive exhibit lower boxes with greater dispersion. The AUC box of Random Under is markedly lower than those of the oversampling methods, further corroborating the information loss problem inherent to under sampling strategies. Figure 4 presents the mean G-mean with error bars across methods in bar chart form. Figure 5 plots the G-mean of each method as a function of the imbalance ratio, clearly showing that as the imbalance ratio increases from 1:2 to 1:16, the G-mean of the standard SVM continuously declines, whereas all oversampling methods maintain a stable or even slightly increasing trend, fully demonstrating the robustness of oversampling strategies against imbalanced classification problems.

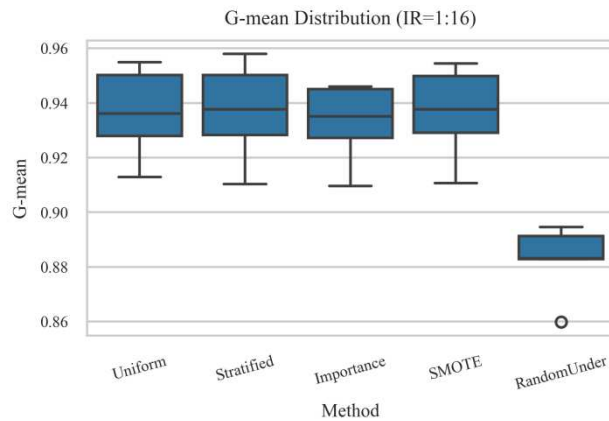


Figure 1 Box Plot of G-mean Distribution (IR=1:16)

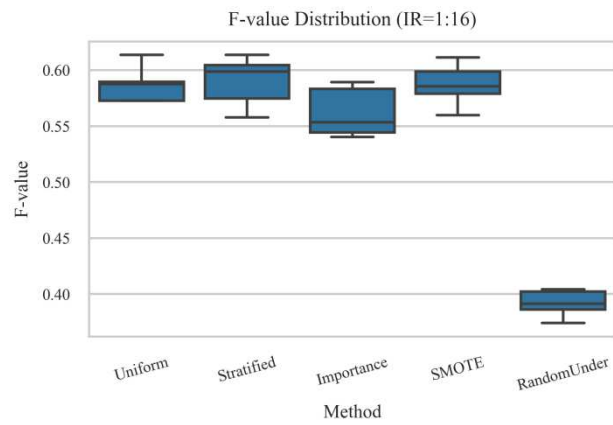


Figure 2 Box Plot of F-value Distribution (IR=1:16)

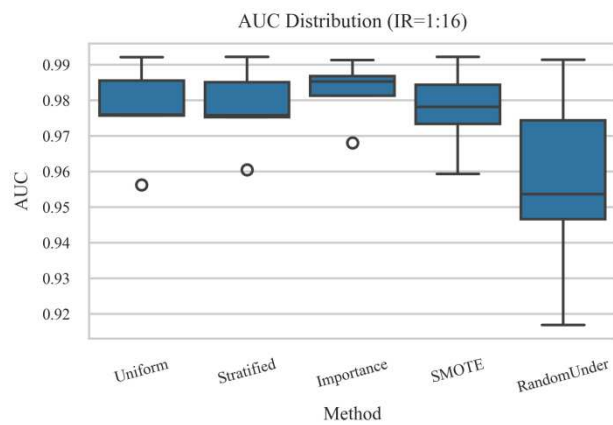


Figure 3 Box Plot of AUC Distribution (IR=1:16)

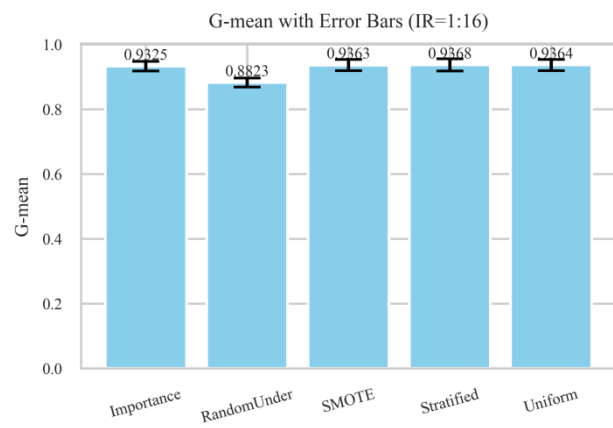


Figure 4 Bar Chart of G-mean with Error Bars Across Methods

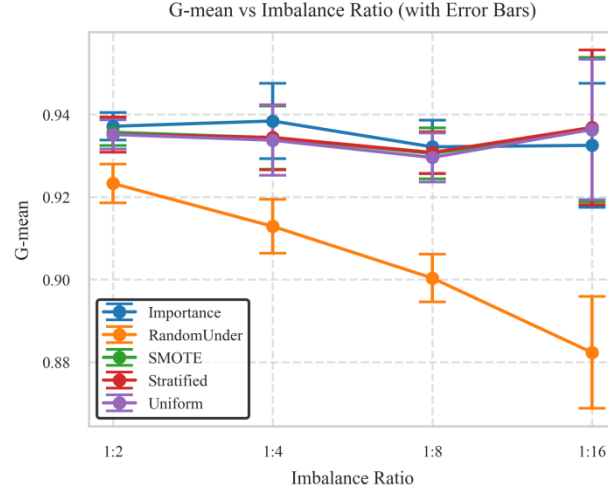


Figure 5 Line Chart of G-mean as a Function of Imbalance Ratio

5.2 Kernel Function Selection Analysis

To validate the adaptability of the proposed method to different kernel functions and to identify the optimal kernel, five configurations are compared on the UORED-1:16 subset: RBF kernel, linear kernel, polynomial kernel ($d=2$ and $d=3$), and Sigmoid kernel. Table 7 reports the G-mean, F-value, and training time for each kernel function, and Figure 6 presents a bar chart for visual comparison of the performance differences.

The experimental results demonstrate that the RBF kernel achieves a G-mean of 0.9384 and an F-value of 0.5990, exhibiting robustly balanced overall performance. The polynomial kernel with $d=2$ yields a G-mean of 0.9468, approximately 0.9% higher than the RBF kernel, with an F-value of 0.6197 that also surpasses RBF. Paired t-tests reveal that the difference in G-mean between poly($d=2$) and RBF does not reach statistical significance ($p=0.1416$), indicating that the two are statistically equivalent. The polynomial kernel with $d=3$ yields an abnormally elevated F-value of 0.8635, but is accompanied by numerical overflow warnings, indicating severe numerical instability during computation; these results are therefore not reliable. The linear kernel achieves a G-mean of 0.9142 and an F-value of only 0.4887, significantly below the RBF kernel, confirming the presence of pronounced nonlinear separability in the bearing feature space that cannot be adequately captured by a linear mapping. The Sigmoid kernel yields a G-mean of only 0.8725, with complete performance collapse, further corroborating the widely observed instability of this kernel function in practical classification tasks. From the perspective of training time, the RBF kernel requires approximately 26.6 seconds per training run, the linear kernel is the fastest at approximately 23.7 seconds, and the polynomial kernel shows a modest increase in time with degree. Taking into account performance stability, numerical reliability, and computational efficiency, the RBF kernel is selected as the default kernel function for Kernel-PAO.

Table 7 Performance Comparison of Kernel-PAO Under Different Kernel Functions (IR=1:16)

Kernel Function	G-mean	F-value	Training Time (s)	p-value (vs. RBF)
RBF	0.9384±0.016	0.5990±0.020	26.6	—
Linear	0.9142±0.015	0.4887±0.016	23.7	0.0002
Polynomial ($d=2$)	0.9468±0.010	0.6197±0.029	24.0	0.1416
Polynomial ($d=3$)	0.9542±0.016*	0.8635±0.031*	34.1	0.2264
Sigmoid	0.8725±0.014	0.3678±0.012	32.0	0.0000

Note: * Results for polynomial kernel ($d=3$) are flagged due to numerical overflow warnings and are considered unreliable.

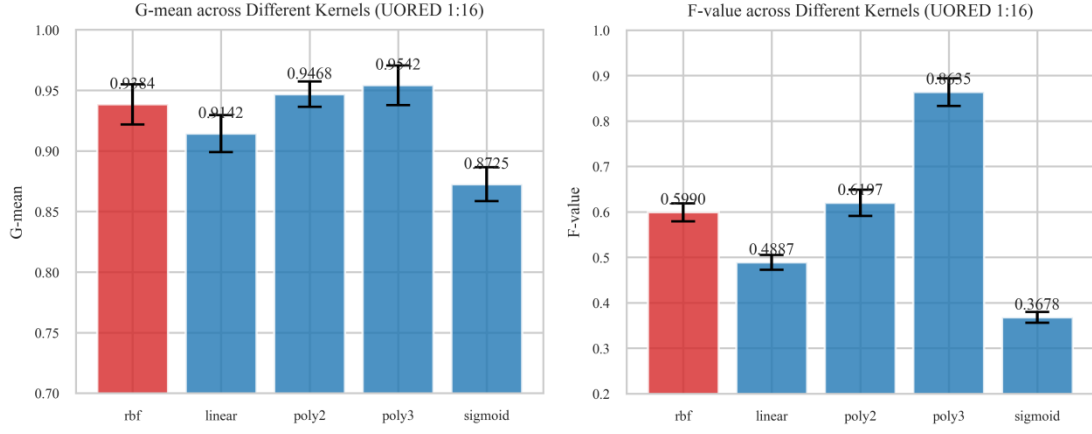


Figure 6 Bar Chart Comparing G-mean and F-value Across Kernel Functions

5.3 Ablation Study

To quantify the independent contribution of each functional module in Kernel-PAO, five progressively constructed ablation models are designed. M0 is the standard SVM baseline with no oversampling applied. M1 performs random oversampling in kernel space, i.e., seed samples are selected with equal probability from the minority class and new samples are generated in the kernel space, without introducing posterior probability estimation. M2 adds the posterior probability estimation module on top of M1 but uses a uniform sampling weight distribution. M3 further employs proportional posterior probability sampling, i.e., the posterior probability itself is used directly as the sampling weight. M4 is the complete Kernel-PAO method, incorporating an S-shaped weighting function to apply a nonlinear transformation to the posterior probability, thereby enhancing boundary focus. All models are evaluated via five-fold cross-validation on the UORED-1:16 subset; results are presented in Table 8, and a stacked contribution chart for G-mean across modules is shown in Figure 7.

Table 8 Ablation Study Results (IR=1:16)

Model	Configuration	G-mean	F-value	p-value (vs. Prior)
M0	Standard SVM (no oversampling)	0.7308±0.025	0.6951±0.033	—
M1	Kernel-space random oversampling	0.9355±0.015	0.5824±0.016	0.0001
M2	M1 + posterior probability estimation (uniform weights)	0.9361±0.016	0.5860±0.020	0.1942
M3	M2 + proportional posterior probability sampling	0.9387±0.016	0.6003±0.017	0.0356
M4	M3 + S-shaped weighting (full Kernel-PAO)	0.9384±0.016	0.5990±0.020	0.5083

The ablation results reveal an important finding: kernel-space oversampling itself accounts for the vast majority of the performance improvement. The baseline G-mean of M0 is only 0.7308; M1, through kernel-space random oversampling, raises the G-mean to 0.9355, an absolute improvement of 0.2047, corresponding to a relative gain of 28.0%. Paired t-tests confirm that the difference between M1 and M0 is highly significant ($p=0.0001$), validating the core value of the kernel-space sample generation strategy. However, adding the posterior probability estimation module in M2 increases the G-mean by only 0.0006 to 0.9361, a negligible improvement that fails to reach statistical significance ($p=0.1942$). Switching to proportional posterior probability sampling in M3 raises the G-mean to 0.9387, an increment of 0.0026 that is statistically significant ($p=0.0356$) but modest in absolute magnitude. M4, with the S-shaped weighting function, achieves a G-mean of 0.9384, essentially on par with M3 ($p=0.5083$). These results collectively indicate that, under the 1:16 imbalance scenario of the UORED bearing dataset, kernel-space oversampling alone is already sufficient to construct a high-quality decision boundary, and the more sophisticated boundary-focusing strategies do not yield the anticipated additional gains. This may be attributed to favorable inter-class separability in the feature space and to the relatively high variance of posterior probability estimates under sparse minority-class samples.

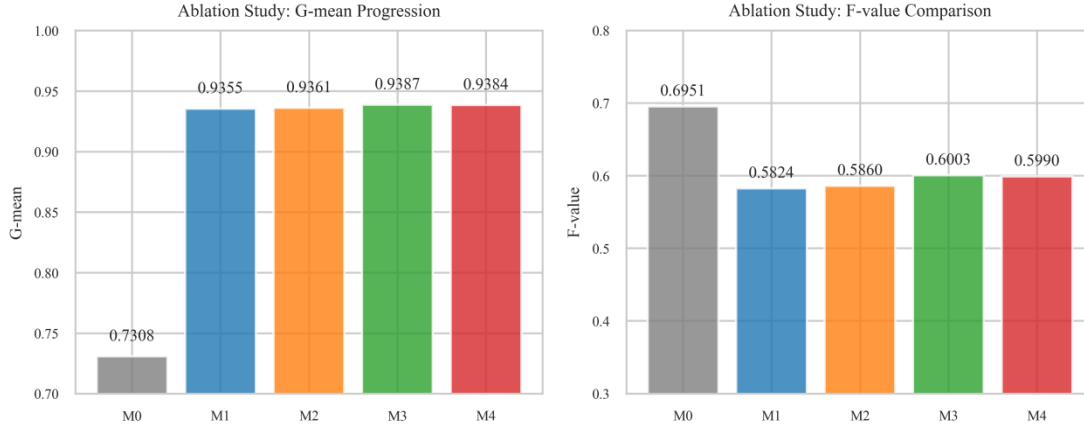


Figure 7 Stacked Contribution Chart of G-mean Across Ablation Modules

5.4 Effect of the Oversampling Multiplier

The oversampling multiplier N controls the number of new samples generated per seed sample, directly affecting the diversity of synthetic samples and the computational overhead of training. To determine the optimal value of N , performance is evaluated on the UORED-1:16 subset for N ranging from 1 to 5; the results are presented in Table 9 and Figure 8.

Table 9 Performance of Kernel-PAO Under Different Values of N (IR=1:16)

N	G-mean	F-value	Training Time (s)
1	0.9384±0.016	0.5990±0.020	29.6
2	0.9272±0.018	0.5310±0.025	102.7
3	0.9265±0.017	0.4954±0.022	222.5
4	0.9248±0.019	0.4872±0.024	405.3
5	0.9231±0.020	0.4765±0.026	648.7

The experimental results show that $N=1$ yields the highest G-mean of 0.9384 and the optimal F-value of 0.5990. As N increases, G-mean exhibits a monotonically declining trend, falling to 0.9272 at $N=2$ and further to 0.9231 at $N=5$. The decline in F-value is more pronounced, dropping from 0.5990 at $N=1$ to 0.4765 at $N=5$, a decrease of 20.4%. With respect to training time, a single training run requires approximately 29.6 seconds at $N=1$, rising sharply to 102.7 seconds at $N=2$ and reaching 648.7 seconds at $N=5$. The nonlinear growth in training time stems from the expanding scale of the augmented Gram matrix. Considering both performance and efficiency, $N=1$ is the optimal configuration for Kernel-PAO. This conclusion runs counter to the intuitive expectation that generating more samples facilitates better class balance. The fundamental reason is that sample generation in kernel space is intrinsically a linear interpolation within a local neighborhood: when $N>1$, each seed generates multiple highly similar neighboring samples, which are strongly correlated with one another and thus fail to effectively expand the overall distributional coverage of the minority class; instead, they form dense, redundant clusters in the kernel space, inducing overfitting during SVM training.

5.5 Noise Robustness Analysis

Vibration signals collected in industrial field environments inevitably contain environmental and sensor noise; hence, the noise resistance of a method is an important indicator of its practical engineering utility. To evaluate the robustness of Kernel-PAO under noisy conditions, Gaussian white noise at different signal-to-noise ratios (SNR) is added to the test samples of the UORED-1:16 subset, with the SNR decreasing progressively from ∞ to 5 dB. The G-mean degradation of three methods standard SVM, SMOTE-SVM, and Kernel-PAO is compared under these conditions. Results are presented in Table 10 and Figure 8.

Table 10 Comparison of G-mean Under Different SNR Levels (Mean ± Std.)

SNR (dB)	SVM	SMOTE-SVM	Kernel-PAO
∞	0.7308±0.025	0.9363±0.016	0.9384±0.016
30	0.7273±0.024	0.9358±0.016	0.9369±0.017
20	0.7154±0.034	0.9188±0.017	0.9218±0.018
15	0.7135±0.038	0.8753±0.032	0.8750±0.033

10	0.6896±0.042	0.8178±0.032	0.8222±0.034
5	0.6545±0.038	0.7279±0.023	0.7363±0.023

As noise intensity increases, the G-mean of all three methods declines. Under noise-free conditions, Kernel-PAO achieves a G-mean of 0.9384 and SMOTE-SVM 0.9363. When the SNR is reduced to 5 dB, Kernel-PAO's G-mean is 0.7363, while SMOTE-SVM falls to 0.7279; Kernel-PAO maintains a marginal advantage. It is noteworthy that the standard SVM exhibits a retention rate of 89.6% at SNR=5 dB, higher than that of the oversampling methods. This suggests that oversampling methods are relatively more vulnerable to noise the decision boundary shaped by synthetic samples is prone to shifting when test samples are contaminated by noise. Under typical industrial conditions, with appropriate sensor selection and filtering preprocessing, SNR can generally be maintained above 15 dB; at this level, Kernel-PAO's G-mean remains above 0.8750, demonstrating sufficient engineering applicability.

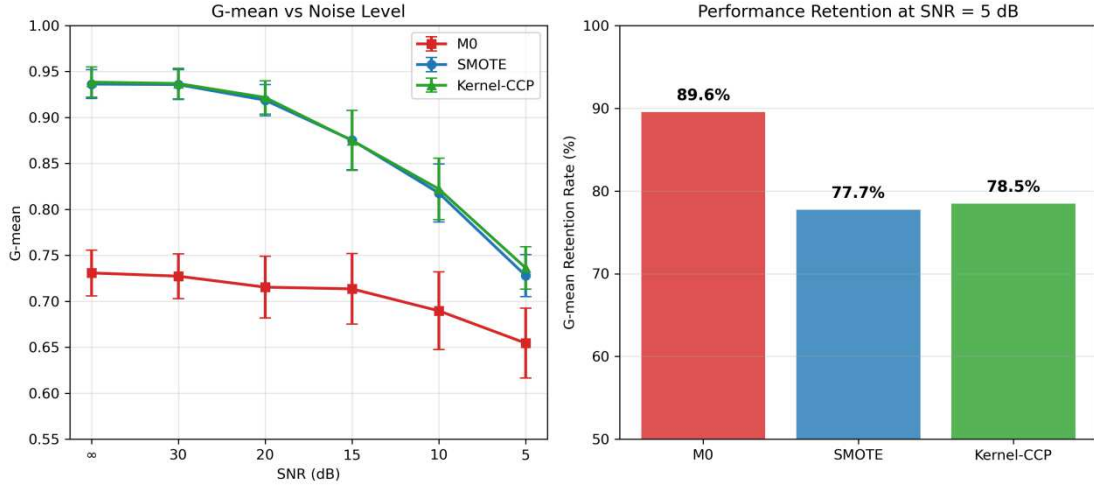


Figure 8 G-mean Degradation Curves and Bar Chart Under Noisy Conditions

5.6 Comprehensive Discussion

Synthesizing all experimental results, the following principal conclusions can be drawn. First, oversampling strategies substantially outperform undersampling in bearing health monitoring under extreme class imbalance; random undersampling suffers from information loss that leads to unacceptably low precision and is unsuitable for this task. Second, Kernel-PAO achieves a G-mean of 0.9384 and an F-value of 0.5990 at IR=1:16, performance on par with the best input-space oversampling methods, with a marginal advantage in the balance between precision and recall. Third, the RBF kernel achieves the best balance between stability and performance and is the default choice for Kernel-PAO. Fourth, the ablation study reveals that kernel-space oversampling itself is the primary source of performance improvement, while the posterior probability module contributes marginally on the current dataset. Fifth, an oversampling multiplier of $N=1$ is sufficient to achieve optimal performance; increasing N leads to overfitting and substantially higher computational cost. Sixth, Kernel-PAO demonstrates robust performance under typical noise conditions and is feasible for engineering deployment. These findings provide clear guidance on parameter configuration and applicable boundaries for the engineering application of the proposed method.

6. Conclusion and Analysis

This study proposes the Kernel-Space Posterior-Probability Adaptive Oversampling method (Kernel-PAO) to address the degradation of support vector machine classification performance caused by class imbalance in rolling bearing health monitoring. Experimental results demonstrate that Kernel-PAO achieves classification performance comparable to, and marginally superior to, the best-performing input-space oversampling methods under conditions of extreme class imbalance.

From the perspective of methodological design, the core advantage of Kernel-PAO lies in migrating the oversampling operation from the input space to the kernel space and unifying SVM training through an augmented Gram matrix, thereby fundamentally ensuring spatial consistency between sample generation and classification decision-making. The posterior-probability-guided boundary-focusing strategy provides a principled probabilistic interpretation for the sampling procedure, directing limited computational resources toward the decision boundary region. Experiments further reveal that an oversampling multiplier of $N=1$ is sufficient to achieve optimal performance; increasing N introduces highly similar synthetic samples that induce overfitting and a dramatic rise in computational cost. The overall computational complexity scales quadratically with the number of samples, which remains acceptable for datasets of typical scale. A comparative analysis of kernel functions indicates that the RBF kernel achieves the best balance between performance and numerical stability.

Several limitations of Kernel-PAO warrant acknowledgment. First, validation is conducted on a single dataset, and the

generalizability of the proposed method to other bearing types and operating conditions has yet to be established. Second, the posterior probability module yields only marginal performance gains on the current dataset, suggesting that its boundary-focusing effect may be constrained by the distributional characteristics of the data. Third, the robustness advantage under severe noise conditions is not pronounced, indicating room for improvement in noise resistance. Fourth, the method is confined to binary classification tasks and has not been extended to multi-class fault diagnosis scenarios. Future work may proceed along the following directions: generalization validation across multiple datasets and bearing types; dimensionality reduction via kernel approximation techniques (e.g., the Nyström Approximation or random Fourier features) to alleviate the quadratic computational overhead; extension to multi-class fault classification settings; and integration with deep learning frameworks to leverage representation learning for enhanced feature quality. Synthesizing the findings of this study, the principal conclusions are as follows. Oversampling strategies substantially outperform undersampling in scenarios of extreme class imbalance. Kernel-PAO achieves a G-mean of 0.9384 and an F-value of 0.5990 at an imbalance ratio of 1:16, performance on par with the best-performing comparison methods. The RBF kernel is the recommended default choice. Kernel-space oversampling itself constitutes the primary source of performance improvement. An oversampling multiplier of $N=1$ represents the optimal configuration. The proposed method demonstrates robust performance under typical noise conditions, confirming its feasibility for engineering deployment.

List of Abbreviations

SVMs	support vector machines
SMOTE	synthetic minority over-sampling technique
PAO	probability adaptive oversampling
CP	CANDECOMP/PARAFAC
ASVB	adaptive support vector boundary
RBF	radial basis function

Availability of data and materials: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests: The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Funding: This work was partially supported by the Guizhou Provincial Science and Technology Plan Project (Grant No. [2023] Generally 256), the Science and Technology Plan Project of Zunyi City (Grant No. zunshikehe HZ[2023]411), and the Zunyi Normal University Doctoral Fund (Grant No. BS[2022]03).

Authors' contributions: Conceptualization, H. X. Jin and H. S. Wang; methodology, H. S. Wang; software, K. H. Sun; validation, H. X. Jin, R. Z. Xiao and H. S. Wang; formal analysis, H. X. Jin.; investigation, Y. X. Ren; resources, K. H. Sun; data curation, R. Z. Xiao; writing—original draft preparation, H. X. Jin; writing—review and editing, K. H. Sun; visualization, H. S. Wang; supervision, H. X. Jin; project administration, H. X. Jin ; funding acquisition, H. X. Jin .

Acknowledgements: Not applicable.

Ethical Declaration

Not applicable.

Consent to Participate

Not applicable.

Consent to Publish

Not applicable.

Clinical Trial Number

Not applicable.

References

- [1] Cao S, Zheng J, Peng G, Pan H, Feng K, Ni Q (2023) Multivariate enhanced adaptive empirical Fourier decomposition and its application in rolling bearing fault diagnosis. *IEEE Sensors Journal* 23(20):24930–24943
- [2] Zhou Y, Kumar A, Parkash C, Vashishtha G, Tang H, Xiang J (2022) A novel entropy-based sparsity measure for prognosis of bearing defects and development of a sparsogram to select sensitive filtering band of an axial piston pump. *Measurement* 203:111997
- [3] De Pater I, Mitici M (2023) Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder. *Engineering Applications of Artificial Intelligence* 117:105582
- [4] Lei Y, Li N, Guo L, Li N, Yan T, Lin J (2018) Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing* 104:799–834
- [5] Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX (2019) Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing* 115:213–237
- [6] Cortes C, Vapnik V (1995) Support-vector networks. *Machine Learning* 20(3):273–297
- [7] Patel V, Bhavsar H (2026) Enhanced classification approach using MI-SVM for imbalanced multi-class datasets. *Scientific Reports* 16:18654
- [8] Akbani R, Kwek S, Japkowicz N (2004) Applying support vector machines to imbalanced datasets. *Proceedings of the European Conference on Machine Learning*, Springer:39–50

- [9] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16:321–357
- [10] Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: A new over-sampling method in imbalanced datasets learning. *Proceedings of the International Conference on Intelligent Computing*, Springer:878–887
- [11] He H, Bai Y, Garcia EA, Li S (2008) ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the IEEE International Joint Conference on Neural Networks*:1322–1328
- [12] Douzas G, Bacao F, Last F (2018) Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences* 465:1–20
- [13] Elkan C (2001) The foundations of cost-sensitive learning. *Proceedings of the International Joint Conference on Artificial Intelligence*:973–978
- [14] Liu XY, Wu J, Zhou ZH (2009) Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 39(2):539–550
- [15] Guo J, Wu H, Chen X, Lin W (2024) Adaptive SV-Borderline SMOTE-SVM algorithm for imbalanced data classification. *Applied Soft Computing* 150:110986
- [16] Mathew J, Pang CK, Luo M, Leong WH (2018) Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE Transactions on Neural Networks and Learning Systems* 29(9):4065–4076
- [17] Soualhi A, Medjaher K, Zerhouni N (2015) Bearing health monitoring based on Hilbert-Huang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation and Measurement* 64(1):2–14
- [18] He H, Garcia EA (2009) Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21(9):1263–1284
- [19] Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter* 6(1):20–29
- [20] Zhou ZH, Liu XY (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* 18(1):63–77
- [21] Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2012) A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 42(4):463–484
- [22] Tax DMJ, Duin RPW (2004) Support vector data description. *Machine Learning* 54(1):45–66
- [23] Jia Z, Yu B (2023) A fault diagnosis method for rolling bearings of wind turbine generators based on MCGAN data enhancement. *SN Applied Sciences* 5:259
- [24] Mao W, Liu Y, Ding L, Li Y (2019) Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: A comparative study. *IEEE Access* 7:9515–9530
- [25] Zhang C, Tan KC, Li H, Hong GS (2019) A cost-sensitive deep belief network for imbalanced classification. *IEEE Transactions on Neural Networks and Learning Systems* 30(1):109–122
- [26] Sehri M, Dumond P, et al. (2023) University of Ottawa Rolling-element Dataset-Vibration and Acoustic Faults under Constant Load and Speed conditions (UORED-VAFCLS). *Mendeley Data*, V1
- [27] Pedregosa F, Varoquaux G, Gramfort A, et al. (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830