Enhancing Machinery Reliability in Lunar Bases: Optimized Machine Learning for Bearing Fault Classification in DC Power Distribution Networks.

Enhancing Machinery Reliability in Lunar Bases: Optimized Machine Learning for Bearing Fault Classification in DC Power Distribution Networks

Muhammad Zain Yousaf ,[1,2], Josep M. Guerrero[,1,2,3], Muhammad Tariq Sadiq [4], Umar Farooq [5]*

1) School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, Zhejiang, 310000, China; zain.yousaf@zju.edu.cn
2) Center for Research on Microgrids (CROM), Huanjiang Laboratory, Zhuji, Shaoxing, Zhejiang, 311800, China; mzainy1@gmail.com
3) Center for Research on Microgrids (AAU CROM), AAU Energy, Aalborg University, 9220 Aalborg, Denmark; joz@energy.aau.dk
4) School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, UK; ms24230@Essex.ac.uk
5) Department of Computer Science, Virtual University, Lahore, Pakistan; BC240439103ufa@vu.edu.pk

Corresponding Authors (*): *Muhammad Zain Yousaf*

- Classifies bearing faults in DC power systems using machine learning (ML).
- Optimal ML architectures used to classify fault patterns based on robust features.
- Reduces operational risks through effective fault classification.
- Critical for space exploration where maintenance is challenging.
- Techniques beneficial for critical, remote power applications.

Abstract

In space missions and extraterrestrial habitats, ensuring the reliability of power systems is critical, particularly for DC distribution networks supporting lunar bases and space stations. These systems rely on rotating machinery such as motors and pumps, making the integrity of rolling bearings essential. There is a significant gap in robust fault detection and classification for such machinery under harsh, variable conditions similar to those in space. Existing machine learning (ML) methods often struggle to capture complex multi-channel patterns in sensor data due to overfitting, hyperparameter sensitivity, and high computational demands. This study proposes an ML-driven framework for fault classification in rolling bearings under extreme conditions, taking into account varying dataset sizes. Using three datasets, the proposed approach employs multi-variate variational mode decomposition (MVMD) and Hilbert-Huang Transform (HHT) to capture fault signatures and extract relevant features. To address overfitting and account for monotonic fault progression, this framework fuses four feature selection methods —Laplacian Score (LS), Minimum Redundancy Maximum Relevance (mRMR), ReliefF, and Mutual Information (mutInf)—with Spearman's rank correlation. The performance of ML classifiers (Neural Networks, Support Vector Machines, Naïve Bayes, K-Nearest Neighbors, Decision Trees, and Ensemble Methods) is optimized by adjusting hyperparameters using Bayesian Optimization (BO), Asynchronous Successive Halving (ASHA), and Random Search (RS), all in parallel settings to improve computational efficiency. These optimizers also help ML architectures to adapt according to available datasets of diverse types. Key quantitative results show that the ASHA-optimized ML model performs well with larger datasets, providing an overall accuracy of 99.94% with the reduced computational load. Meanwhile, BO and RS attained accuracies of 99.90% and 98.0%, which proved effective for scarce datasets. This innovative framework integrates signal decomposition, feature selection, and optimization techniques, creating an efficient predictive maintenance tool. It improves fault classification, boosting the reliability of machinery in extraterrestrial environments and enhancing the safety and sustainability of long-term space missions.

Keywords: Fault Recognition, Rolling Element Bearing (REB), Machine Learning, Lunar Bases, Hyperparameter Optimization.

1. Introduction

Motivation

A DC distribution system is vital for future lunar bases and space stations as it allows efficient, reliable power management, especially in the use of renewable energy sources such as solar power. NASA's Artemis Base Camp, projected for the lunar South Pole by the early 2030s, is projected to cost around $93 billion for its first three phases, with additional investments to support habitats, laboratories, and transportation systems [1, 2]. This pioneering base will heavily rely on DC power systems, which will harness solar energy to power critical missions while minimizing energy loss and optimizing power management in the Moon's extreme conditions. In Europe, the European Space Agency multi-national base concept is planned for the late 2020s, and it is estimated to cost €5 billion to €10 billion [3]. This multi-national base concept will also leverage DC grids to stimulate scientific research and international cooperation.

Similar to Western initiatives, China's space program is also undergoing rapid development, showcasing significant innovation with its International Lunar Research Station (ILRS) scheduled by 2035[4]. The Chinese ILRS is estimated to cost between $20 billion and $25 billion, relying heavily on solar power and DC systems to store and distribute energy efficiently for robotic and crewed missions. However, a critical challenge for these advanced infrastructures is the maintenance of rolling bearings in essential machinery such as motors, pumps, and cooling systems. These bearings are integral to the operation of rotating machinery by allowing rotors to rotate and produce mechanical energy. In this context, over 40% of mechanical failures are due to bearing issues [5],[6]. Moreover, a bearing failure could also compromise astronaut safety and mission success if vital systems are disrupted. Therefore, the integration of early and advanced fault monitoring and classification techniques for rolling bearings is imperative in the challenging environment of space.

Literature review

Bearing fault monitoring and classification is commonly conducted using three techniques: 1) physical-based techniques [7, 8], 2) model-based techniques [9, 10], and 3) knowledge-based techniques [5]. The physical-based techniques help to analyze the spectrum of measured signals from the rolling bearings (RB) to identify abnormalities that may indicate faults. On the other hand, model-based techniques employ complex equations to simulate bearing dynamic behaviour or degradation phenomena. However, the bearing is a complex system, and their degradation mechanism is stochastic in nature, plus these models have high computational costs, and it is difficult to capture all the small details by mathematical models. Hence limiting their adaptability. As an example, it is used only in 10% of real-world applications to detect bearing faults inside induction motors [11]. In comparison, knowledge-based techniques use fuzzy systems or machine learning (ML) models to recognize patterns and classify fault types. This paper focuses on ML-driven techniques to monitor and classify bearing faults. This is because these techniques can handle the complex and challenging environments of space, often achieving high accuracy to analytically model bearing degradation through their ability to capture intricate patterns [12]. Moreover, ML-driven predictive maintenance has reduced machine shutdown time [13], prolonging the lifespan of equipment, further bolstering space mission productivity, and increasing safety.

Traditionally, ML-driven techniques for fault classification involve four basic steps: 1) The primal step is preprocessing (noise removal and data preparation from vibrational signals); 2) feature extraction (signal decomposition and extracting useful information from different domains to capture linear and non-linear aspects of the signal). For instance, phase entropy [14], quadrant entropy [15], and swarm intelligence optimization entropy [16] have been able to extract rich and hierarchical information from decomposed signals in recent times. These methodologies represent the forefront of innovation in bearing fault diagnosis, addressing issues such as signal non-stationarity, sensitivity in noise, and high-dimensional data complexities. 3) Feature selection (FS) (selecting the most relevant features to avoid the "curse of dimensionality" and overfitting for better generalization); and 4) classification (estimating the difference between two or more classes). A summary of the most used ML-driven techniques based on Neural Networks (NN), Support Vector Machine (SVM), Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Classification Trees are elaborated in Table 1.

Table 1. Related work summary of Rolling Bearing fault classification using Machine Learning (ML)

Related work summary of Rolling Bearing fault classification using Machine Learning (ML)						
Serial Number	ML Models	Signal Decomposition	Extracted Features	Feature Selection (FS) Technique	Public Data Sets from Different Sources	No. Of Classes Explored
Single ML Models of NN, SVM, KNN, NB, Classification Trees						
Ref [17]	ANN	Empirical Mode Decomposition (EMD)	Multi-domain (Root Mean Square (RMS), Kurtosis, Skewness, Peak-to-Peak value, Crest factor, Shape factor, Impulse factor, Margin factor, Mean energy, Entropy of decomposed modes)	×	Multiple datasets	7 classes
Ref [18]	ANN	EMD and wavelet packet transform (WPT)	Multi-domain (Skewness, Kurtosis, Crest indicator, Clearance indicator, Shape indicator, Impulse indicator, the signals of WPT, and the intrinsic mode functions (IMFs) of EMD)	Distance Evaluation Technique	Multiple datasets	4 classes
Ref [19]	ANN	Wavelet Packet Decomposition (WPD)	Single-domain (WPD coefficients energy)	×	Case Western Reserve University dataset (CWRU dataset	4 classes
Ref [20]	SVM	Local mean decomposition (LMD)	Single-domain (Sample entropy and Energy Ratio)	×	CWRU dataset	4 classes
Ref [21]	SVM	Hilbert-Huang Transform (HHT)	Single-domain (Hilbert Marginal Spectrum)	×	FEMTO platform dataset	4 classes
Ref [22]	SVM	EMD	Multi-domain (Mean value, Maximum point, Minimum value, Standard deviation, Peak-to-peak value, kurtosis, Pulse index, Waveform factor, Energy of IMFs of EMD, Shannon entropy)	Modified Genetic Algorithm (GA)	Multiple datasets	Binary classes
Ref [23]	Single models of SVM and NB were analyzed.	Short Time Fourier Transform (STFT)	Multi-domain (Mean, Absolute median, Standard deviation, Skewness, Kurtosis, Crest factor, Energy, RMS, Number of peaks, and Zero crossings, Shapiro test, KL divergence from time and statistical domains; Real value of FFT from frequency domain; STFT modes)	Principal Component Analysis (PCA)	FEMTO, Xi'an Jiaotong University (XJTU), CWRU datasets	Binary classes
Ref [24]	Single models of SVM, KNN, NB, and Classification Trees (Decision Tree).	Fast Fourier Transform (FFT)	Multi-domain (Mean, RMS, Standard deviation, Shape factor, kurtosis, Peak value, Skewness, Crest factor, Impulse factor, Clearance factor, Noise and distortion-related features from time and statistical domains; Peak amplitude, Peak frequency, and Band power from frequency domain)	PCA	CWRU dataset	4 classes
Ref [25]	KNN	Vibration, speed, voltage & other signals	Raw signal	PCA	Historical data	2 classes
Hybrid ML Models of NN, SVM, KNN, NB, DT, Ensemble methods						
Ref [26]	Genetic Algorithm based on ANN	Envelope Detection, Hilbert Transform, FFT	Multi-domain (Information obtained from signal decomposition of FFT and HT)	×	Experimental data	Binary classes
Ref [27]	Adaptive Neuro-Fuzzy based on ANN	FFT	Multi-domain (RMS, Standard deviation, Kurtosis, Skewness, and Specific frequencies)	×	Experimental data	9 classes
Ref [28]	Particle Swarm Optimization-SVM (PSO-SVM)	Statistical analysis, FFT, and Variational Mode Decomposition (VMD)	Multi-domain (16 time and statistical domain features, 13 frequency domain, and instantaneous permutation from entropy domain)	Laplacian Score (LS)	CWRU dataset	12 classes
Ref [29]	SVM optimized by Grid Search (GS)	Ensemble EMD (EEMD)	Single-domain Permutation Entropy (PE)	×	CWRU dataset	4 classes
Ref [30]	Bayesian Optimized (BO) SVM	Recurrence Quantification Analysis (RQA)	Multi-domain (RQA features capture various aspects of the system's stability, periodicity, complexity, and predictability)	×	CWRU dataset	4 classes
Ref [31]	Whale Optimization Algorithm (WOA-	Multi-variate synchrosqueezing WT	Multi-domain (Mean, RMS, Square Root amplitude, Average Amplitude, Maximum	×	CWRU dataset	7 classes

	SVM)		peak, Standard deviation, Skewness, Kurtosis, Peak, Waveform and Pulse index, Barycenter frequency, Variance)			
Ref [32]	Cuckoo Search SVM (CS-SVM)	VMD	Single-domain (Permutation Entropy)	✖	CWRU dataset	4 classes
Ref [33]	CNN-SVM	✖	Autonomous CNN features	Min-Redundancy/Max-Relevance (mRMR)	CWRU dataset	10 classes
Ref [34]	Weighted KNN	FFT, WPD, VMD	Multi-domain (13 time-domain, 4 frequency-domain, 17 entropy-based features)	ReliefF algorithm	CWRU and MFPT dataset	4 & 7 classes
Ref [11]	CNN-NB	✖	Autonomous CNN features	✖	Helical Gearbox and XJTU dataset	7 & 5 classes
Ref [6]	Ensemble of trees (Random Forest)	WPD	Single-domain (Time-frequency features from Wavelet packet decomposition)	Internal voting mechanism	Laboratory dataset	4 classes
Ref [35]	Ensemble of tree (Random Forest)	Basic preprocessing on vibration data	Multi-domain (Statistical and Frequency domain features)	Not specified	CWRU dataset	4 classes
Ref [36]	Ensemble (boosting, bagging, stacking)	Discrete Wavelet Transform (DWT)	Multi-domain (10 different kurtosis and 10 entropy features)	✖	Experimental data	13 classes

Research Gaps

The literature review in Table 1 highlights key characteristics of the present research, including signal decomposition (SD), feature extraction, FS methods for derived features, sources of databases, and classification classes for single and hybrid ML models. After thorough investigations and reviewing related work summaries, we outlined the following limitations.

***1) Cross-Channel Dependencies and Signal Variance*:** The primal step in feature extraction is SD. Table 1 demonstrates recent works on different SD techniques based on FFT, STFT, HHT, WT, EMD, and VMD, both in discrete and continuous forms and envelope analysis. However, these SD approaches often struggle to capture essential fault signatures due to a lack of cross-channel interdependence analysis, which is critical as rolling bearing failures overall produce multi-variate, interdependent signals across frequency bands. Furthermore, these methods do not adapt well to signal drift (a common occurrence in bearing vibrations where frequencies change over time due to changing load and speed conditions). Advanced techniques like multi-variate synchrosqueezing WT [31], EMD [37], and empirical WT (EWT) [38] have been proposed to address this. However, these approaches remain constrained by their poor performance on long-duration signals, sensitivity to noise, delicate to sampling rates, and reliance on empirical parameters. Another obstacle is the dependence on adaptive wavelet filters for accurate spectrum segmentation, which creates difficulties for real-time monitoring systems that must adjust to shifts in the bearing's operating environment, like the lunar base.

2) Limited Analysis of FS Methods: The literature review in Table 1 shows that FS methods, namely PCA and GA, etc., have been used to derive feature candidates from decomposed modes, but this led to unsatisfactory results as valuable information was lost and model performance was affected. To address this issue, several uni and multi-variate FS methods (such as LS, mRMR, ReliefF, etc.) were studied, but no comparative investigation has been conducted between these FS methods. Moreover, these studies use a single or limited database, which limits the scope and application of the findings.

3) Neglect of Monotonic Characteristics: In these limited databases, monotonic characteristics (one-directional non-linear trends) are often present in the oscillation amplitude of bearings and serve a crucial role in tracking degradation trends in fault conditions [39, 40]. Current FS methods often overlook these trends, which limits their scope to track subtle degradation from initial damages.

4) Limitations of Single and Hybrid ML Models: In Table 1, recent studies are categorized into single- and hybrid models for RB fault classification. Single ML models use an exclusive single technique for classification (such as NN, SVM, NB, KNN, and Classification Trees), while hybrid ML models combine two techniques (such as ANN with GA, SVM with PSO, and so on). However, the studies in Table 1 that focus on single ML models overlook the following limitations: 1) NNs are prone to overfitting and require large training sets. 2) SVM is sensitive to hyperparameters and needs fine-tuning along with relatively high training and prediction times. 3) NB has limited interpretability with correlated data. 4) KNNs are computationally intensive and suffer from the curse of dimensionality 5) Classification Trees similar to ANN suffer overfitting. If input data has any slight variations, it can lead to misclassification and often fail to capture complex, non-linear patterns.

In comparison, hybrid models play a crucial role in improving classification accuracy and robustness over single models. While some previous studies have demonstrated the improved classification performance of hybrid models, these efforts often combine only two techniques. Hence, their adaptability to complex data conditions is limited. Moreover, challenges such as increased model complexity, larger data requirements, higher computational time, and precise tuning for large numbers of hyperparameters are frequently overlooked with hybrid models. Therefore, to genuinely assess the effectiveness of hybrid models, it is essential to train widely used ML algorithms with different optimizers to determine the true efficacy of the ML algorithm. Meanwhile, Grid search [29] and Bayesian optimization [30] have been reported in

the literature for ML model optimization with limited hyperparameters, and their traditional implementations don't run parallel settings to take full advantage of multi-core processors.

Contributions

The following contributions have been made to overcome the aforementioned research gaps and facilitate advancement.

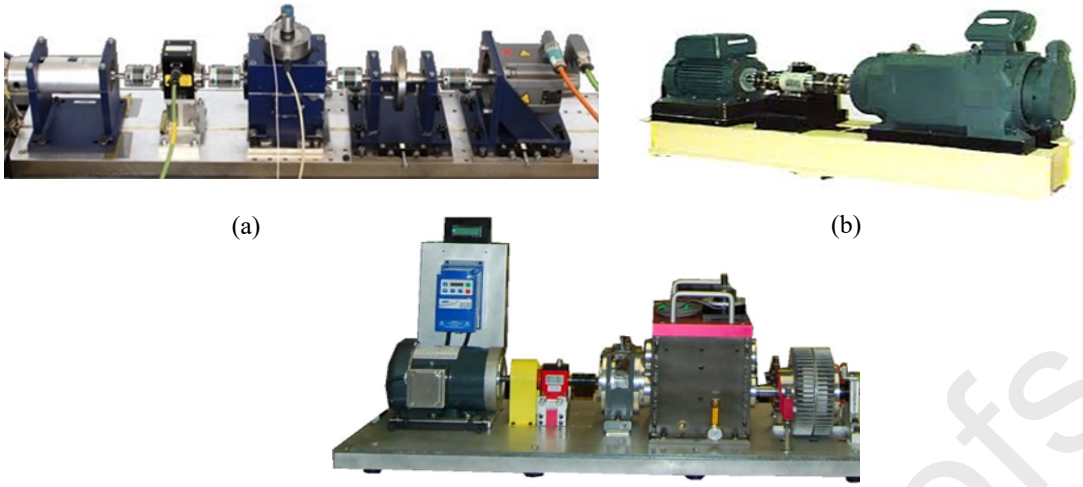
- An in-depth analysis of the proposed techniques was done by collecting vibrational data through three experimental testbeds based on bearings and gearboxes. The collected data is denoised and normalized to implement SD. In doing so, the MVMD is proposed to capture the interdependence of signals across multiple channels, in particular for long-duration signals with varying load and speed conditions.
  1. The MVMD method addresses the limitations of the multi-variate EMD extension, such as the lack of a strong mathematical foundation, limited effectiveness with small sample sizes, and the problem of mode mixing across different channels.
  2. Offering enhanced decomposed mode separation resolves a key challenge in the multi-variate extension of SST.
  3. Unlike the multi-variate extension of EWT, the MVMD method simplifies the process by eliminating the need for pre-defined wavelet filter bank boundaries.
  4. Following this, HHT is applied on enhanced nodes to obtain a detailed understanding of non-stationary patterns. This dual-stage MVMD-HHT processing enhances the ability to detect subtle, initial faults in rolling bearings, which are often characterized by weak, non-stationary signal components that could be overlooked via traditional methods.
- Unlike Pearson's correlation, Spearman's rank (SR) correlation is a pre-screening step that helps capture non-linear relationships, such as monotonic characteristics, and discard features with low correlation [41]. We have proposed four feature selection methods for automated FS (i.e., LS, mRMR, ReliefF, and Mutual Information (mutInf) FS methods) and integrated them with SR correlation, resulting in multi-domain features that help to provide a holistic and refined understanding of vibration signals.
- Before training, an extended Yeo-Johnson method was introduced that helped to make feature distributions more Gaussian-like, remove outliers, and improve model performance.
- This study also facilitates a rigorous optimization and training process via three optimizers: Asynchronous Successive Halving Algorithm (ASHA), Bayesian Optimization (BO), and Random Search (RS). These optimizers facilitate adaptive hyperparameter tuning with parallel settings, improving model accuracy while accounting for computational constraints.
- By optimizing hyperparameters, ASHA, BO, and RS help adapt ML models to specific data characteristics under different speeds and load working environments, addressing the unique demands of lunar base applications where fault conditions face unpredictable circumstances.
- Instead of manual exploration, this automated process explores the ML model with the highest accuracy from six widely used ML algorithms (NN, SVM, NB, KNN, DT, and Ensemble) across diverse testing conditions,
- Benchmark comparison of FS and optimizer techniques highlight the individual strengths of specific datasets and provide guidelines for future implementation in space-related fault detection tasks.
- Our core contribution lies in the novel integration and adaptation of existing techniques to lunar base environments, which present unique challenges such as long-duration signals from distributed sensors, noise, and dynamic load and speed variations of bearings in DC power systems.

Paper Organization

The remainder of the paper is organized as follows. Section 2 provides the databases included in this analysis, whereas Section 3 provides a description of the planned approach. The findings, analytical discussion, and comparison studies are listed in Section 4, Section 5, 6 ,7 and 8. Finally, Section 9 onwards the concluding remarks of the study.

2. Materials

This study aims to develop a data-driven method for fault classification in rolling bearings. Therefore, the initial step is to prepare vibration datasets for verification purposes. To do so, three datasets were collected based on real and artificially generated bearing faults. Figure 1 presents the experimental environments for different datasets, and their description is as follows, whereas the supplementary file includes signal waveforms (Figure A) of the bearing in different states:



(a) (b)

(c)

Figure 1: Experimental platforms (a) PU experimental platform (b) CWRU experimental platform (c) DDS experimental platform

Table 2. Experimental data set description

Rolling bearing data set description							
Bearing State	**Class**	**Health condition**	**Fault Degree**	**Motor speed (rpm)**			
CWRU dataset	1	Healthy	Normal (NOR)	1730	1750	1772	1797
	2	IR faults	0.07//	1730	1750	1772	1797
	3		0.14//	1730	1750	1772	1797
	4		0.021//	1730	1750	1772	1797
	5	Ball faults	0.07//	1730	1750	1772	1797
	6		0.14//	1730	1750	1772	1797
	7		0.021//	1730	1750	1772	1797
	8	OR faults	0.07//	1730	1750	1772	1797
	9		0.14//	1730	1750	1772	1797
	10		0.021//	1730	1750	1772	1797
PU dataset	Class	Health condition	Fault Degree	Motor speed			
	1	Healthy	Normal	1500 rpm with a load torque of 0.7 kgm^2sec^{-2}			
	2	Combined OR & IR	Multiple damages				
	3	IR faults	Single, repetitive, and multiple damages				
	4	OR faults	Single and repetitive damages				
DDS dataset	Class	Health condition		Motor frequency			
	1	Rolling race defect (RED)		The motor frequency is 20 Hz with 0 load			
	2	Inner race defect (IRD)					
	3	Outer race defect (ORD)					
	4	Healthy (NOR)					

2.1 CWRU Bearing Dataset:

The experiment conducted for the Case Western Reserve University (CWRU) dataset used a two-horsepower (hp) reliance electric motor, and vibration signals from faulted bearings were collected via an accelerometer, which is placed at the drive end of the motor housing [42]. While collecting vibrational data, faulted bearings are situated inside the test motor with a motor load varying from 0 to 3 hp with motor speeds of 1720-1797 rpm. Meanwhile, there are three types of faults, namely bearing ball, inner raceway, and outer raceway faults, with a range of diameters from 0.007 to 0.040, respectively. Table 2 shows details of the operational condition. Taking fault diameter, operational condition, and motor speed into account, there are ten classes that need classification. The vibrational signals for given classes are divided into 1024 data points for input segments with a sampling frequency of 12 kHz at different time durations. Hence, this creates compact information in data sets that helps to examine the proposed Method under various operating conditions.

2.2 PU Bearing Dataset:

Although CWRU is widely used as a benchmark to investigate proposed studies, it lacks real damage since faults are artificially generated. Hence, the Paderborn University (PU) dataset is also utilized, which has real bearing damages with

combined defects [43]. The experiment was conducted to collect datasets with a 425-W permanent magnet synchronous motor (PMSM) with a motor speed of 1500 rpm and load torque = 0.7 Nm, whereas the radial force on the bearing is 1000 N. The Accelerometer is mounted at the top end of the rolling bearing module that collects vibrational signals at the sampling frequency of 64 kHz. In this study, only accelerated lifetime defects from the PU dataset were utilized, which mimics real damages. As shown in Table 2, there are four classes: three faulty operational classes and one undamaged. To construct the feature vector and classify these classes, the input segment is decomposed into 2048 data points.

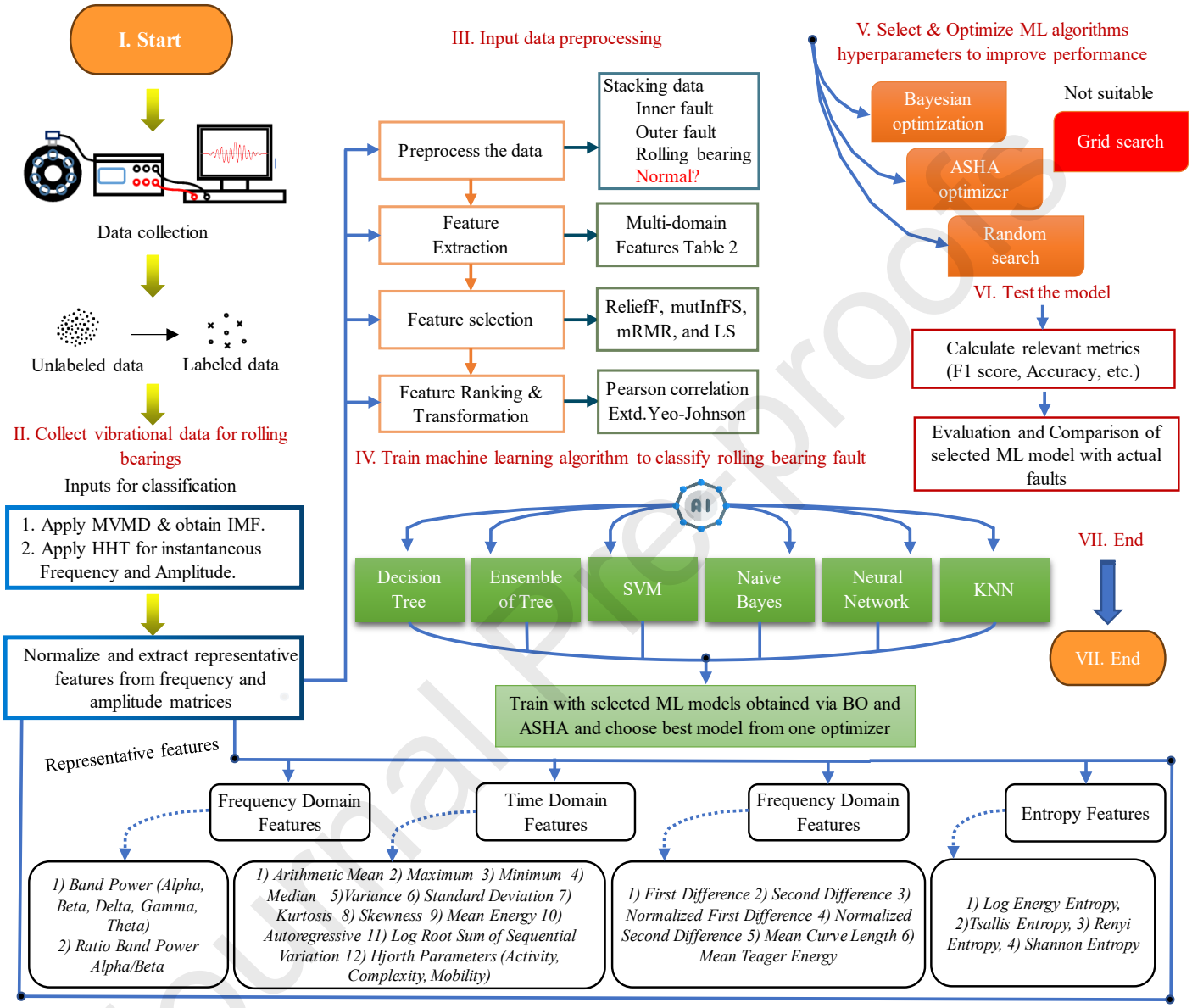


Figure 2: Proposed Framework

2.3 DDS bearing Dataset:

To further verify the robustness and generality of the proposed Method, this study also uses SpectraQuest's Drivetrain Dynamics Simulator (DDS) for experiments. DDS aims to simulate industrial drivetrains for educational and experimental purposes. The experimental design of the test bench has six parts, including 1) a 2-stage planetary gearbox, 2) a 2-stage parallel shaft gearbox with a rolling bearing inside, 3) the speed regulator, 4) the driving motor, 5) the programmable magnetic brake, and 6) the brake regulator [44]. ER-16K rolling bearing model is used with bearing and rolling element bearing diameters of 15.16 mm and 3.125 mm, respectively. This model has nine rolling elements, and the contact angle is zero degrees. The motor frequency is 20 Hz with 0 load, and the sampling frequency is 12.8 kHz. A total of 200k data points were produced for approximately 15.625 sec. In doing so, four different classes were generated: 1) normal state, 2) inner race defect, 3) outer race defect, and 4) rolling race defect. For each class, 200 groups are taken, each 2048 in length.

3. Methods

Figure 2 provides a clear visualization of the proposed framework. At first, the proposed approach denoised the real

bearing damages by employing the empirical Bayesian Method with a Cauchy prior [45]. The sym4 wavelet is applied with a posterior median threshold. Following denoising, MVMD and HHT are sequentially applied for robust and adaptive data decomposition, where HHT matrixes of instantaneous frequency and amplitude are used to extract multi-domain features. Subsequent to feature extraction, pre-screening with Spearman's rank correlation ensures high-quality data feeds into the machine-learning algorithms. We extensively studied four feature selection techniques: ReliefF, Mutual Information FS (mutInfFS), Minimum Redundancy Maximum Relevance (mRMR), and Laplacian Score (LS). These techniques aim to capture relevant and non-redundant features [46, 47]. Following feature selection, the extended Yeo-Johnson method was implemented to transform the feature distribution into a more symmetrical form, enhancing the robustness, interpretability, and accuracy of the machine-learning models [48]. Six machine-learning models were then trained with BO with parallel settings and ASHA optimizers. These optimizers are used to select and validate the best classification model and architecture while ensuring robust performance. In addition, random search (RS) and grid search (GS) were also studied to compare with the aforementioned optimizers. During the initial stages of the research, it was observed that GS is not feasible for large-scale hyperparameter optimization. This is due to the intensive computational nature of grid search, which explores all possible hyperparameter combinations within a pre-defined grid. Such an exhaustive search becomes impractical when dealing with extensive hyperparameter spaces, as it requires significant computational resources and time. The following sections describe each step in the proposed framework.

3.1 Signals decomposed via MVMD

This study empirically obtained 5 joint modes by applying the multi-variate variational mode decomposition (MVMD) to the CWRU, PU, and DDS datasets. These datasets contain recordings of vibrational signals acquired from multiple sensor channels (*N*) under different fault conditions. The multi-variate time series from the CWRU, PU, and DDS datasets are represented as $x(t) = [x_1(t), x_2(t), \text{¼}, x_n(t)]$. The goal of MVMD is to extract m number of multi-variate modulated oscillations $y_m(t) = [y_1(t), y_2(t), \text{¼}, y_N(t)]$ from these multi-variate time series signals. These modes help understand the mechanism of oscillatory behavior under different fault conditions. However, to accomplish these objectives, the collected decomposed mode's bandwidth must be kept to a minimum, ensuring modes are as narrowband as possible. This enables effective isolation and analysis of specific oscillating components while preserving the original signal [49]. The whole process achieved as follows:

1) For a given multi-variate time series, $y_m(t)$ expressed in analytical form as:

$$y_m(t) = y_m^+(t) = y_m(t) + j\mathrm{H}(y_m(t)) = \begin{bmatrix} y_{m,1}^+(t) \\ y_{m,2}^+(t) \\ \mathrm{M} \\ y_{m,P}^+(t) \end{bmatrix} \tag{1}$$

H denotes the Hilbert transform that helps to transform multi-variate time series into analytical signals. To determine the bandwidth of $y_m(t)$, calculate the L_2-norm of the gradient function of the harmonically shifted $y_m^+(t)$.

2) The cost function used in the decomposition process is modified for MVMD. The aim is to minimize this cost function to achieve the best decomposition:

$$\sum_m \left\| \partial_t \left(y_m^+(t) e^{-jw_m t} \right) \right\|_2^2 \tag{2}$$

It is important to note that a single-frequency element w_m was used in the harmonic mixing of the total vector $y_m^+(t)$.

3) Afterward, the modulated multi-variate oscillatory bandwidth is calculated by adjusting the spectrum range of all channels $y_m^+(t)$ via w_m and applying the Frobenius norm ($\|\cdot\|_F$ a measure of matrix size) to the resulting matrix. This is formulated as:

$$\left[\sum_{m,n} \left\| \partial_t \left(y_{m,n}^+(t) e^{-jw_m t} \right) \right\|_F^2 \right] \tag{3}$$

In this equation, $y_{m,n}^+(t)$ is the analytic modulated signal corresponds to the channel m and the n mode. By adjusting w_m, we ensure the optimal harmonic mixing, which minimizes the cost function and thus the signal's bandwidth. The complete

constrained variational problem can be formulated as an optimization problem:

$$\min_{y_m, w_m} \sum_{m,n} \left\| \partial_t \left(y^+_{m,n}(t) e^{-jw_m t} \right) \right\|_2^2 \tag{4}$$

subject to:

$$\sum_m y_{m,n}(t) = x_n(t), n = 1,2,...,N \tag{5}$$

A variational problem in MVMD aims to find the optimal set of modes that minimize the bandwidth while reconstructing the original multi-channel signal. To turn the constrained variational problem into an unconstrained one, quadratic penalty b and Lagrangian multipliers (L.M.) were introduced for all channels. ADMM (alternating direction method of multipliers) is used to solve the variational problem by iteratively updating the intrinsic mode functions (IMFs) and their center frequencies. In the spectral domain, computed center frequency and decomposed modes are represented as:

Iteratively update rule for $\hat{y}^{(k+1)}_{m,n}(w_m)$:

$$\hat{y}^{(k+1)}_{m,n}(w_m) = \frac{\hat{x}_n(w_m) - \sum_{i \neq m} \hat{y}^{(k)}_{i,n}(w_m) + \frac{\hat{l}_n(w_m)}{2}}{1 + 2b(w_m - w^{(k)}_m)} \tag{6}$$

The update rule for center frequency $w^{(k+1)}_m$:

$$w^{(k+1)}_m = \frac{\int_0^\infty |\hat{y}_{m,n}(w_m)|^2 \, w_m dw_m}{\int_0^\infty |\hat{y}_{m,n}(w_m)|^2 dw_m} \tag{7}$$

Repeat the process until convergence. As a result of following these steps, signals can be decomposed into different modes, where each mode is characterized by specific frequencies, which helps analyze the data.

3.2 HHT on decomposed modes.

The HHT method is suitable for analyzing the aforementioned non-stationary and non-linear decomposed modes of vibration signals from rolling bearings. Unlike wavelet transforms (i.e., DWT, EWT, and CWT), which use pre-defined basis functions, it is an adaptive and data-driven method [50]. Its ability to handle intra-wave frequency modulations, as well as the ability to detect weak faults and low computational requirements, makes it an ideal candidate for analyzing vibration signals from rolling bearings [50, 51]. To achieve these advantages, the Hilbert transform is applied to the collected modes $y_m(t)$, constructing an analytical signal in the following manner:

$$z_m(t) = y_m(t) + j\mathrm{H}[y_m(t)] \tag{8}$$

From the analytic signal, compute the instantaneous amplitude $A_m(t)$ and instantaneous phase $f_m(t)$:

$$A_m(t) = |z_m(t)| = \sqrt{y_m(t)^2 + \mathrm{H}(y_m(t))^2} \tag{9}$$

$$f_m(t) = \arg(z_m(t)) = \arctan\left(\frac{\mathrm{H}(y_m(t))}{y_m(t)}\right) \tag{10}$$

Now differentiate the instantaneous phase to obtain instantaneous frequency:

$$w_m(t) = \frac{1}{2p}\frac{df_m(t)}{dt} \tag{11}$$

This results in capturing feature matrixes that contain the instantaneous amplitude and frequency of each IMF.

3.3 Feature Extraction

Rolling bearing fault diagnosis is challenging due to operational conditions in space and variations in bearing characteristics. To gain deeper insights into complex vibration signals from bearings, we first decomposed the signals into 5 IMFs and then extracted the instantaneous amplitude and frequency for each IMF, resulting in 10 feature matrixes (*5×2=10*). Next, we extracted 33 multi-domain features per matrix from different domains, such as frequency, entropy, and statistical domains, as shown in Table 3.

Leveraging features from multiple domains offers several benefits, as outlined below:

3.3.1 Time-Domain and Statistical Features

A key advantage of time-domain and statistical features is simplicity since these features only consider amplitude

variations in raw vibration signals without any transformations. Another advantage of time-domain features is the ability to detect transient events and impulsive behaviour associated with bearing faults like cracks and spalls, whereas statistical features provide detailed information by analyzing signal properties. In this study, *F1-F23* represents the time domain and statistical features extracted from each input segment of CWRU, PU, and DDS datasets.

Table 3. List of representative features arrangement

List of multi-domain features		
No.	**Feature**	**Remarks**
	Time-Domain Features:	
F1	First Difference	Difference between consecutive data points in the signal, for identifying abrupt changes in vibration signals.
F2	Second Difference	*F2* measures the rate of change of the *F1*. This measures how quickly the first differences change. It shows where the changes are speeding up or slowing down.
F3	Normalized First Difference	*F3* is calculated by taking the *F1* of the signal and then normalizing it. This normalization process ensures the measurements are consistent under different operational conditions by accounting for varying amplitudes.
F4	Normalized Second Difference	*F4* measures how quickly the normalized first differences change providing insights into the acceleration and deceleration of the signal.
F5	Mean Curve Length	Measures the total variation in the signal's amplitude. This can indicate factors such as wear, cracks, or impacts within a bearing, providing insights into the condition and performance of the system.
F6	Mean Teager Energy	Teager energy provides a measure of the instantaneous energy of the signal, which helps detect faults that introduce high-frequency vibrations. These components are difficult to detect via traditional energy measures.
	Statistical Features:	
F7	Arithmetic Mean	The increase in the mean value of the vibration signal suggests that there is a change in the overall energy or force within the bearing system. This change can be attributed to various factors such as wear, misalignment, imbalance, or other forms of damage.
F8-11	Autoregressive Model	When fault occurs, vibration signal shows periodic peaks every few milliseconds. The periodicity corresponds to the rotation speed of the bearing and the position of the fault. It captures these intricate periods over time.
F12	Maximum	Identifies the peak values of the signal. Faults often introduce large spikes.
F13	Minimum	Faults often introduce dips in the signal, which is useful for identifying anomalies.
F14	Median	Represents the middle value of the signal. When the median shifts, faults can be detected.
F15	Variance	The spread between numbers in a data set. Faults typically introduce higher variance.
F16	Standard Deviation	Data dispersion relative to mean. Increased value reflects higher variability due to faults.
F17	Kurtosis	It is a measure of the presence of outliers or extreme deviation due to bearing faults like cracks, spalls, pitting.
F18	Skewness	For early detection, it measures the asymmetry of the signal distribution. Fault signal indicate +,- skewness.
F19	Mean Energy	It represents the average power of the signal. The overall energy level of the signal increases ↑ during fault.
F20	Log Root Sum of Sequential Variation	Detects increased variability in vibration signals. By capturing the total amount of change over time, this measure helps maintenance teams identify faults early and take proactive steps to address them.
F21	Hjorth (Activity)	Reflects signal variance. ↑ activity indicates vibration energy, which indicates faults.
F22	Hjorth (Complexity)	Describes the intricate frequency composition and rate of change of the signal. Increased complexity indicates the presence of multiple vibration sources or irregular behavior within the bearing.
F23	Hjorth (Mobility)	Measures the square root of the variance of the signal's first derivative, indicating speed of changes in signal. Higher mobility indicates irregular behaviors due to transient such as impacts, shocks or sudden changes.
Frequency-Domain Features: The vibration signal is often analyzed within distinct frequency bands to isolate and identify specific types of faults**.**		
F24	Band Power (Alpha)	Band power measures the signal power within specific frequency ranges. Alpha (8-12 Hz): Capture low-frequency vibrations due to slow-moving components in bearings.
F25	Band Power (Beta)	Beta (12-30 Hz): Often associated with moderate-speed rotations and can indicate issues such as faults
F26	Band Power (Delta)	Delta (0.5-4 Hz): Help indicate low-frequency phenomena, such as bearing structural vibrations.
F27	Band Power (Gamma)	Gamma (30-100 Hz): Reflects high-frequency components related to early-stage faults.
F28	Band Power (Theta)	Theta (4-8 Hz): May capture intermediate frequency vibrations, indicates structural defects or imbalances.
F29	Ratio Band Power Alpha/Beta	Compares power in different frequency bands, useful for identifying shifts in spectral content. Such shifts can indicate early changes in the vibrational characteristics due to faults.
	Entropy Features:	
F30	Shannon Entropy	Quantify randomness in the signal. In the context of rolling bearing faults, an increase in entropy signifies more complex and erratic vibrations caused by defects.
F31	Log Energy Entropy	A bearing fault may cause an increase in energy at specific frequencies, leading to change in overall energy distribution. *F31* measures the spread of energy across the different components of the vibration signal.
F32	Tsallis Entropy	Tsallis entropy is a generalization of the Shannon entropy It provides a robust measure of complexity that can capture the non-linear and non-extensive characteristics of vibration signals from faulty bearings.
F33	Renyi Entropy	Higher orders of Renyi entropy are sensitive to the impulsive nature and irregularities introduced by bearing faults, enabling better quantification of the signal complexity.

3.3.2 Frequency-Domain Features

Bearing fails such as IR, OR, and ball faults produce distinct frequency signatures based on the bearing geometry and rotational speeds. These signatures can be identified in the frequency spectrum for robust fault classification [52]. With

this in mind, in order to extract frequency-domain features, we estimate the power spectral density using the Welch method with a Hamming window of size 50 while fixing a 50% overlap among consecutive segments of the waveform. According to Table 3, features from *F24-F29* provide comprehensive information about the fault signatures.

3.3.3 Entropy Features

Vibration signals from rolling bearings have complex, uncertain, and non-linear characteristics in faulty conditions. Entropy measures can adapt to various operational conditions that help capture non-linear dynamics induced by IR, OR, and ball faults. Moreover, these features complement time-domain, frequency-domain, and statistical features, improving overall analytical capabilities. Therefore, several entropy features (*F30-F33*) were calculated to detect transient events and capture these complexities and uncertainties. Figure 3 shows the 2D projection of features from multi-domains; it is demonstrated that non-faulty features are clustered separately from faulty features.

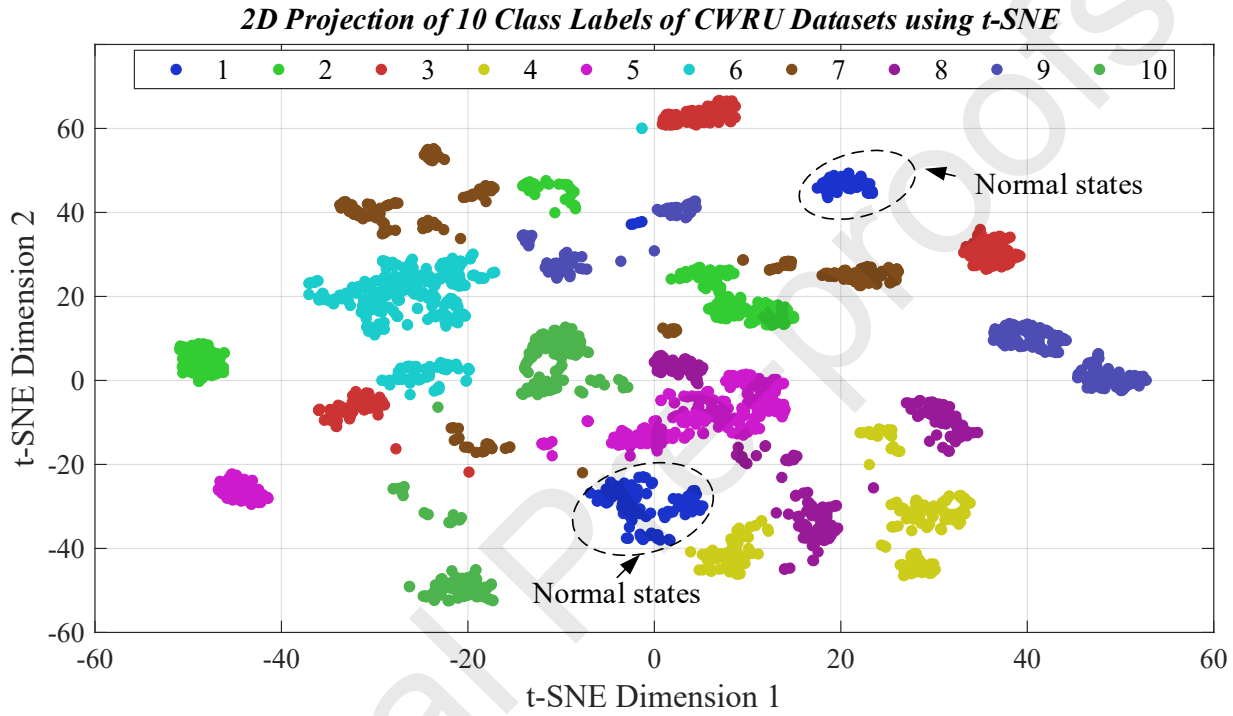


Figure 3: 2D projection of features from multiple domains via t-SNE (t-distributed stochastic neighbour embedding)

3.4 Feature Selection

Multi-domain features shown in Table 3 were collected for each input segment of CWRU, PU, and DDS multi-channel signals, forming a large feature matrix. In the case of the CWRU input segment with a length of 1024, the feature vector of each sample acquired 330 multi-domain features (*33 features* × *10 matrixes* = *330*). Similarly, for a PU input segment with a length of 2048, 330 multi-domain features were obtained. According to Table 2, the CWRU, PU and DDS datasets contain 4000, 1800 and 400 samples, creating data matrixes of size 330×4000, 330×1800 and 330×400, respectively. This entire process is part of feature extraction. The next process is feature selection, in which different feature selection (FS) techniques were implemented to rank features and select the best features based on empirical method.

FS methods can be categorized into wrapper [53] and filter [54]. When choosing the suitable subset of features while evaluating features, wrapper runs a greedy backward, forward, or bi-directional feature search to find the local optimum in the search domain. As a result, most wrapper-based methods are exhaustive (involving 2^d combinations, where d represents the number of features) and computationally demanding. Contrary to this, filter FS methods are simple and lightweight. This is because filter FS techniques are scheme-independent, which helps to remove redundant information without relying on a specific classifier. These filter FS techniques offer attractive results in real-time applications with simplicity, which is another motivation for using them and testing different filter FS techniques. Moreover, we explore the effectiveness of combining Spearman's Rank Correlation [55] with 4 filter FS techniques to select suitable features, namely ReliefF, mutInfFS, mRMR, and LS with multi-domain features for the first time in this study.

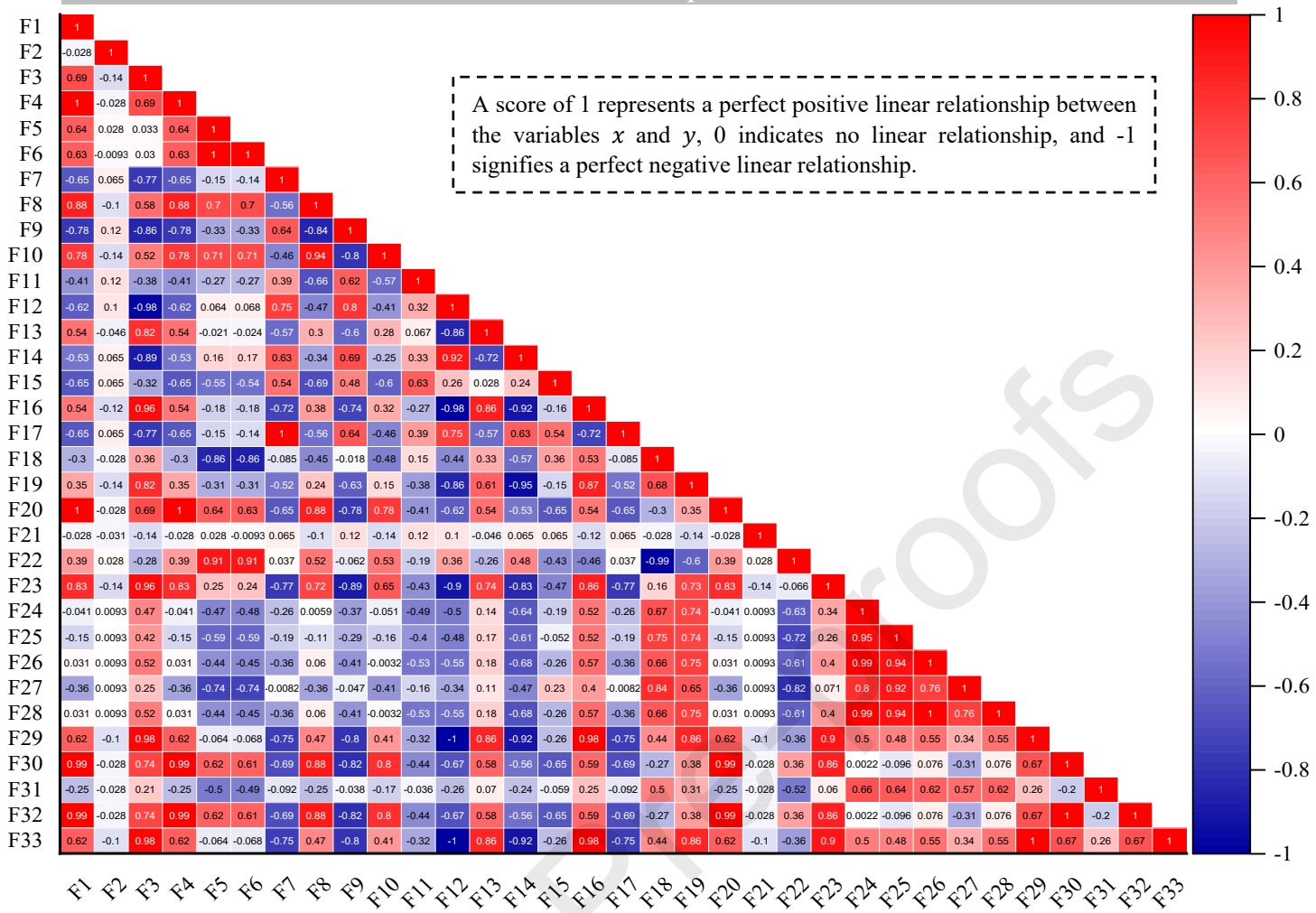


Figure 4: Statistical significance and association of input and output variables via Spearman's rank correlation (multi-domain features extracted from an instantaneous frequency of IMF1)

3.4.1 Spearman's Rank Correlation (*SR*)

The SR correlation coefficient is used to determine the statistical significance of data, as shown in Figure 4. For FS techniques like mutInfFS and mRMR, the rank-based monotonic correlation is helpful in determining how one feature affects or is associated with another feature or the target variable [56]. This is important because accurate fault classification relies on detecting trends of degradation in rolling bearing data. On the contrary, methods like ReliefF or LS often overlook these monotonic relationships, which leads to the choice of inadequate features [56, 57]. These shortcomings can be addressed by computing the SR correlation coefficient as:

$$SR(Coefficient) = 1 - \frac{6\sum b_i^2}{e(e^2 - 1)} \tag{12}$$

Where b is the difference between the ranks of corresponding values of features and targets, e is pairs numbers. Following the pre-screening with *SR*, we apply the following 4 FS techniques:

a) ***Laplacian Score (LS):***

LS is an unsupervised univariate algorithm that uses the nearest graph to model the local structure for the whole feature set and scores each feature according to its locality. Let A represent an affinity matrix, its degree matrix be D, and Laplacian matrix as $L = D - A$. Then, the Laplacian score of an individual feature h is calculated as [58]:

$$LS(h) = \frac{\tilde{h}^T L \tilde{h}}{\tilde{h}^T D \tilde{h}}, \quad adjusted\text{-}vector\ \tilde{h} = h - \frac{h^T D \mathbf{1}}{\mathbf{1}^T D \mathbf{1}}\mathbf{1} \tag{13}$$

The goal is to select k features from the feature matrix that minimize the following objective function:

$$\min_{j_1,\ldots,j_k} \sum_{j=1}^{k} LS(h_{j_j}) = \frac{\tilde{h}_{j_j}^T L \tilde{h}_{j_j}}{\tilde{h}_{j_j}^T D \tilde{h}_{j_j}}, \quad j_j \in \{1,\ldots,l\}, p \neq q \Rightarrow j_p \neq j_q \tag{14}$$

b) *ReliefF:*

As an extension of the Relief family of algorithms, ReliefF is a supervised algorithm that handles multi-class problems. ReliefF determines the rankings of features based on its ability to distinguish between classes that are close to each other. Using the Relief method, the feature ranking for q randomly sampled instances is shown using the following mathematical formulation [59]:

$$Relief(h_j) = \frac{1}{2}\sum_{v=1}^{q}\left[D(h_{v,j} - h_{v,NM(x_v)}) - D(h_{v,j} - h_{v,NH(x_v)})\right] \tag{15}$$

where $D(\cdot)$ denotes the distance function, $h_{v,j}$ is the value of the j^{th} feature for the v^{th} instance, $h_{v,NM(x_v)}$ is the value of the j^{th} feature for the nearest miss (the nearest instance from a different class), and $h_{v,NH(x_v)}$ is the value of the j^{th} feature for the nearest hit (the nearest instance from the same class). The following mathematical formulation describes the multi-class extension of the ReliefF algorithm [59]:

$$ReliefF(h_j) = \frac{1}{q}\sum_{v=1}^{q}\left[-\frac{1}{s_{x_v}}\sum_{x_t \in NH(x_v)} D(h_{v,j} - h_{t,j}) + \sum_{u \neq y_{x_v}} \frac{1}{s_{x_v,u}} \frac{P(u)}{1 - P(y_{x_v})} \sum_{x_t \in NM(x_v,u)} D(h_{v,j} - h_{t,j})\right] \tag{16}$$

Considering this equation, x_v is the class category denoted by y_{x_v}. $P(u)$ is the probability of an instance belonging to the category u. $NH(x)$ and $NM(x,u)$ denotes the sets of nearest hits (nearest instances with the same class x) and nearest misses (nearest instances from different classes y), respectively. The sizes of these sets are specified by s_{x_v} and $s_{x_v,u}$.

c) *Mutual Information (mutInfFS):*

A univariate feature ranking algorithm based on Shannon entropy determines the relationship between two features. The mutual information can be calculated as follows [60]:

$$mutInf(C_j, R) = \sum P(C_j, R)\log\left(\frac{P(C_j, R)}{P(C_j)P(R)}\right) \tag{17}$$

While delving into the above equation, for random variables C_j, R, $P(C_j, R)$ present joint probability distributions. $P(C_j), P(R)$ are marginal probability functions.

d) *Minimum Redundancy Maximum Relevance (mRMR):*

A multi-variate feature ranking algorithm based on maximizing the feature's relevance to the target variable while minimizing redundancy. The mRMR criterion can be defined as:

$$mRMR(h_j) = \max_{h_j \in D}\left[I(h_j; c) - \frac{1}{|S|}\sum_{h_l \in S} I(h_j; h_l)\right] \tag{18}$$

D represents a set of all candidate features for the target variable c. $I(h_j; c)$ and $I(h_j; h_l)$ represents mutual information between feature/target and feature/feature, respectively.

Table 4 shows the results of applying the aforementioned FS methods to three datasets (e.g., CWRU PU and DDS). We have gained valuable insights into the most important features for fault classification of rolling bearings. Each FS technique ranked these features in a different order, demonstrating unique strengths in capturing different aspects of data. For the CWRU dataset, the LS method highlights Log Energy Entropy '*F31*' and Kurtosis' *F17*', suggesting it is able to capture the distribution of energy and extreme deviations present in features due to the presence of faults. On the other hand, the ReliefF method ranked autoregressive model features (e.g., '*F9-F11*') in higher order, indicating the importance of intricate cyclic patterns and their fluctuation in fault detection. Meanwhile, the mutInfFS method focused on statistical features like '*F14*' (Median) and '*F12*' (Maximum), emphasizing the importance of how the data is spread and its peak values, whereas mRMR prioritized features like '*F31*' and *'F8-F11'*. Overall, when we analyzed the CWRU dataset, it was determined that entropy and autoregressive-based features are the most vital features that possess greater discriminatory power.

Similar trends were observed in both PU and DDS datasets as well. LS method again highlights entropy and statistical features in the PU dataset, that is, Renyi Entropy '*F33*' and kurtosis '*F17*' and Skewness' *F18*'. However, the ReliefF method took an alternative route as it highlights rapid changes in the vibration signal's amplitude via Second Difference '*F2*' and Normalized Second Difference *'F4'* as important features in PU and DDS datasets. Reinforcing the rate of

change significance of these features for fault detection in the PU and DDS context. Overall analysis suggested that features like *F31, F30, F33, F32, F1, F2, F3, F4, F5, F8* to *F11, F17, F18, F14, F12, F13* were identified important at regular intervals. However, features like *'F16'* and *'F15'* consistently rank lower. Therefore, we discarded those features for further study. This analysis also underlines the need for diverse sets of features due to the complicated conditions under which the rolling bearings operated in CWRU PU and DDS setup, including varying loads, speeds and other environmental factors like noise.

Table 4. Ranked features with four univariate and multi-variate FS techniques

Ranked features after implementing MVMD & HHT on CWRU, PU, and DDS datasets	
Feature ranking of CWRU datasets	
Laplacian Score	*'F31', 'F17', 'F13', 'F12', 'F33', 'F18', 'F30', 'F29', 'F22', 'F9', 'F8', 'F10', 'F19', 'F6', 'F23', 'F2', 'F4', 'F21', 'F16', 'F15', 'F1', 'F3', 'F5', 'F20', 'F11', 'F14', 'F27', 'F25', 'F24', 'F26', 'F28', 'F32', 'F7'*
ReliefF	*'F9', 'F11', 'F8', 'F14', 'F10', 'F22', 'F31', 'F27', 'F20', 'F23', 'F30', 'F25', 'F1', 'F3', 'F5', 'F33', 'F6', 'F26', 'F28', 'F2', 'F4', 'F24', 'F32', 'F17', 'F13', 'F18', 'F12', 'F29', 'F7', 'F19', 'F21', 'F16', 'F15'*
mutInfFS	*'F14', 'F12', 'F18', 'F8', 'F23', 'F20', 'F24', 'F25', 'F22', 'F11', 'F9', 'F26', 'F28', 'F13', 'F1', 'F5', 'F3', 'F6', 'F2', 'F4', 'F10', 'F30', 'F17', 'F32', 'F33', 'F27', 'F31', 'F29', 'F19', 'F7', 'F21', 'F16', 'F15'*
mRMR	*'F31', 'F8', 'F9', 'F10', 'F18', 'F33', 'F23', 'F12', 'F11', 'F30', 'F6', 'F17', 'F5', 'F22', 'F13', 'F1', 'F3', 'F2', 'F4', 'F20', 'F29', 'F27', 'F25', 'F24', 'F14', 'F26', 'F28', 'F32', 'F19', 'F7', 'F21', 'F16', 'F15'*
Feature ranking of PU datasets	
Laplacian Score	*'F33', 'F17', 'F13', 'F18', 'F22', 'F12', 'F33', 'F30', 'F29', 'F9', 'F10', 'F8', 'F20', 'F11', 'F14', 'F2', 'F4', 'F24', 'F25', 'F27', 'F6', 'F1', 'F3', 'F5', 'F28', 'F26', 'F23', 'F32', 'F7', 'F19', 'F21', 'F16', 'F15'*
ReliefF	*'F2', 'F4', 'F1', 'F3', 'F5', 'F7', 'F29', 'F31', 'F14', 'F6', 'F10', 'F33', 'F12', 'F9', 'F26', 'F30', 'F27', 'F18', 'F8', 'F25', 'F23', 'F13', 'F11', 'F17', 'F32', 'F28', 'F24', 'F22', 'F20', 'F21', 'F19', 'F16', 'F15'*
mutInfFS	*'F30', 'F18', 'F17', 'F32', 'F33', 'F1', 'F5', 'F3', 'F14', 'F31', 'F2', 'F4', 'F8', 'F9', 'F12', 'F13', 'F10', 'F22', 'F6', 'F25', 'F27', 'F24', 'F20', 'F11', 'F23', 'F28', 'F29', 'F26', 'F7', 'F21', 'F19', 'F16', 'F15'*
mRMR	*'F31', 'F32', 'F8', 'F2', 'F30', 'F10', 'F33', 'F20', 'F12', 'F18', 'F9', 'F13', 'F11', 'F4', 'F22', 'F14', 'F17', 'F28', 'F6', 'F1', 'F15', 'F3', 'F25', 'F27', 'F24', 'F23', 'F26', 'F7', 'F29', 'F21', 'F19', 'F16', 'F15'*
Feature ranking of DDS datasets	
Laplacian Score	*'F31', 'F17', 'F29', 'F13', 'F12', 'F18', 'F33', 'F22', 'F30', 'F9', 'F10', 'F8', 'F11', 'F23', 'F20', 'F14', 'F6', 'F2', 'F4', 'F27', 'F25', 'F32', 'F28', 'F24', 'F1', 'F3', 'F5', 'F26', 'F7', 'F21', 'F19', 'F16', 'F15'*
ReliefF	*'F2', 'F4', 'F5', 'F1', 'F3', 'F11', 'F6', 'F26', 'F10', 'F29', 'F20', 'F23', 'F25', 'F28', 'F24', 'F22', 'F13', 'F27', 'F14', 'F7', 'F12', 'F9', 'F32', 'F17', 'F30', 'F33', 'F8', 'F31', 'F18', 'F21', 'F19', 'F16', 'F15'*
mutInfFS	*'F1', 'F5', 'F3', 'F2', 'F4', 'F13', 'F23', 'F20', 'F8', 'F6', 'F30', 'F9', 'F18', 'F31', 'F17', 'F32', 'F33', 'F22', 'F27', 'F12', 'F28', 'F24', 'F26', 'F10', 'F14', 'F25', 'F7', 'F11', 'F29', 'F21', 'F19', 'F16', 'F15'*
mRMR	*'F31', 'F1', 'F5', 'F11', 'F20', 'F3', 'F2', 'F30', 'F10', 'F4', 'F13', 'F23', 'F24', 'F6', 'F32', 'F33', 'F27', 'F8', 'F28', 'F22', 'F26', 'F14', 'F9', 'F17', 'F7', 'F25', 'F12', 'F21', 'F19', 'F16', 'F15', 'F18', 'F29'*

3.5 Feature Transformation

Power transformer with the Yeo-Johnson method is a feature transformation technique designed to address skewness in real-world datasets, such as the CWRU, PU, and DDS datasets, where the distribution of values is often asymmetric. After implementing 4 previously mentioned FS techniques, we proceeded with the extended Yeo-Johnson method for stabilizing variance and reducing skewness by transforming features into a symmetrical distribution, which is beneficial for the performance of ML algorithms [61]. This transformation helps to stabilizes variance and reduce impact of extreme values such a outliers, this helps to ML models which are sensitive to feature distribution. Apart from that, it is more versatile than its counterpart, the Box-Cox transform, due to its ability to handle both positive and negative values [61]. The Yeo-Johnson transformation is defined for each positive and negative element g in a data matrix and a transformation parameter t . The transformation can be expressed as follows:

For positive values g^3 0,

$$T(g;t)=\begin{cases}\dfrac{(g+1)^t-1}{t} & \text{if} \quad t \neq 0\\ \log(g+1) & \text{if} \quad t=0\end{cases} \tag{19}$$

For negative values $g<0$,

$$T(g;t)=\begin{cases}\dfrac{-(|g|+1)^{2-t}+1}{2-t} & \text{if} \quad t \neq 2\\ -\log(|g|+1) & \text{if} \quad t=2\end{cases} \tag{20}$$

Redundancy and repetition are common in real-time data, so an extended Yeo-Johnson transformation eliminates duplicate features to decrease overfitting and increase computation complexity without adding new information to the model. Table 5 summarizes the feature transformation approach, which leads to robust learning with swift training by providing related and non-redundant multi-domain features with symmetrical distribution, as shown in Figure 5. In supplementary file, Figure B showcase the performance results of machine learning models with and without the Yeo-Johnson method for ReliefF features of PU datasets.

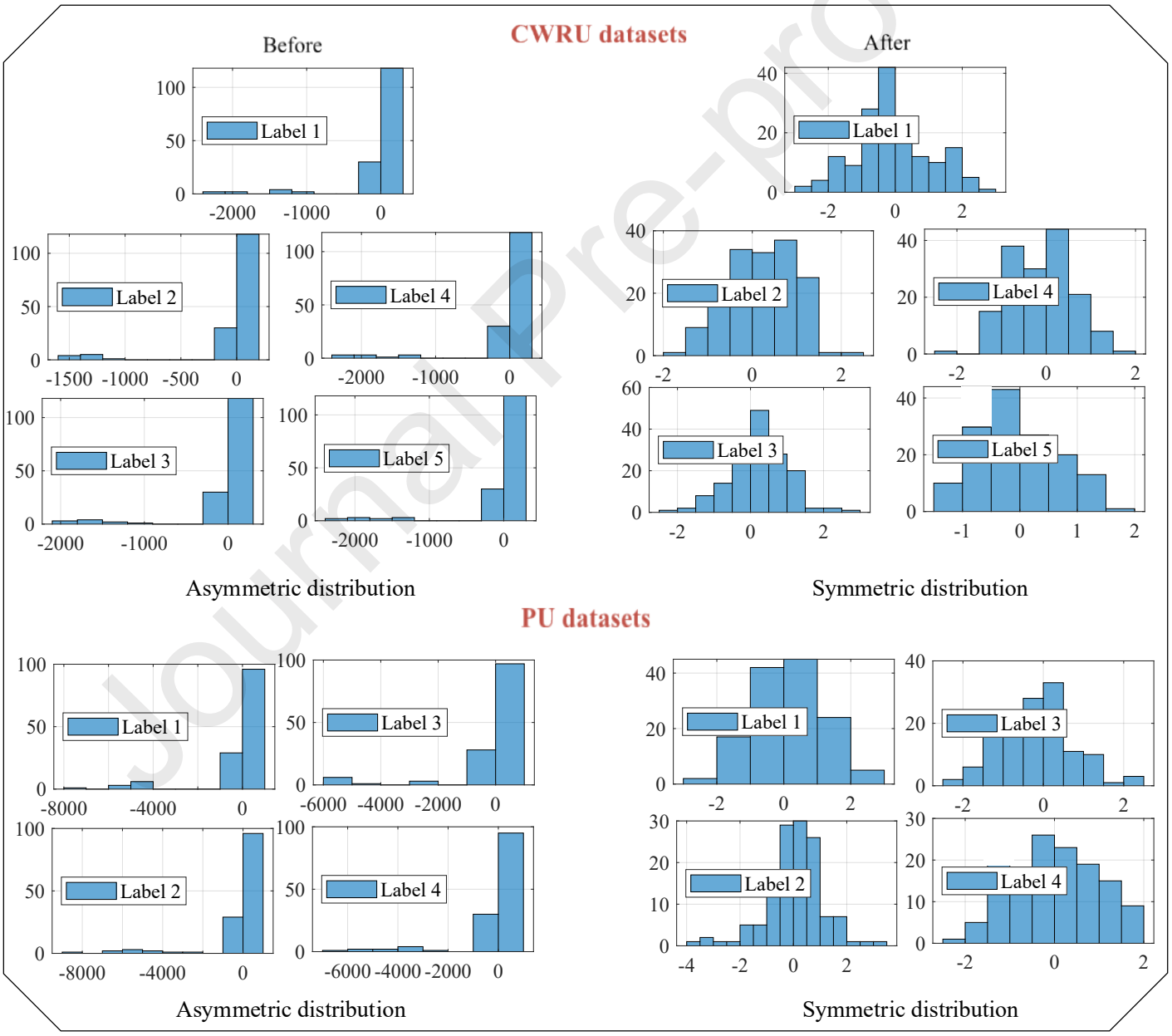


Figure 5: Symmetrical distribution (Histogram) of data after the Extended Yeo-Johnson Transformation

Table 5. Extended Yeo-Johnson Transformation

Algorithm 1: Extended Yeo-Johnson Transformation for multi-domain features

Input: Data matrix G, size $n \times p$.

Output: Transformed data matrix $T(g,t)$.

1: Eliminate Duplicate Features:
- Remove duplicate features.

2: Standardize Data Matrix:
-Standardize data matrix G to have zero mean and unit variance as $G \leftarrow \frac{G-m}{s}$.
- m is the mean and s is the standard deviation of the feature.

3: **Estimate Optimal t:**
- Estimate the optimal t using maximum likelihood estimation (MLE):
- $t = \arg\max_t \sum_{i=1}^{n} \log L(T(g_i;t))$, where L is the likelihood function.

4: Apply transformation for $g \geq 0$:
- For positive values, apply Equation **19**.

5: Apply transformation for $g < 0$:
- For negative values, apply Equation **20**.

6: Output Transformed Data:
- Output the transformed dataset $T(G;\hat{t})$.
- Each feature transformed with the estimated optimal t.

7: **End.**

Table 6: ML algorithm search range for hyperparameters (for details, see reference [64]).

Machine learning algorithm search range for hyperparameters						
Classifier	**Optimizable Hyperparameters**					
Ensemble	***Hyperparameters search space for the Ensemble algorithm***					
	Ensemble method	Maximum No. of splits	Number of Learners	Split Criterion	Number of predictors to sample	Learning rate
	AdaBoost, RUSBoost, Bag	[1, max (2, *n*-1)], Number of observations *n*	[10,500]	Deviance, Gdi, Twoing	'all'	[0.001,1]
SVM	***Hyperparameters search space for the SVM algorithm***					
	Box constraint	Multi-class coding	Kernel scale	Kernel function		Standardize data
	[0.001,1000]	One-vs-One, One-vs-All	[0.001,1000]	Gaussian, Linear, Quadratic, Cubic		Yes, No
KNN	***Hyperparameters search space for the KNN algorithm***					
	Number of Neighbors	Distance weight	Exponent	Distance kernel		Standardize data
	[1, max(2, *n*-1)]	Equal, inverse, square-inverse	[0.5, 3]	seuclidean, City block, Chebyshev, Minkowski, Mahalanobis, Cosine, Correlation, Spearman, Hamming, Jaccard.		Yes, No
ANN	***Hyperparameters search space for the NN algorithm***					
	1st, 2nd & 3rd Layers Sizes:	Layer Biases Initializer	Activation	Layer Weights Initializer	Regularization strength (lambda)	Standardize data
	[1-300]	zeros, ones	ReLU, Tanh, Sigmoid, None	glorot, he	[0.00001/*n*,100000/*n*]	Yes, No
Decision Trees	***Hyperparameters search space for the Decision Tree algorithm***					
	Min. leaf size	Maximum No. of splits	No. Variables to Sample	Split criterion		
	[1, max (2, *n*/2)]	[1, max (2, n-1)]	[1,max(2,No. of Predictors)]	Gini's diversity index, Twoing rule, and max. Deviance reduction.		
Naïve Bayes	***Hyperparameters search space for the Naïve Bayes algorithm***					
	Width	Distribution names		Kernel type		Standardize data
	NaN	Gaussian, Kernel		Gaussian, Box, Epanechnikov, and Triangle		Yes, No

3.6 Classifier

In this section, 6 sophisticated ML algorithms are examined: Naïve Bayes (NB), Decision Trees, Neural Networks (NN), K-Nearest Neighbors (KNN), ensemble methods (AdaBoost, RUSBoost, Bagging), and Support Vector Machine (SVM) for classifying faults by analyzing complex patterns in multi-domain features. As an example, SVM is considered a mature theory for fault classification in rolling bearings due to its intelligent hyperplane that handles high-dimensional vibration data from multiple sensors. Meanwhile, ensemble methods combine numerous classifiers to mitigate the weaknesses of each classifier, where KNN is an intuitive and non-parametric classifier with easier implementation [62].

These ML algorithms rely on hyperparameter optimization to fine-tune model parameters to achieve optimal performance, which helps to enhance accuracy, generalization, and reduce overfitting, collectively vital for reliable fault classification. The methods for optimizing these hyperparameters can be classified into synchronous and asynchronous models [63]. In this study, we have explored two synchronous models (i.e., grid and random search) and two asynchronous models (i.e., Bayesian optimization (BO) with parallel computing and the Asynchronous Successive Halving Algorithm (ASHA). Table 6 presents the hyperparameters that need to be optimized for the aforementioned 6 ML algorithms with their respective search range [64].

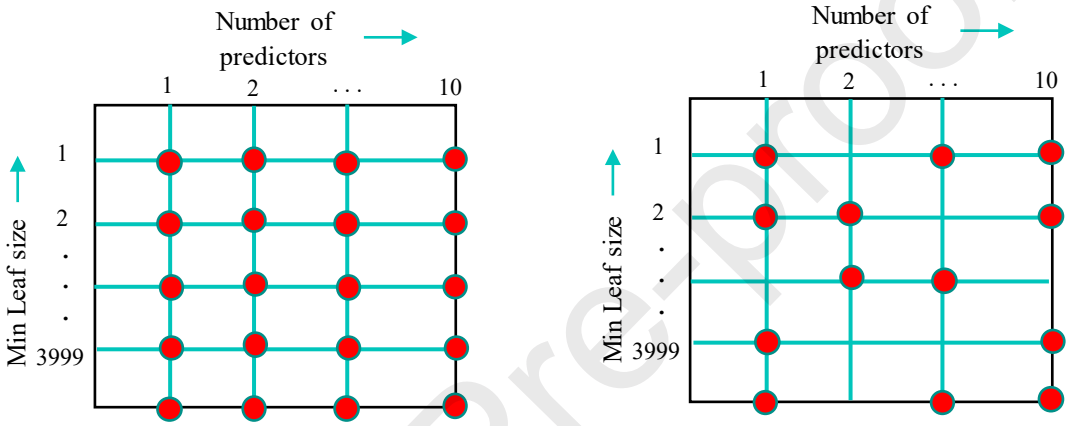


Figure 6: (a) Tree algorithm hyperparameter optimization via grid search (GS) (b) Tree algorithm hyperparameter optimization via random search (RS)

3.7.1 Optimizers

a) ***Random Search and Grid Search Optimizer***

The aim of RS is to explore the hyperparameter search space by selecting values in a random manner from a specified distribution. On the other hand, GS evaluates all possible combinations in a rigorous manner from a pre-defined grid [63]. A 2-D search space is illustrated in Figure 6 (a) and (b) for CWRU datasets with 4000 samples, where GS and RS attempt to optimize two Ensemble hyperparameters: 1) the number of predictors to sample and 2) minimum leaf size. While RS allows efficient search space exploration by randomly determining the best hyperparameter configuration using:

$$x^* = \arg\min_{x_i \in X} f(x_i) \tag{21}$$

In RS, X represents possible hyperparameter values, where $f(x_i)$ is the objective function evaluates the performance of each random combination x_i. In contrast, GS evaluates all possible combinations of 10 predictors against 3999 leaf sizes. As a result, due to the computational burden, GS cannot be implemented on large samples with a high-dimensional search space. Moreover, despite using the same number of trials, RS with parallel settings can outperform GS in a large search space by distributing the evaluation of different hyperparameter combinations across multiple processors [65].

b) ***Bayesian Optimization (BO)***

Optimizers like RS and GS scan the entire space without taking into account previous findings, which often leads to inefficiency because the optimal answer is usually found in a small area. On the other hand, BO uses previous findings to build a surrogate model of the objective function, allowing for a more narrowed and efficient search. For this purpose, BO uses a Gaussian Process (GP) to model the objective function. A GP is characterized through a mean $m(\bar{x})$ and a covariance $k(\bar{x},\bar{X})$ function, where $\bar{x}$ denotes the input vector. The GP prior is defined as [66]:

$$f(\bar{x}) \sim GP(m(\bar{x}), k(\bar{x},\bar{X})) \tag{22}$$

When the observed data points $D = \{(\bar{x}_i, \bar{y}_i)\}_{i=1}^{n}$ are provided, the mean and variance are updated in the posterior

distribution of the GP to include the new observations. The posterior mean $\mu(\bar{x})$ and variance $s^2(\bar{x})$ are updated in GP based on observed points as:

$$\mu(\bar{x}) = k(\bar{x},\bar{X})(k(\bar{X},\bar{X}) + s_n^2 I)^{-1}\bar{y} \tag{23}$$

$$s^2(\bar{x}) = k(\bar{x},\bar{x}) - k(\bar{x},\bar{X})(k(\bar{X},\bar{X}) + s_n^2 I)^{-1}k(\bar{X},\bar{x}) \tag{24}$$

In these equations, the matrix $\bar{X}$ contains the observed input vectors, the vector $\bar{y}$ contains the observed values, and the variability s_n^2 is the noise variance. As GP updates based on observed points, the next data point $\bar{x}_{n+1}$ is selected by maximizing the acquisition function $a(\bar{x})$:

$$\bar{x}_{n+1} = \arg\max_{\bar{x}} a(\bar{x}; D) \tag{25}$$

This study uses the expected-improvement-per-second-plus (*EIps+*) function as an acquisition function that aims to find the next point by optimizing the balance between exploring new areas and exploiting known best-performing areas with faster convergence [66].

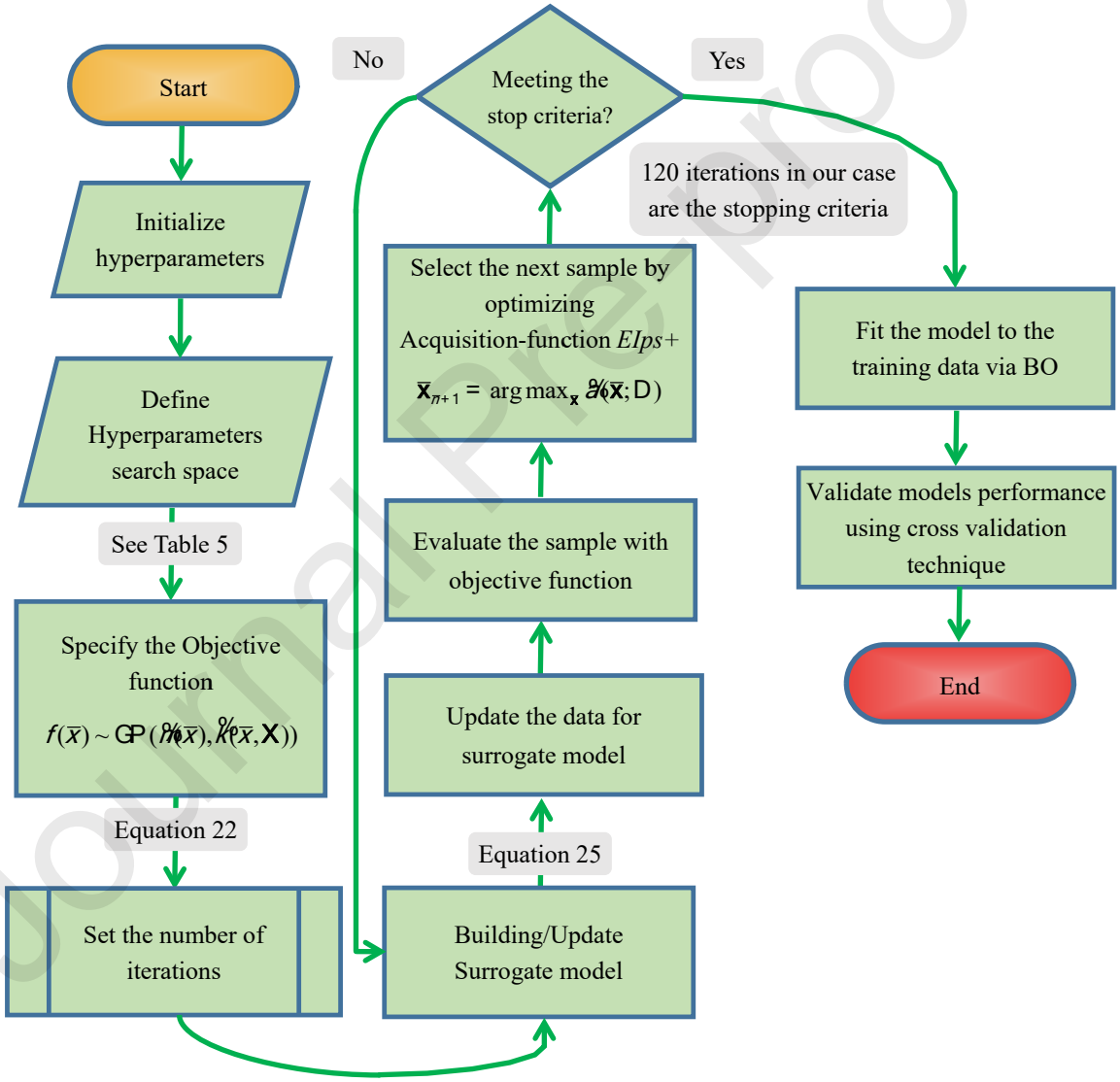


Figure 7: Bayesian optimization flowchart.

The BO flowchart is illustrated in Figure 7. By iterative building and updating the surrogate model and maximizing an acquisition function for 120 iterations, BO efficiently focuses on areas of the search space that are likely to contain the global optimum. This way BO identifies better model settings with fewer evaluations compared to RS or GS, making it suitable for optimizing functions on large samples with a high-dimensional search space.

Table 7. Asynchronous Successive Halving (ASHA)

Algorithm 2: Asynchronous Successive Halving (ASHA)

Input: r , max-resource R, h, min-early-stopping rate s .

Output: The best configuration from large-scale hyperparameters.

1: Initialize:

- Input Parameters:
  - r : Minimum resource allocated to any configuration. R: Maximum resource allocated to any configuration. h : Reduction factor; typically, $h = 2$. s : Minimum early-stopping rate.
- Setup:
  - Compute the total number of rungs: $s_{max} = \lfloor \log_h(R/r) \rfloor$.

2: ASHA Function:

- For each available worker:
  - Retrieve a job (hyperparameter configuration q and rung k) using the **get job().**
  - Evaluate the configuration q for rh^{s+k} resources and return the validation loss.

3: **Update Completed Jobs:**

- For each Completed Job (q, k) with Loss l :
  - Update the loss for configuration q in the rung k. Repeat the Loop.

4: End and Loop: Nonstop cycle through ASHA function until the chosen configuration is found.

Job Retrieval (get job ()):

- Check for Promotable Configurations:
  - For $k = s_{max} - s - 1, \ldots, 1, 0$:
  - Select top configurations in rung k: $candidates = top_k(rung\ k, \frac{|rung\ k|}{h})$.
  - Identify promotable configuration $promotable = \{t \in candidates : t \text{ not promoted}\}$
  - If $|promotable| > 0$: Then Return: First promotable configuration and the next rung: $(promotable[0], k+1)$.
- If no promotable configurations are found:
  - Draw a new random configuration q.
  - Return: New configuration and base rung $(q, 0)$.

End function.

c) ***Asynchronous Successive Halving Algorithm (ASHA) optimizer***

BO with parallel settings evaluates multiple acquisition functions at once, which reduces overall optimization time, but it does not scale well on large-scale hyperparameter configurations due to the continuous demand of maintaining and updating the surrogate model [63]. ASHA tackles this issue by exploiting parallelism and earlier stopping, outperforming BO, RS, and GS with swift and effective results. ASHA optimizes the allocation of computing resources by promoting the best-performing configurations to the next iteration (upper rungs) with additional resources, whereas poor configurations are discontinued. This procedure ensures that computational resources are directed toward the most promising configurations, resulting in a more efficient and optimized process. The ASHA optimizer is presented in Table 7.

There are several notable parameters that regulate this optimizer, which include the minimum resource r , maximum resource R, reduction factor h, and rate of early stopping s. The asynchronous nature of the ASHA optimizer allows top configurations to progress to the upper rungs for assessment when they complete their existing rung instead of waiting for all configurations to finish. In this way, each worker has little idle time with optimized computational resources. Mathematically, the configurations number n_i and resources r_i for each rung i are calculated as:

$$n_i = \left\lfloor \frac{n}{h^i} \right\rfloor \tag{26}$$

$$r_i = rh^i \tag{27}$$

These equations ensure that few configurations are evaluated with more resources in each upper rung. Also, it is possible to estimate the time required by each configuration from training to completion by aggregating the time spent on all rungs,

which is represented as follows:

$$\left(\sum_{i=s}^{\log_{\eta}(R)} \eta^{i}\right) \times \text{time}(R) \leq 2 \times \text{time}(R) \tag{28}$$

In the above equation, $\text{time}(R)$ is the time needed to train a single configuration with maximum resources. In that case, the ASHA optimizer will take no more than $2 \times \text{time}(R)$ to complete the training for all hyperparameters. Thus, by combining effective utilization of idle time with the asynchronous promotion of configurations, ASHA can manage stragglers and dropped jobs throughout the entire process, allowing it to scale well in large-scale distributed settings.

Table 8. Optimized hyperparameters for Proposed Feature Selection methods for CWRU Bearing Dataset

Summary of optimized hyperparameters for Proposed Feature Selection Methods using BO, ASHA, and RS optimizers							
Feature Selection Method	**Optimization Method**	**Machine Learning Classifier (Algorithm)**	**Hyperparameters**				
LS Method			*Optimized Hyperparameters*				
	Optimizers	**Best Classifier**	Number of Neighbors	Distance weight	Exponent	Distance metric	Standardize data
	Bayesian	KNN	3	Equal	-	Cosine	No
			Box constraint	Multi-class coding	Kernel scale	Kernel function	Standardize data
	ASHA	SVM	575.71	onevsone	18.314	Gaussian	No
	Random Search	SVM	482.69	onevsone	47.063	Gaussian	Yes
ReliefF Method			*Optimized Hyperparameters*				
	Optimizers	**Best Classifier**	Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization (lambda)
	Bayesian	ANN	Two, [30, 12]	Zeros	Tanh	Glorot	1.7982e-05
	ASHA	ANN	One, [23]	Ones	Sigmoid	Glorot	1.1876e-05
	Random Search	ANN	One, [20]	Ones	ReLU	Glorot	5.8357e-07
mutInfFS Method			*Optimized Hyperparameters*				
	Optimizers	**Best Classifier**	Box constraint	Multi-class coding	Kernel scale	Kernel function	Standardize data
	Bayesian	SVM	986.92	onevsall	13.804	Gaussian	No
	ASHA	SVM	6.66	onevsall	12.324	Gaussian	No
			Number of Neighbors	Distance weight	Exponent	Distance metric	Standardize data
	Random Search	KNN	3	Inverse	-	seuclidean	No
mRMR Method			*Optimized Hyperparameters*				
	Optimizers	**Best Classifier**	Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization-strength(lambda)
	Bayesian	ANN	One, [98]	ones	tanh	he	2.3495e-06
			Number of Neighbors	Distance weight	Exponent	Distance metric	Standardize data
	ASHA	KNN	4	squaredinverse	-	Cosine	No
			Box constraint	Multi-class coding	Kernel scale	Kernel function	Standardize data
	Random Search	SVM	0.053603	onevsall	-	Polynomial	No

4. Research Data and Analytical Discussion

4.1 Overview

This section presents the fault classification results obtained after implementing four proposed FS methods on the available datasets. The datasets were split randomly into two parts: 80% for training and 20% for testing purposes. During optimization, BO, ASHA, and RS utilized the training data to train and cross-validate classification errors (validation loss) for each classifier, ensuring that the selected hyperparameters were optimized for performance. The validation phase refined these hyperparameters, while the test data was reserved for final evaluation to assess the models' generalization ability. This separation of training/validation and testing phases is critical for obtaining unbiased performance metrics. Apart from the accuracy metric, this study also focused on four key performance metrics: accuracy, precision, recall, F1-score, and kappa. These metrics provide a comprehensive evaluation of the classifier's performance, ensuring a robust assessment of their effectiveness. The next sub-section provides a detailed analysis of the best classifier performance for each FS method across the CWRU, PU, and DDS datasets.

4.2 Case #1: CWRU Bearing Dataset.

The fault diagnosis for rolling bearings consists of two parts: detection and classification. In the detection stage, segregating healthy faults from faulty ones requires less information for this purpose. However, for a robust classification stage, the advanced ML algorithm requires sophisticated signal processing, feature extraction, and selection methods to obtain more sensitive and non-redundant information. This is because the complex vibration signals of rolling bearings operate under variable conditions and contain intricate patterns that need to be identified and classified into various types of faults, such as inner race faults, outer race faults, and ball faults. Based on the analysis in Section 3.4, we have selected 250 features for each sample to capture these intricate patterns. The data used for the FS methods in this section are from the CWRU testbed, ensuring a robust and reliable dataset. Moreover, the best classifiers selected by BO, AHSA, and RS optimizers with their tuned hyperparameters are showcased in Table 8. This comprehensive approach highlights the blend of meticulous data handling and cutting-edge optimization techniques essential for effective fault diagnosis in rolling bearings.

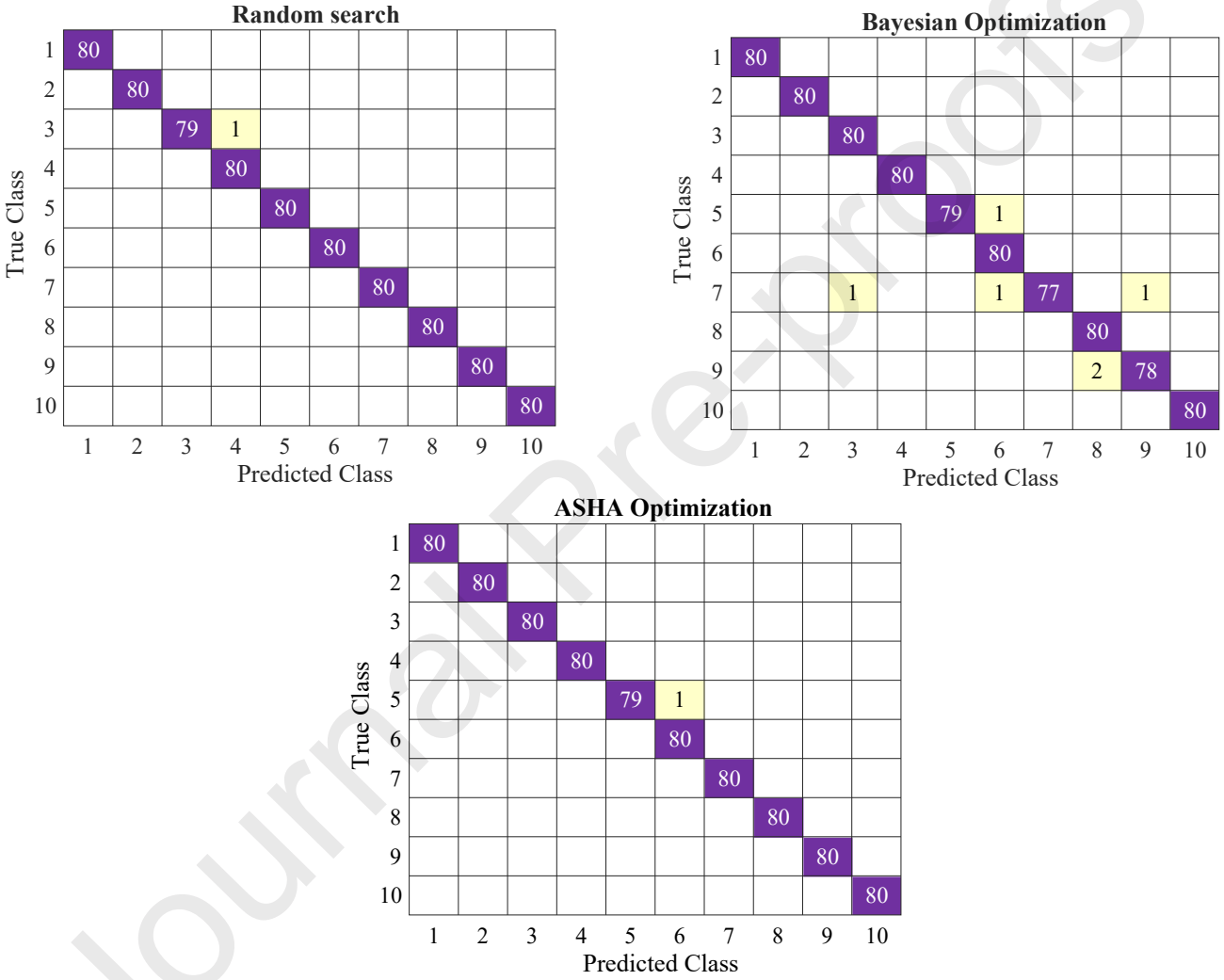


Figure 8: Confusion matrices (Test data) for LS Method comparing RS, BO, and ASHA Optimization for classification accuracy

Table 9. LS Method Training and Test Results with BO, ASHA, and RS Optimizers

LS Method with Best Classifier Training and Test Results with BO, ASHA, and RS optimizers for CWRU datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
LS Method	Bayesian	KNN	99.66%	99.25%	0.99	0.99	0.99	0.99	3.2147	0.0084256
	ASHA	SVM	100%	99. 88%	1	1	1	1	6.9034	0.0028125
	Random Search	SVM	100%	99.875%	0.99877	0.99875	0.99875	0.99861	8.5723	0.003125

4.2.1 Analysis of LS Results

This section presents the results of the LS method to provide an insightful evaluation of the classifiers selected through BO, ASHA, and RS optimizers. Table 9 illustrates the performance metrics such as accuracy, precision, recall, F1 score,

kappa, training and validation time, and validation loss for each optimizer.

When dealing with complex models and large hyperparameter spaces, RS sequentially ran through different random combinations to find the best settings, which is very time-consuming. However, when combined with parallel settings, it can evaluate multiple combinations at once by leveraging the power of parallel processing. This significantly reduces the time needed for hyperparameter tuning and makes the whole process more efficient and scalable. Meanwhile, the objective is to minimize the validation loss across different learner types: Ensemble, KNN, NB, ANN, SVM, and Tree. The experiment ran for a total of 180 iterations, leveraging 24 active workers to evaluate multiple configurations in parallel. The total evaluation time was 1027.3289 seconds. As shown in Table 8, the RS method identified SVM as the best classifier. This setup achieved a perfect training accuracy of 100% and a test accuracy of 99.875%.

During the training and validation process, SVM creates optimal decision boundaries for the data generated by the LS method, which helps to classify various bearing faults in the CWRU testbed. By mapping the data points to a higher-dimensional space, the SVM identifies the optimal hyperplane that maximizes the margin between different fault categories, aiming to minimize classification errors. In doing so, the choice of hyperparameters, including box constraint, multi-class coding, kernel scale, function, and data standardization, considerably influences the SVM model's classification accuracy for bearing faults. In particular, the box constraint controls the penalty for misclassified data points in the decision boundary. A high box constraint value of 482.69 suggests that a strong penalty is added on misclassification errors that help to create precise decision boundaries between different classes. Meanwhile, one-vs-one multi-class coding creates $[\acute{n}(\acute{n}\text{-}1)/2]$ classifiers for $\acute{n}$ fault types, where each classifier creates a decision boundary in pairwise form among 10 fault types. This pairwise distinction is easier to implement while minimizing the risk of including outliers. However, adding multiple classes makes it computationally expensive, which is clearly outlined in the training and validation time for the RS-based SVM model in Table 9.

Unlike linear kernels, the Gaussian kernel can handle non-linear relationships in the data. It maps the data into a higher-dimensional space where a linear and smooth decision boundary can be applied, making it suitable for complex datasets. The kernel scale with 47.063 controls the spread of decision boundary, allowing the model to accurately capture intricate data patterns while performing well on test sets. This effectiveness is evident in Figure 8 confusion matrix, which shows only one misclassified fault from the test data after training. Apart from that, high values of precision, recall, F1 Score, and Kappa on test data also suggest that RS suggested model performed well.

Unlike RS, which evaluates random combinations sequentially or in parallel, BO builds a cheap surrogate model to approximate the objective function and uses this model to select the most promising hyperparameters for evaluation. This informed search reduces the number of evaluations needed while making the hyperparameter tuning process more efficient. In the BO experiment, the total evaluation time for all six ML algorithms is 8762.7027 seconds, which is higher than the RS evaluation setup. However, BO found a much simpler architecture for a selected classifier than RS. After 180 iterations, KNN was selected as the best classifier with the following hyperparameters: number of neighbors set to 3, equal distance weight, and cosine distance metric without data standardization. A small value of 3 neighbors, where each neighbor is weighted equally, ensures that the decision boundary is not overly complex and has a lower computational burden [67]. This is reflected in the lowest training and validation time of 3.2147 seconds in Table 9. The transition from distances to weights should follow some kernel functions, such as rectangular, triangular, cosine, Gauss, and inversion kernels. The use of a cosine kernel was particularly effective in handling the high-dimensional fault-bearing data, focusing on the direction of the vibration patterns, which is critical in fault detection. The effectiveness of these hyperparameters is evident in the high-performance metrics achieved by the KNN classifier in Table 9.

Designed to handle large datasets and high-dimensional search spaces, ASHA employs an iterative pruning process that dynamically reallocates computational resources to the most promising hyperparameter configurations. This approach prevents wasted computation on poor-performing models, which is crucial when dealing with extensive datasets. For the dataset with 3200 training samples, the number of iterations is set to $21\times(L+1)$, where L is the number of learner types. Given L=6, this results in 147 iterations. This non-user-defined iterative process ensures that the most promising configurations receive more computational resources in subsequent iterations, leading to optimal hyperparameter selection. Consequently, the optimized hyperparameters included a box constraint of 575.71, one-vs-one multi-class coding, a kernel scale of 18.314, and a Gaussian kernel function without standardizing the data.

In comparison to the RS SVM model, the ASHA SVM model achieves lower training and validation time with the lowest validation loss, showcasing its efficiency. Furthermore, compared to the BO experiment, ASHA's iterative pruning and dynamic resource reallocation resulted in a total evaluation time of 714.3297 seconds for all six ML algorithms. Therefore, considering the balance between computational cost and accuracy, the ASHA SVM model is superior in

performance for LS features, with a training accuracy of 100% and a test accuracy of 99.88%.

4.2.2 Analysis of ReliefF Results

ReliefF inputs were used in this section to select an optimized model via BO, AHSA, and RS based on its relevance to the target variable. Surprisingly, for the ReliefF method, all optimizers selected ANN as the best classifier from six ML algorithms. In this work, we consider various hyperparameters of an ANN, including the number of hidden layers (denoted as layers), the number of neurons in each layer (denoted as size), the initial biases and weights of the layers, the activation functions (ReLU, Tanh, sigmoid), and the regularization strength (lambda). While delving into RS performance, we have found that it performed much better than the BO and ASHA models. An optimal choice of ReLU ensures that the RS ANN model learns complex relations between features; it also possesses the ability to avoid vanishing gradients and ensure efficient computation. This is crucial when coupled with lower lambda, as it controls model complexity by penalizing large weights, thereby preventing overfitting and promoting better generalization to unseen data. This is clearly evident in Figure 9, as the RS ANN model performs well on unseen data.

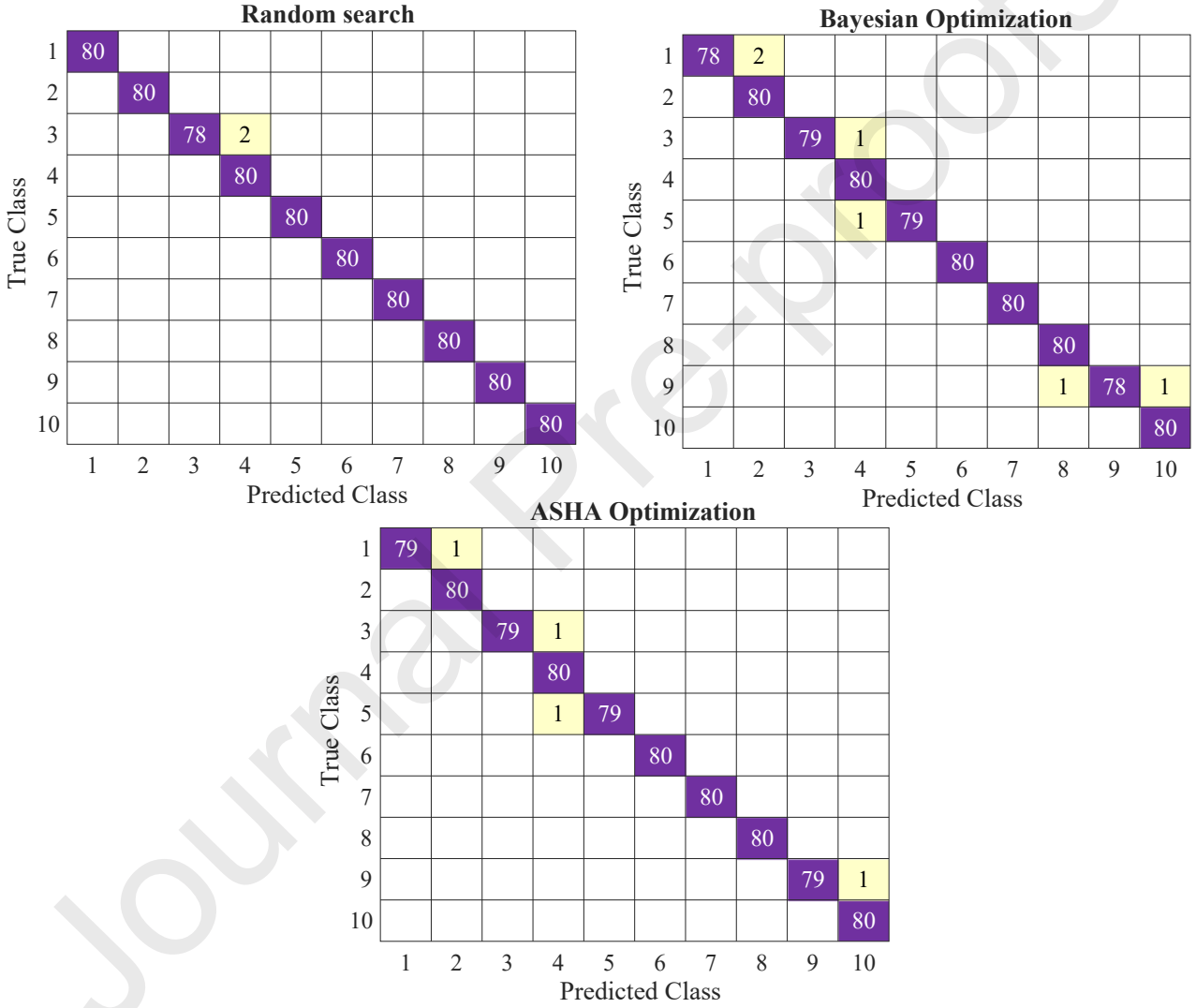


Figure 9: Confusion matrices (Test data) for ReliefF Method comparing RS, BO, and ASHA Optimization for classification accuracy

Table 10. ReliefF Method Training and Test Results with BO, ASHA, and RS Optimizers

ReliefF Method with Best Classifier Training and Test Results with BO, ASHA, and RS Optimizers for CWRU datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
ReliefF Method	Bayesian	ANN	100%	99.25%	0.99	0.99	0.99	0.99	22.1234	0.0046428
	ASHA	ANN	100%	99.50%	1	0.99	1	0.99	12.3222	0.0040625
	Random Search	ANN	100%	99.75%	0.99756	0.9975	0.9975	0.99722	4.9243	0.0021875

Proper initialization for biases can help in faster convergence and better model performance. Zero initialization can make the training process more stable, while one initialization can help the model learn faster. The 'Ones' initializer for biases

ensures neurons start with non-zero gradients, assisting in early learning, while the 'Glorot' initializer for weights keeps the training stable by preventing the gradient from becoming too small or too large. Together, they ensure faster training with stable learning. This is also evident in Table 10, where this configuration achieved the highest test accuracy (99.75%) with robust precision, recall, and F1 scores. Moreover, the training time was the shortest among the ANN models (4.9243 seconds), and the validation loss was the lowest (0.0021875), reflecting an efficient and effective optimization process.

Compared to the RS ANN model, the BO and ASHA models are more constrained due to the large values of lambda. Plus, selecting the Tanh and Sigmoid activation functions may lead to the vanishing gradient problem, which can slow down training and potentially hinder performance. This issue is also reflected in the training and validation times of the BO ANN (22.1234 seconds) and ASHA ANN (12.3222 seconds) models. In addition, the RS ANN model has the advantage of a simple structure with 20 neurons in a single hidden layer. The evaluation time for the six ML algorithms is shortest for RS (1213.1090 seconds) compared to BO (11284.8635 seconds) but longer than ASHA (854.5576 seconds). Therefore, considering both accuracy and computational efficiency, the ReliefF method with the RS ANN model performed the best overall.

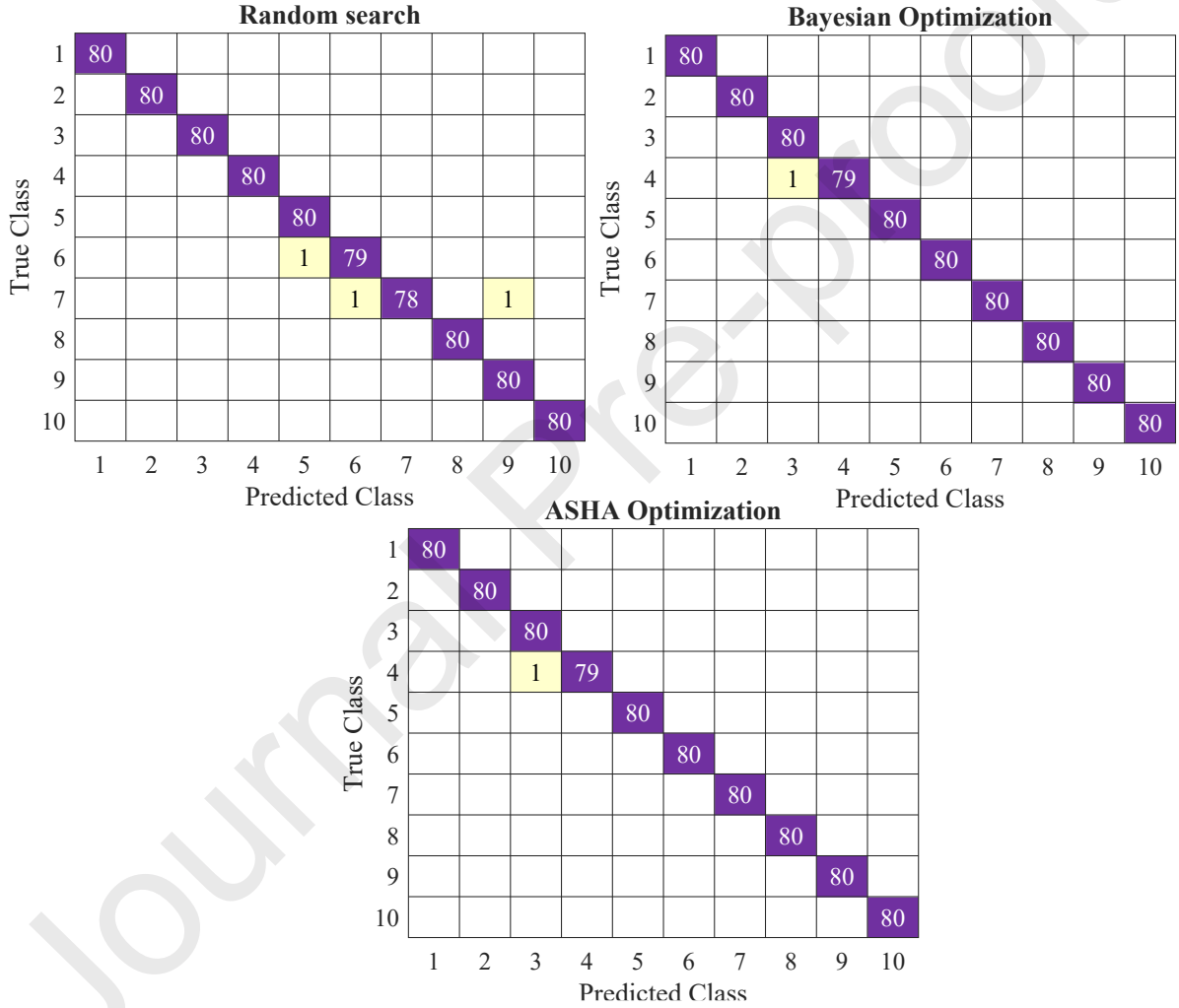


Figure 10: Confusion matrices(Test data) for mutInfF Method comparing RS, BO, and ASHA Optimization for classification accuracy

Table 11. mutInfFS Method Training and Test Results with BO, ASHA, and RS Optimizers

mutInfFS Method with Best Classifier Training and Test Results with BO, ASHA, and RS optimizers for CWRU datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
mutInfFS Method	Bayesian	SVM	100%	99.88%	1	1	1	1	9.7707	0.0027602
	ASHA	SVM	100%	99.88%	1	1	1	1	10.0123	0.0025
	Random Search	KNN	100%	99.625%	0.99628	0.99625	0.99624	0.99583	2.3418	0.005625

4.2.3 Analysis of mutInfFS Results

The mutInfFS method, focusing on the mutual information between features and the target, yielded impressive results

with the SVM classifier under Bayesian and ASHA optimization, as shown in Table 11. BO resulted in the test accuracy of 99.88% with SVM, using a box constraint of 986.92, one-vs-all multi-class coding, a kernel scale of 13.804, and a Gaussian kernel function without standardization. The training and validation time was 9.7707 seconds, and the validation loss was 0.0027602. ASHA optimization matched this accuracy with somewhat different hyperparameters: a box constraint of 6.66, one-vs-all coding, and a kernel scale of 12.324, with a training time of 10.0123 seconds and a validation loss of 0.0025. In the one-vs-all approach for the CWRU dataset, SVM generates 10 models, each designed to identify one specific class against all others. The final prediction is made by selecting the class with the highest confidence score from these models. However, the ASHA model is less complex than the BO model. This is because the box constraint influences the balance between the tolerance for misclassification errors and the complexity of the SVM model. Since the ASHA SVM model with a box constraint of 6.66 is less complex and has the lowest validation loss, it is selected as the best model.

The total evaluation time for the six ML algorithms is the shortest for RS, taking only 1111.3154 seconds, compared to ASHA's 1275.48 seconds and BO's 8972.71 seconds. RS selects the simplest KNN model, using three neighbors, inverse distance weighting, and the seuclidean distance metric without standardization, with the lowest training and validation time of 2.3418 seconds. The standardized Euclidean (seuclidean) distance measures how far apart points are by taking into account the differences in feature variances. It standardizes each feature before calculating the distance, ensuring features with larger scales don't dominate the calculation. However, it has the highest validation loss of 0.005625 among all models. As shown in Figure 10, the RS KNN model misclassifies 3 unseen classes, performing not well as the other optimizers.

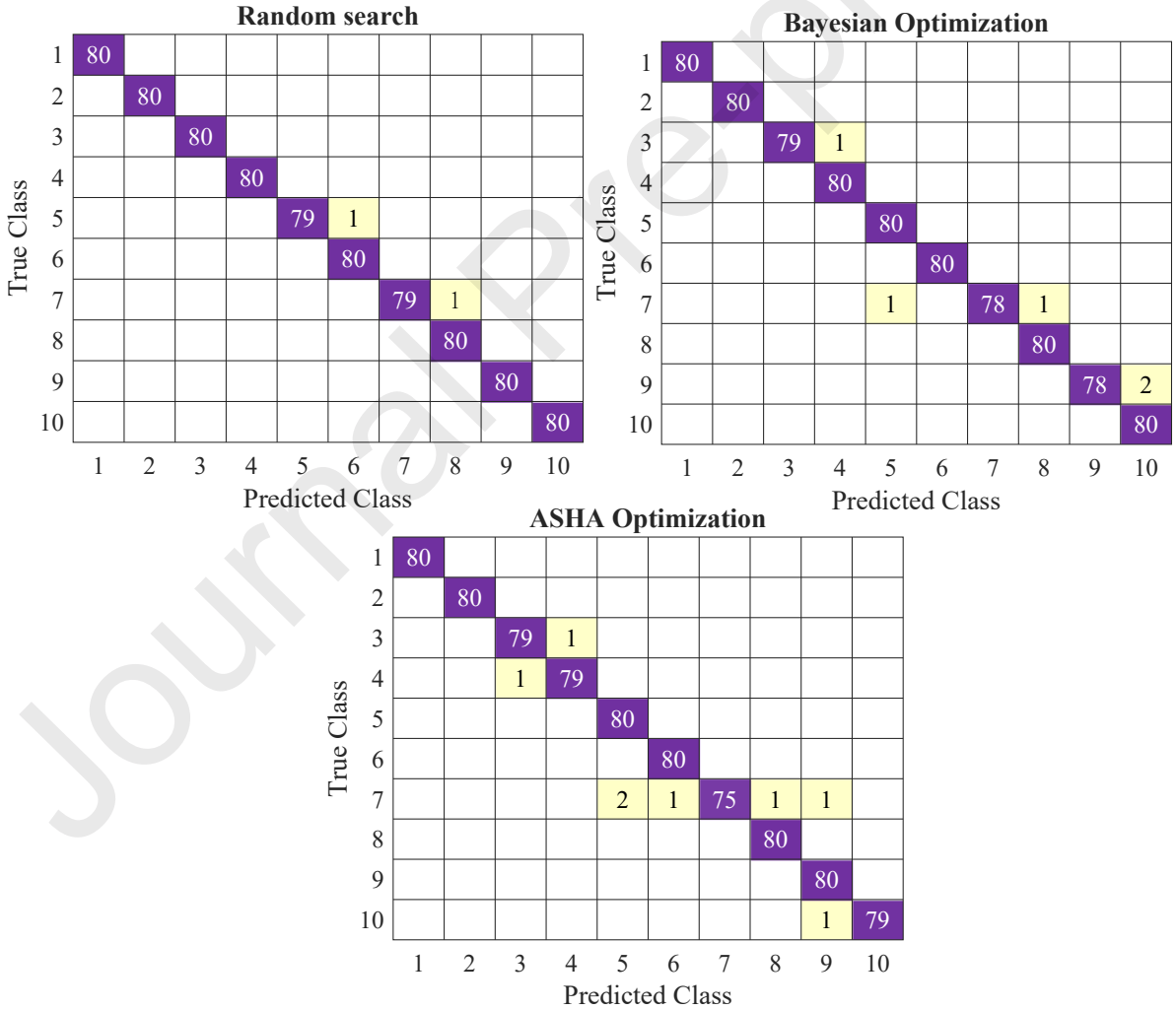


Figure 11: Confusion matrices (Test data) for mRMR Method comparing RS, BO, and ASHA Optimization for classification accuracy

4.2.4 Analysis of mRMR Results

The mRMR method, designed to balance feature relevance and redundancy, demonstrated high performance with BO ANN and RS SVM classifiers, as shown in Table 12. Similar to LS and mutInfFS results, KNN continuously shows the

lowest training and validation time (2.7524 seconds) with the highest validation loss of 0.0046875. Moreover, the ASHA KNN model misclassifies 8 unseen class points in confusion matrices, see Figure 11.

Table 12. mRMR Method Training and Test Results with BO, ASHA, and RS Optimizers

mRMR Method with Best Classifier Training and Test Results with BO, ASHA, and RS optimizers for CWRU datasets									
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values						
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
mRMR Method	Bayesian	ANN	100%	99.38%	0.99	0.99	0.99	0.99	17.0816	0.0041872
	ASHA	KNN	100%	99.00%	0.99	0.99	0.99	0.99	2.7524	0.0046875
	Random Search	SVM	100%	99.75%	0.99753	0.9975	0.9975	0.99722	7.7397	0.0028125

BO with ANN achieved 99.38% test accuracy, with a configuration of one layer of 98 hidden neurons, tanh activation, ones biases, He initializer, and a lambda of 2.3495e-06. Proper initialization with the He Initializer helps prevent gradients from becoming too small (vanishing) or too large (exploding), leading to more stable and efficient training. The He initializer works by setting the initial weights of a neural network based on the number of input neurons, and it pairs well with activation functions like tanh. He initializer and regularization parameter lambda (2.3495e-06) with 98 hidden neurons may improve the training stability and final accuracy but require more iterations to converge optimally, adding to the training time of 17.0816 seconds. Moreover, compared to ASHA 557.7281 seconds, BO took 7921.0021 seconds to evaluate 6 ML algorithms. In comparison, RS took 994.042 seconds and selected a less complex (more regularized) SVM model using a small box constraint of 0.053603. RS SVM model achieved balanced training in 7.7397 seconds and observed the lowest validation loss of 0.0028125. In summary, RS with SVM outperformed both BO and ASHA models.

Table 13. Comparison across Feature Selection Techniques

Comparison for best optimizers with selected ML algorithms in each Feature Selection Method for CWRU datasets								
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values					
			Test Precision	Test Recall	Test F1Score	Test Kappa	Training and Validation time	Validation Loss
LS Method	ASHA	SVM	1	1	1	1	6.9034	0.0028125
ReliefF Method	Random Search	ANN	0.99756	0.9975	0.9975	0.99722	4.9243	0.0021875
mutInfFS Method	ASHA	SVM	1	1	1	1	10.0123	0.0025000
mRMR Method	Random Search	SVM	0.99753	0.9975	0.9975	0.99722	7.7397	0.0028125

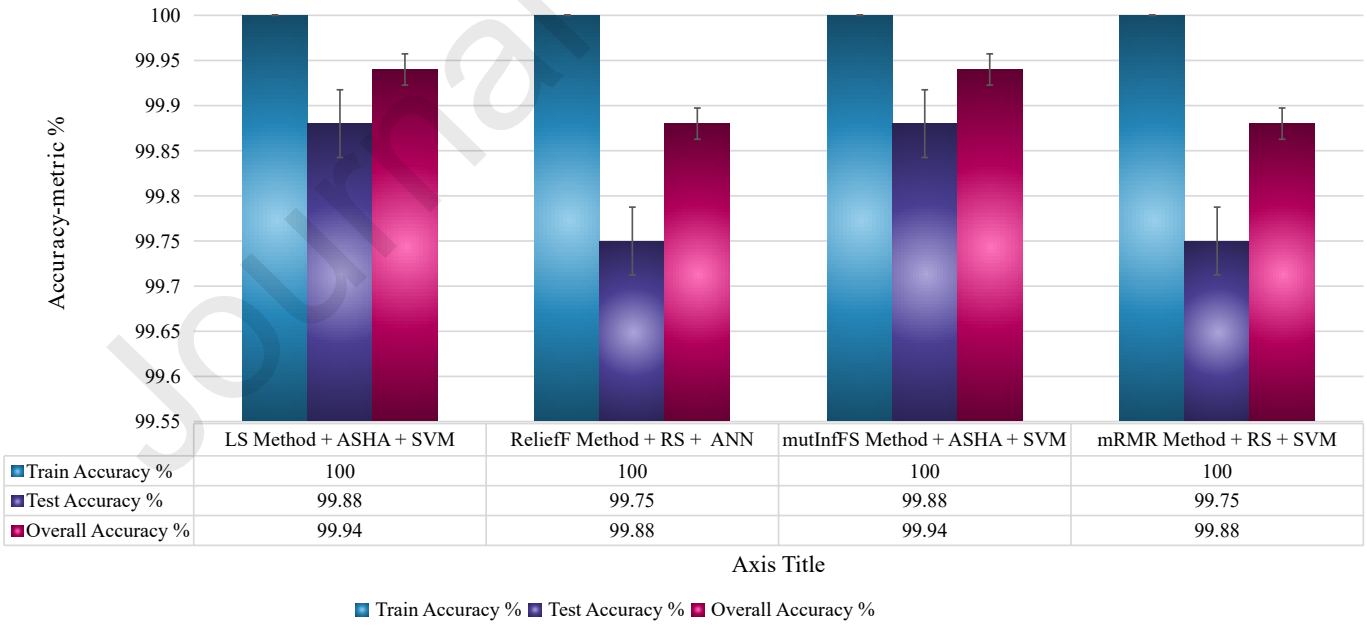


Figure 12: Bar chart (Test data) comparing the performance of best-performed FS methods and Optimizers for CWRU datasets

4.2.5 Optimal Combinations and Recommendations

Based on the findings presented in Table 13, this section explores the optimal combinations of optimizers and classifiers for each FS method using the CWRU datasets. The ASHA optimizer demonstrates superior performance when paired with SVM classifiers in both the LS and mutInfFS methods. 10-fold cross-validation on the selected candidate model, optimized hyperparameter selection (like box constraint), and robust feature selection help reduce overfitting and

generalization in the clean and balanced CWRU dataset. As illustrated in Figure 12, this combination achieved an overall accuracy of 99.94%, while maintaining reasonable training and validation time of 6.9034 and 10.0123 seconds, respectively. Moreover, ASHA with parallel settings took the shortest time to evaluate all 6 ML algorithms. In contrast, the RS optimizer gives an overall accuracy of 99.88% when paired with ANN and SVM classifiers in both the ReliefF and mRMR methods. However, in terms of computational resources, the classifiers selected by RS exhibited the shortest training and validation times.

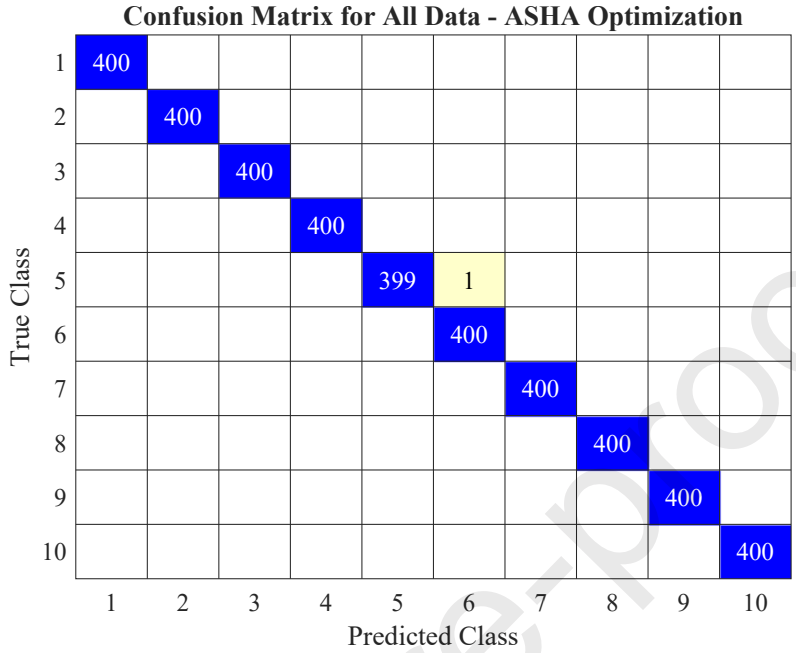


Figure 13: Confusion Matrix for LS Method with ASHA optimizer and SVM classifier (CWRU datasets)

LS method excels in preserving the small-scale patterns within the data, leading to high accuracy in complex datasets, as shown in Figure 13. Given the results, ASHA with LS method is recommended for fault classification of rolling bearings in CWRU datasets due to its overall accuracy and moderate computational burden. However, the ReliefF method with RS optimizer serves as a suitable alternative for situations where computational efficiency is paramount in real-time applications.

Table 14. Optimized hyperparameters for Proposed Feature Selection methods for PU Bearing Dataset

Summary of optimized hyperparameters for Proposed Feature Selection Methods using BO, ASHA, and RS optimizers							
Feature Selection Method	**Optimization Method**	**Machine Learning Classifier (Algorithm)**	**Hyperparameters**				
			Optimized Hyperparameters				
	Optimizers	**Best Classifier**	Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization (lambda)
LS Method	Bayesian	ANN	Three, [13,25,227]	Ones	ReLU	Glorot	0.00035175
	Random Search	ANN	Two, [22,15]	Zeros	Sigmoid	Glorot	4.8066e-07
			Box constraint	Multiclass coding	Kernel scale	Kernel function	Standardize data
	ASHA	SVM	5.2787	onevsall	10.047	Gaussian	No
			Optimized Hyperparameters				
	Optimizers	**Best Classifier**	Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization (lambda)
ReliefF Method	Bayesian	ANN	Three,[45,64,36]	Zeros	ReLU	Glorot	1.7369e-05
	ASHA	ANN	Two, [10, 6]	Ones	ReLU	Glorot	0.014349
	Random Search	ANN	Two, [29, 20]	Ones	ReLU	Glorot	1.7534e-08
			Optimized Hyperparameters				
	Optimizers	**Best Classifier**	Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization (lambda)
mutInfFS Method	Bayesian	ANN	Three, [21,13,10]	Zeros	ReLU	Glorot	0.0001307
	Random Search	ANN	Two, [126, 54]	Zeros	Sigmoid	he	1.0439e-06
			Box constraint	Multiclass coding	Kernel scale	Kernel function	Standardize data
	ASHA	SVM	30.752	onevsall	13.012	Gaussian	No
			Optimized Hyperparameters				
	Optimizers	**Best Classifier**	Layers with Sizes:	Layer Biases	Activation	Layer Weights	Regularization

				Initializer	function	Initializer	(lambda)
mRMR Method	Bayesian	ANN	Three, [15,11,22]	Zeros	ReLU	he	0.00026371
	ASHA	ANN	Three, [8,10,48]	Zeros	ReLU	he	6.9152e-06
	Random Search	ANN	Two, [39, 113]	Ones	Sigmoid	he	3.6141e-06

4.3 Case #2: PU Bearing Dataset:

This section evaluates the performance of feature selection methods when paired with optimizers and selected classifiers for the PU-bearing dataset. Real-bearing damage data in the PU bearing dataset offers a more accurate representation of industrial faults than artificially induced faults in the CWRU dataset. This case study investigates the impact of different hyperparameter optimization methods on feature selection techniques for the PU Bearing Dataset. The best classifiers with their optimized hyperparameters are presented in Table 14.

Figure 14: Confusion matrices(Test data) for LS Method comparing RS, BO, and ASHA Optimization for classification accuracy

Table 15. LS Method Training and Test Results with BO, ASHA, and RS Optimizers

LS Method with Best Classifier Training and Test Results with BO, ASHA, and RS optimizers for PU bearing datasets									
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
LS Method	Bayesian	ANN	100%	95.83%	0.96	0.96	0.96	0.94	40.9546	0.052594
	ASHA	SVM	100%	94.72%	0.95	0.95	0.95	0.93	4.3874	0.056944
	Random Search	ANN	100%	94.72%	0.94783	0.94722	0.94747	0.92963	11.1585	0.042361

4.3.1 Analysis of LS Results

This section discusses the results of the LS method when paired with BO, ASHA, and RS optimizer. Similar to the previous case, 180 iterations were set for BO and RS optimizers. For the ASHA optimizer, the selection of iterations is not user-defined and set based on the settings of $5\times(L+1)$. Given L=6, the total iterations are 35. ASHA optimizer took

55.3544 seconds to evaluate 6 ML algorithms. In comparison, BO took 4307.4306 seconds, and RS took 595.1034 seconds. Both ASHA SVM and RS ANN models achieved the same test accuracy of 94.72%. However, the RS ANN model had a longer training time (11.16 seconds), indicating the complexity of the ANN's hyperparameters. In the BO case, ANN complexity arises further with three layers of sizes, tabulated in Table 15. Moreover, the lambda value is higher (0.00035175) as well, indicating stronger regularization to mitigate overfitting. As a result, this setup achieved a training accuracy of 100% and a test accuracy of 95.83%. According to the confusion matrix in Figure 14, the BO ANN model misclassifies 15 instances, as compared to 19 of the RS ANN and ASHA SVM models. The BO ANN model not only achieved higher test accuracy but also reduced misclassifications, demonstrating its effectiveness in classifying the test data.

Table 16. ReliefF Method Training and Test Results with BO, ASHA, and RS Optimizers

ReliefF Method with Best Classifier Training and Test Results with BO, ASHA, and RS Optimizers for PU bearing datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
ReliefF Method	Bayesian	ANN	100%	97.78%	0.98	0.98	0.98	0.97	18.6383	0.042693
	ASHA	ANN	98.96%	94.72%	0.95	0.95	0.95	0.93	6.4733	0.085417
	Random Search	ANN	100%	96.11%	0.96154	0.96111	0.9612	0.94815	2.7392	0.045139

4.3.2 Analysis of ReliefF Results

This section discusses the results of the ReliefF method when paired with BO, ASHA, and RS optimizers in Table 16. Similar to the previous ReliefF case, all optimizers selected ANN as the best classifier. ASHA took 60.978 seconds, RS took 603.4802 seconds, and BO took 8753.8706 seconds to select the best classifier. BO is probabilistic in nature; as the number of training datasets decreases, it tends to pick complex models with longer training times. In this case, BO selected a deeper ANN model with three layers of sizes. As the model is deeper with effective hyperparameters optimization, it has the highest training and validation time of 18.6383 seconds, with the lowest validation loss of 0.042693. Moreover, it misclassifies only 8 instances, as compared to 14 instances of RS ANN and 19 instances of ASHA SVM models. as shown in Figure 15.

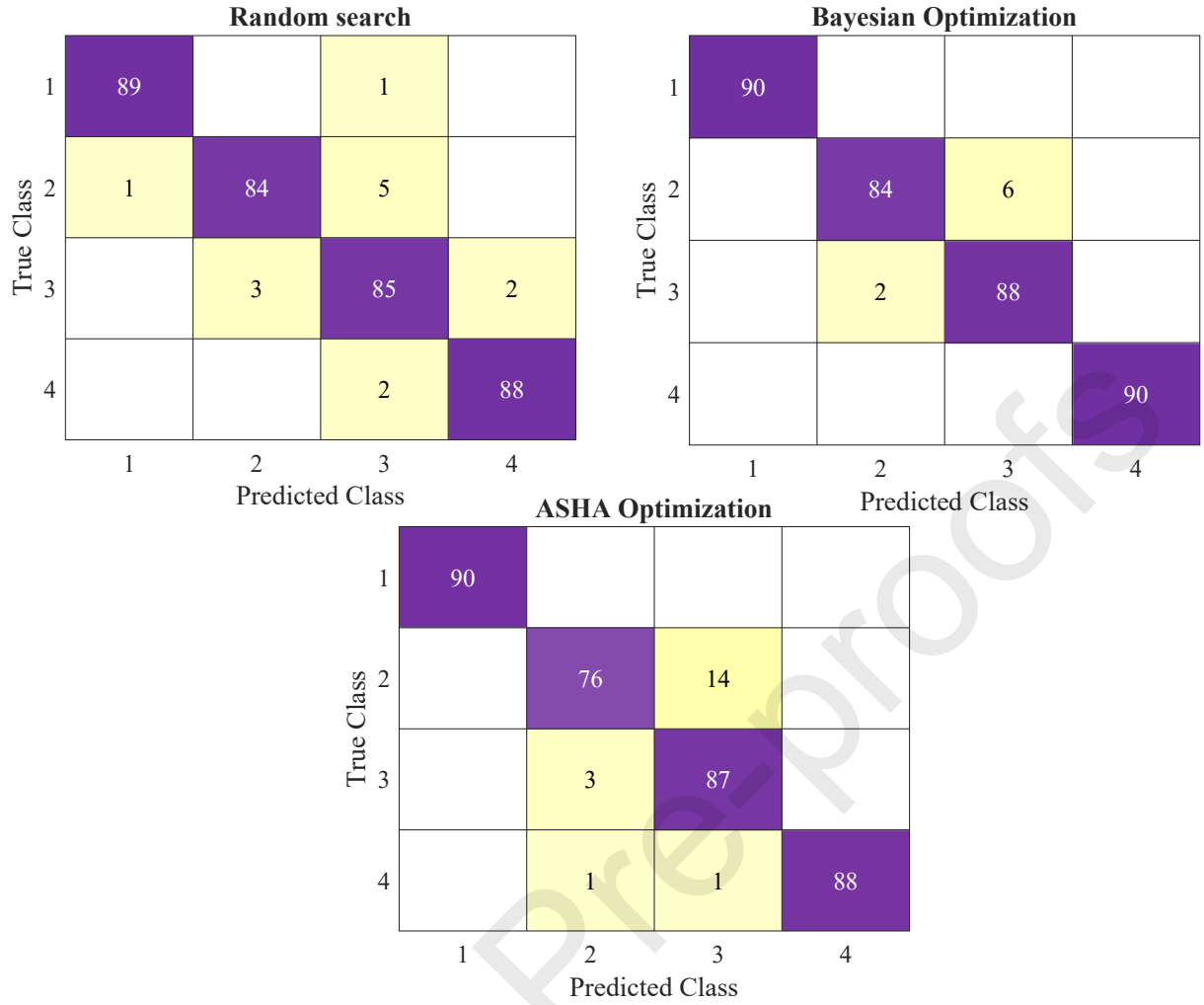


Figure 15: Confusion matrices(Test data) for ReliefF Method comparing RS, BO, and ASHA Optimization for classification accuracy

Table 17. mutInfFS Method Training and Test Results with BO, ASHA, and RS Optimizers

mutInfFS Method with Best Classifier Training and Test Results with BO, ASHA, and RS Optimizers for PU bearing datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
mutInfFS Method	Bayesian	ANN	100%	96.11%	0.96	0.96	0.96	0.95	28.9526	0.047896
	ASHA	SVM	100%	94.44%	0.94	0.94	0.94	0.93	0.9657	0.10625
	Random Search	ANN	100%	97.78%	0.9783	0.97778	0.97785	0.97037	22.5534	0.05

4.3.3 Analysis of mutInfFS Results

This section discusses the results of the mutInfFS method when paired with BO, ASHA, and RS optimizers in Table 17. The time taken to select the best classifier varied among the methods: BO took 4775.3461 seconds, ASHA took 53.3479 seconds, and RS took 734.1748 seconds. The RS ANN model employs a simpler architecture with two layers, compared to the BO model's more complex structure with three layers. RS ANN's simpler architecture may contribute to better generalization and less overfitting. In this case, the choice of sigmoid activation function leads to better convergence and smoother gradient with 100% training. Although ReLU activation is more prominent, it could be dealing with dying ReLU, where neurons stop learning. Moreover, the RS model utilizes a much smaller regularization parameter (1.0439e-06) compared to the BO model (0.0001307). This smaller parameter helps prevent the over-penalization of the weights, thus avoiding underfitting. Despite both RS and BO ANN models achieving 100% training accuracy, the RS model performs better on unseen data, as demonstrated by the metrics in Table 17 and Figure 16.

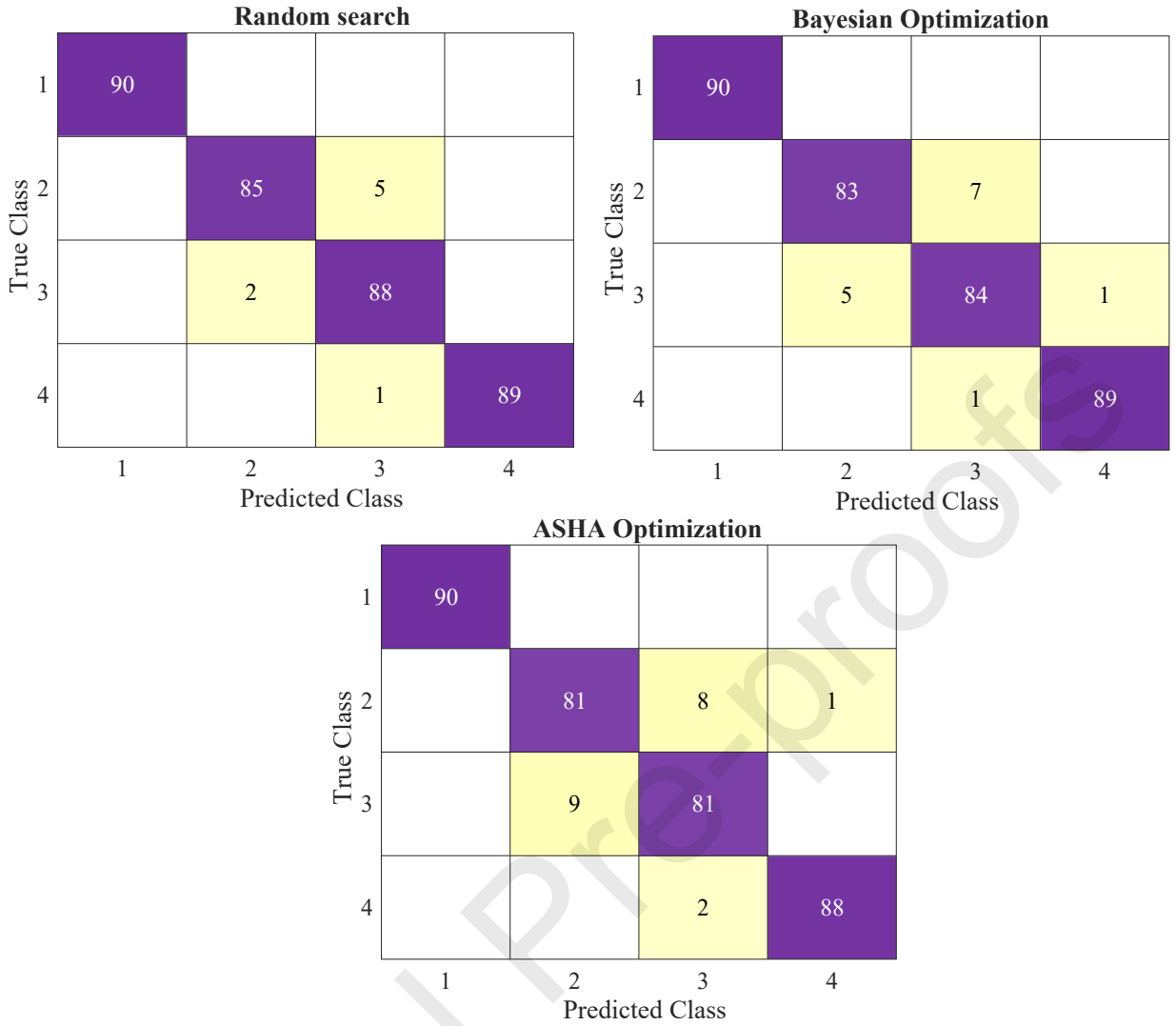


Figure 16: Confusion matrices(Test data) for mutInf Method comparing RS, BO, and ASHA Optimization for classification accuracy

4.3.4 Analysis of mRMR Results

This section presents the mRMR Results when paired with BO, ASHA, and RS. In comparison to CWRU datasets, it is clear that for more complex datasets like PU-bearing datasets, longer training and validation time was required, as evident in Table 18. More complex and deeper ANN layers were selected by BO ANN models in all FS cases. In comparison to BO three-layer ANN with the size of neurons, RS two-layer ANN architecture required less training and validation time. It is clear that the number of layers also requires more computation resources.

Table 18. mRMR Method Training and Test Results with BO, ASHA, and RS Optimizers

mRMR Method with Best Classifier Training and Test Results with BO, ASHA, and RS Optimizers for PU bearing datasets									
Feature Selection Method	**Best Classifier**	**Best Classifier**	Values						
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
mRMR Method	Bayesian	ANN	100%	96.67%	0.97	0.97	0.97	0.96	25.2751	0.043551
	ASHA	ANN	100%	95.83%	0.96	0.96	0.96	0.94	4.231	0.090972
	Random Search	ANN	100%	96.94%	0.97	0.96944	0.96943	0.95926	17.5293	0.045833

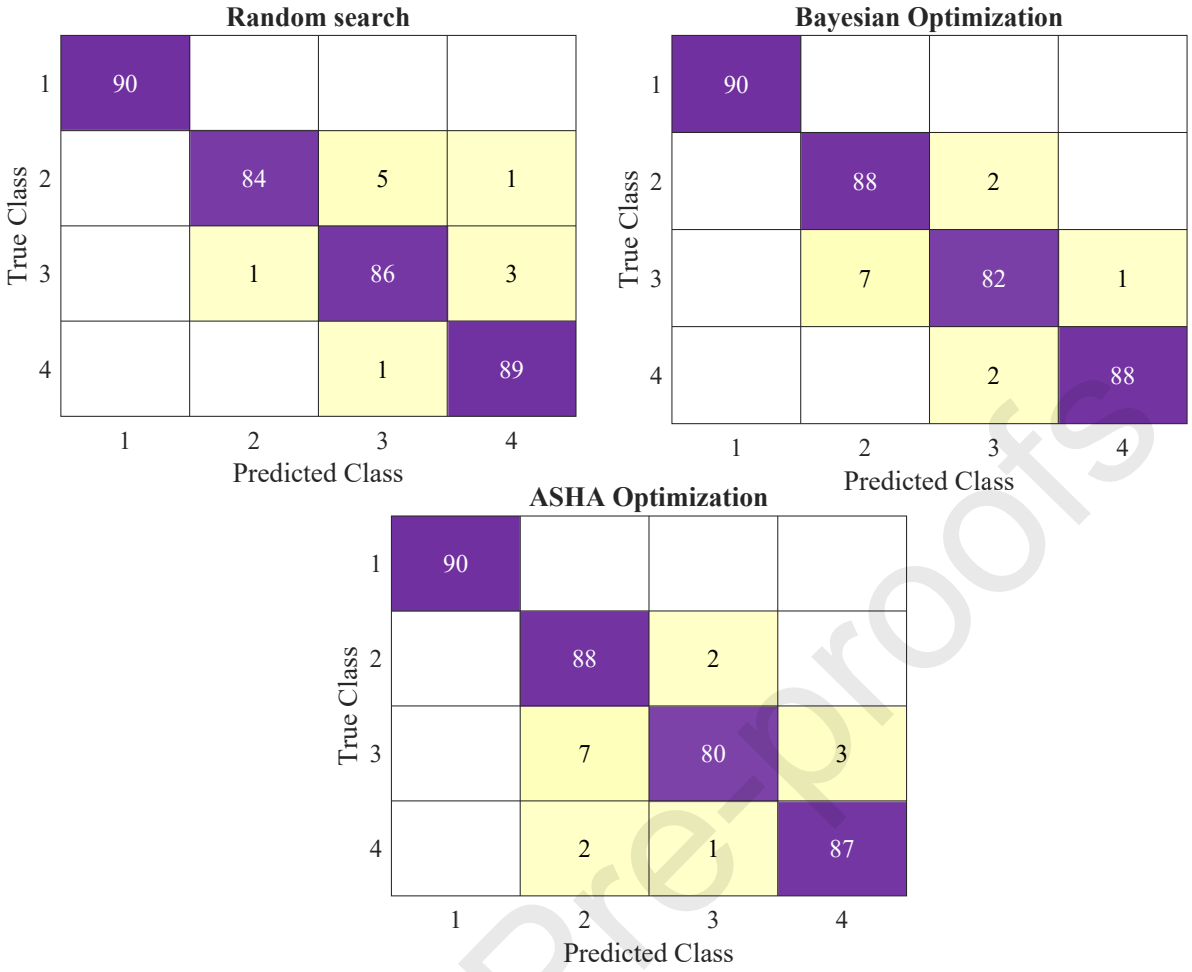


Figure 17: Confusion matrices(Test data) for mRMR Method comparing RS, BO, and ASHA Optimization for classification accuracy

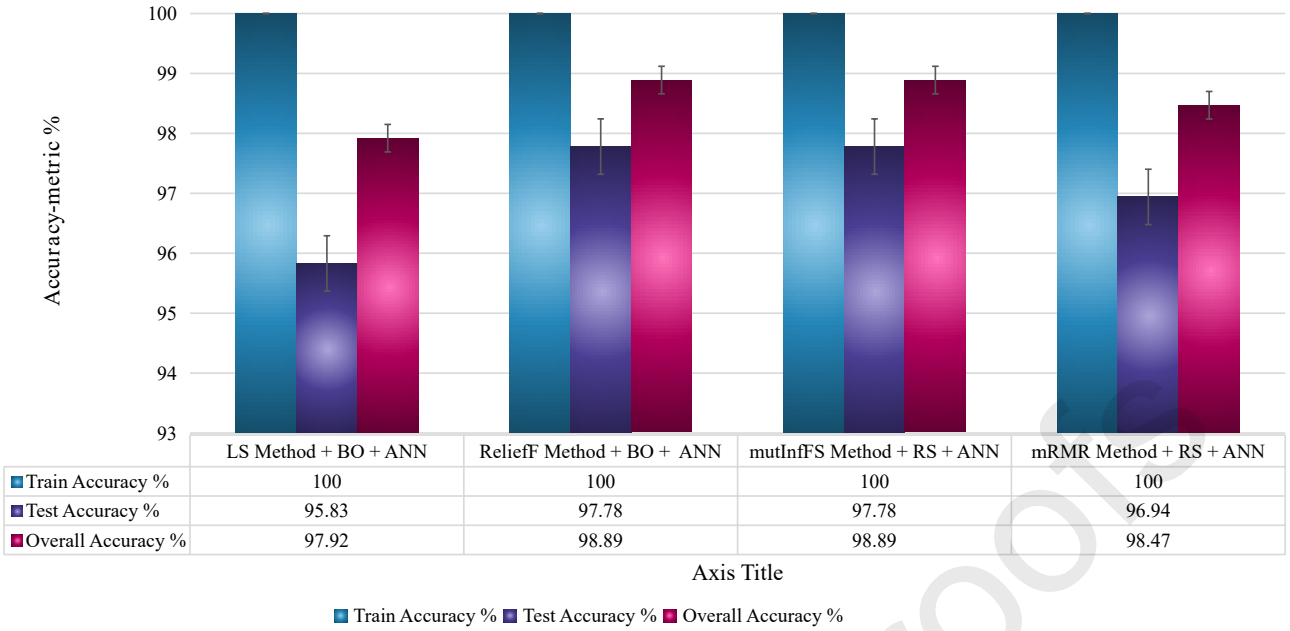
Moreover, the BO tuning process is more time-consuming than RS. As shown in Figure 17, RS marginally outperformed the BO ANN model in terms of performance metrics. Considering both test accuracy and computational efficiency, RS ANN performed well.

Table 19. Comparison across Feature Selection Techniques

Comparison for best optimizers with selected ML algorithms in each Feature Selection Method for PU-bearing datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
LS Method	Bayesian	ANN	100%	95.83%	0.96	0.96	0.96	0.94	40.9546	0.052594
ReliefF Method	Bayesian	ANN	100%	97.78%	0.98	0.98	0.98	0.97	18.6383	0.042693
mutInfFS Method	Random Search	ANN	100%	97.78%	0.9783	0.97778	0.97785	0.97037	22.5534	0.05
mRMR Method	Random Search	ANN	100%	96.94%	0.97	0.96944	0.96943	0.95926	17.5293	0.045833

4.3.5 Optimal Combinations and Recommendations

Based on the findings presented in Table 19, this section explores the optimal combinations of optimizers and classifiers for each FS method using the PU datasets. As mentioned earlier, BO and RS were recommended as suitable for PU datasets, whereas each optimizer selected ANN as the best classifier with 100% training accuracy. However, test accuracies varied, with the ReliefF and mutInfFS methods achieving the highest test accuracy at 97.78%. In terms of computational efficiency, the ReliefF and mRMR methods required the least training and validation time, significantly outperforming the LS Method, which took the longest. In addition, the validation loss was lowest for the ReliefF method, indicating a more effective learning process. Considering both Figure 18 and Figure 19, the ReliefF method, when paired with BO and employing an ANN classifier, emerged as the most effective and efficient approach.



	LS Method + BO + ANN	ReliefF Method + BO + ANN	mutInfFS Method + RS + ANN	mRMR Method + RS + ANN
Train Accuracy %	100	100	100	100
Test Accuracy %	95.83	97.78	97.78	96.94
Overall Accuracy %	97.92	98.89	98.89	98.47

Figure 18: Bar chart (Test data) comparing the performance of best-performed FS methods and Optimizers for PU datasets

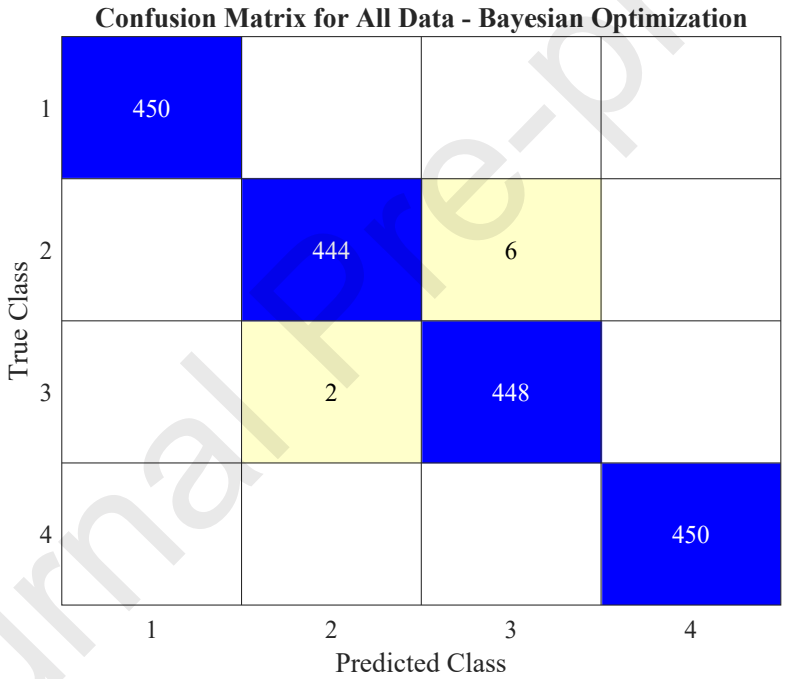


Figure 19: Confusion Matrix for ReliefF Method with BO and ANN classifier (PU datasets)

In comparison to BO, the ASHA optimizer underperforms for PU datasets. Limited data is available for PU bearings in this case study. This scarcity of data can lead to premature convergence on suboptimal solutions in ASHA due to reliance on rapid evaluations and early stopping criteria. As a result, ASHA might stop the optimization process before the model has enough time to completely explore all possible solutions.

4.4 Case #3: DDS Bearing Dataset:

Since DDS-bearing datasets are more scarce than PU-bearing datasets, therefore based on the experiment in the previous section, the ASHA optimizer was excluded from this section. Table 20 presents the best-selected classifiers for each feature selection method when paired with BO and RS optimizer. It is observed that more complex ANN and SVM models were selected by BO and RS optimizers.

Table 20. Optimized hyperparameters for Proposed Feature Selection methods for DDS Bearing Dataset

Summary of optimized hyperparameters for Proposed Feature Selection Methods using BO and RS optimizers			
Feature Selection Method	**Optimization Method**	**Machine Learning Classifier (Algorithm)**	**Hyperparameters**
			Optimized Hyperparameters

	Optimizers	Best Classifier	Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization (lambda)
LS Method	Bayesian	ANN	One, [30]	Ones	Tanh	Glorot	0.00039166
	Random Search	ANN	Two, [16, 17]	Ones	Sigmoid	Glorot	0.001336
			Optimized Hyperparameters				
	Optimizers	**Best Classifier**	Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization (lambda)
ReliefF Method	Bayesian	ANN	One, [29]	Zeros	Tanh	Glorot	1.4623e-05
			Box constraint	Multiclass coding	Kernel scale	Kernel function	Standardize data
	Random Search	SVM	144.49	onevsall	168.84	Gaussian	No
			Optimized Hyperparameters				
	Optimizers	**Best Classifier**	Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization (lambda)
mutInfFS Method	Bayesian	ANN	Three,[52,14,147]	Ones	Tanh	Glorot	6.1534e-05
	Random Search	ANN	Two, [36, 26]	Ones	Tanh	he	3.1184e-05
			Optimized Hyperparameters				
	Optimizers	**Best Classifier**	Box constraint	Multiclass coding	Kernel scale	Kernel function	Standardize data
	Bayesian	SVM	19.003	Onevsone	56.178	Gaussian	No
			Layers with Sizes:	Layer Biases Initializer	Activation function	Layer Weights Initializer	Regularization (lambda)
mRMR Method	Random Search	ANN	One, [35]	Zeros	ReLU	he	4.5724e-08

Table 21. LS Method Training and Test Results with BO and RS Optimizers

LS Method with Best Classifier Training and Test Results with BO and RS optimizers for DDS datasets									
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values						
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
LS Method	Bayesian	ANN	100%	93.75%	0.95	0.94	0.94	0.92	2.3728	0.080461
	Random Search	ANN	100%	95.00%	0.96	0.95	0.95	0.93	2.0602	0.059375

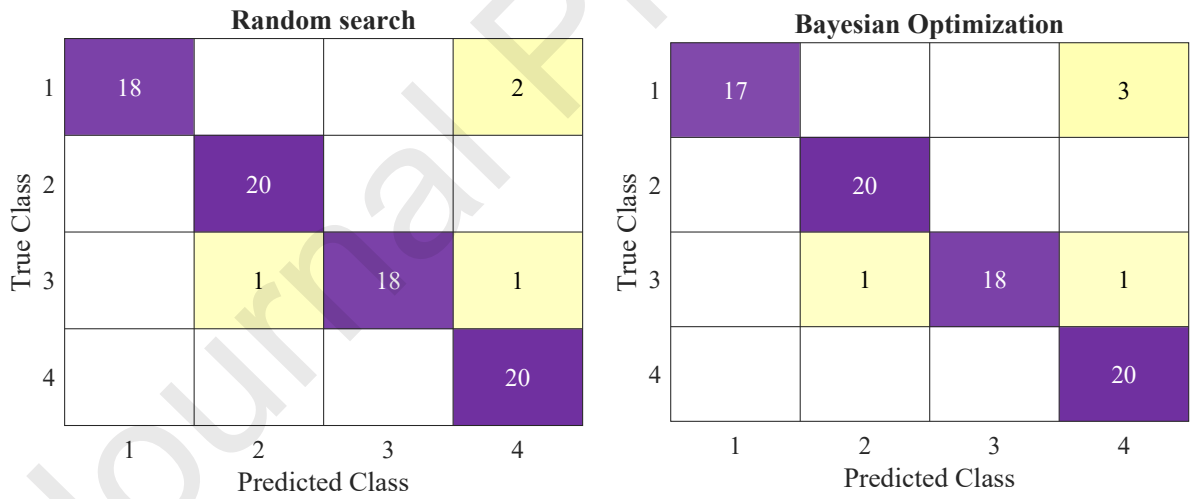


Figure 20: Confusion matrices (Test data) for LS Method comparing BO and RS Optimization for classification accuracy

4.4.1 Analysis of LS Results

Features with higher Laplacian Scores are considered more important as they better capture the intrinsic geometry of the data distribution. From Table 21, the analysis of the LS method's performance reveals that both optimization techniques yielded high training and test accuracies, with RS slightly outperforming BO in terms of test accuracy and validation loss. Here, validation loss tells us about how the train model generalized well on unseen new data. RS, to some extent, generalized well with the ANN model as compared to BO, as evident in Figure 20. RS took 486.6676 seconds to evaluate 6 ML algorithms compared with 694.1281 seconds seconds for BO.

4.4.2 Analysis of ReliefF Results

Based on the analysis in Table 22, the ANN model optimized using BO provides higher accuracy and better performance metrics (Precision, Recall, F1 Score, Kappa) compared to the SVM model optimized using RS. However, the RS SVM model has a lower computational burden, requiring less training and validation time. In comparison, RS took 664 seconds

to evaluate 6 ML algorithms, while BO took 698.2895.

Table 22. ReliefF Method Training and Test Results with BO and RS Optimizers

ReliefF Method with Best Classifier Training and Test Results with BO and RS Optimizers for DDS datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
ReliefF Method	Bayesian	ANN	100%	95.00%	0.96	0.95	0.95	0.93	1.6483	0.072787
	Random Search	SVM	99.38%	92.50%	0.93	0.93	0.93	0.90	0.57517	0.071875

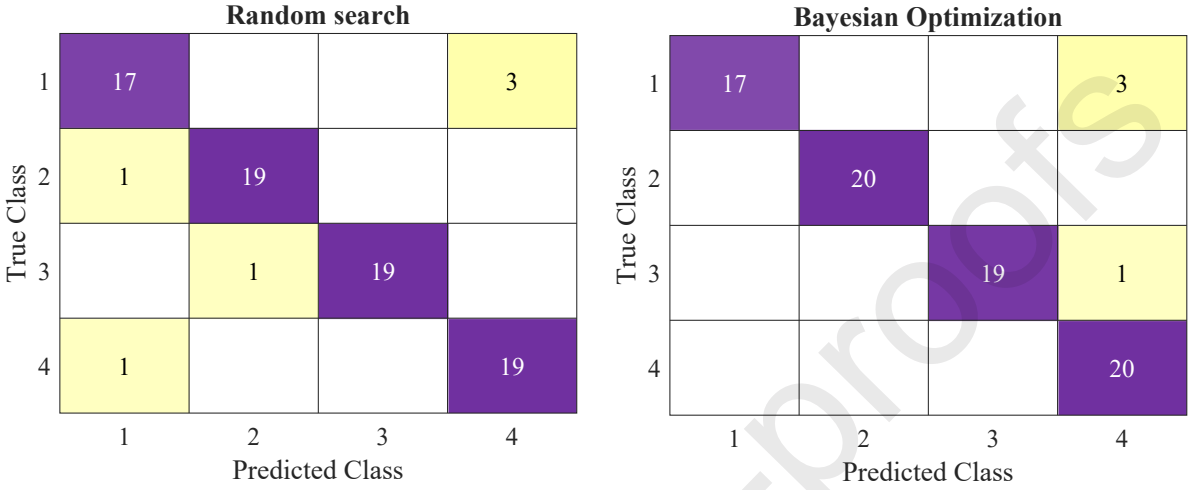


Figure 21: Confusion matrices (Test data) for ReliefF Method comparing BO and RS Optimization for classification accuracy

For applications where accuracy is critical and computational resources are not a primary concern, the ANN model is recommended see Figure 21. On the other hand, for applications where computational efficiency is more important, the high-constraint RS SVM model could be a better choice.

Table 23. mutInfFS Method Training and Test Results with BO and RS Optimizers

mutInfFS Method with Best Classifier Training and Test Results with BO and RS Optimizers for DDS datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
mutInfFS Method	Bayesian	ANN	100%	92.50%	0.93	0.93	0.93	0.90	5.6553	0.079307
	Random Search	ANN	100%	95.00%	0.96	0.95	0.95	0.93	1.8929	0.071875

4.4.3 Analysis of mutInfFS and mRMR Results

In Table 23, BO and RS both selected ANN models, whereas the complex BO ANN model underperformed the less complex model of the RS ANN model. Moreover, the RS ANN model is less regularized, which allows the model to have more complex decision boundaries that can separate the fault classes, potentially capturing more intricate patterns in the unseen data, as evident in Figure 22.

Table 24. mRMR Method Training and Test Results with BO and RS Optimizers

mRMR Method with Best Classifier Training and Test Results with BO and RS Optimizers for DDS datasets										
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Values							
			Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
mRMR Method	Bayesian	SVM	100%	90.00%	0.91	0.90	0.90	0.87	1.2081	0.075962
	Random Search	ANN	100%	87.50%	0.88	0.88	0.88	0.83	0.63978	0.0625

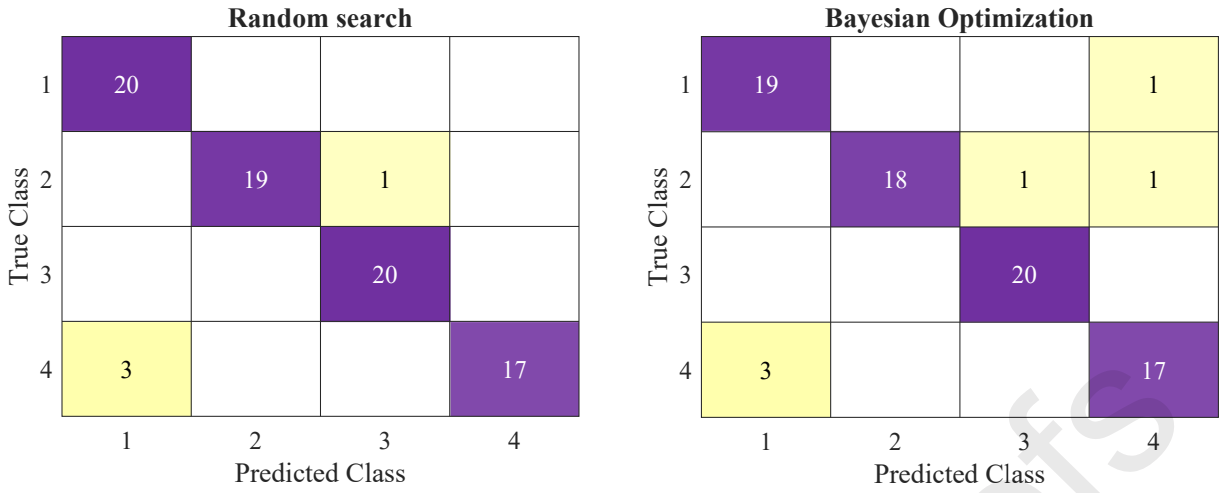


Figure 22: Confusion matrices(Test data) for mutInf Method comparing BO and RS Optimization for classification accuracy

In comparison to the mutInfF method, training and test accuracy reveal that the mRMR features give overfitted ML algorithms when paired with BO and RS. Despite mRMR's intent to minimize redundancy, it may have selected features that are redundant when combined with the model's hyperparameters. For the mRMR method, the BO SVM model has a higher value of the hyperparameter, i.e., kernel scale (56.178), which makes the Gaussian kernel decision boundary smoother but risks overfitting in some cases. This overfitting is evident in Table 24. It achieves 100% training accuracy but only 90% test accuracy. The RS ANN model further degrades with 87.5% test accuracy, indicating a 3% overfit. Figure 23 again shows poor performance by the mRMR method, which made the mutInfF method the preferred choice.

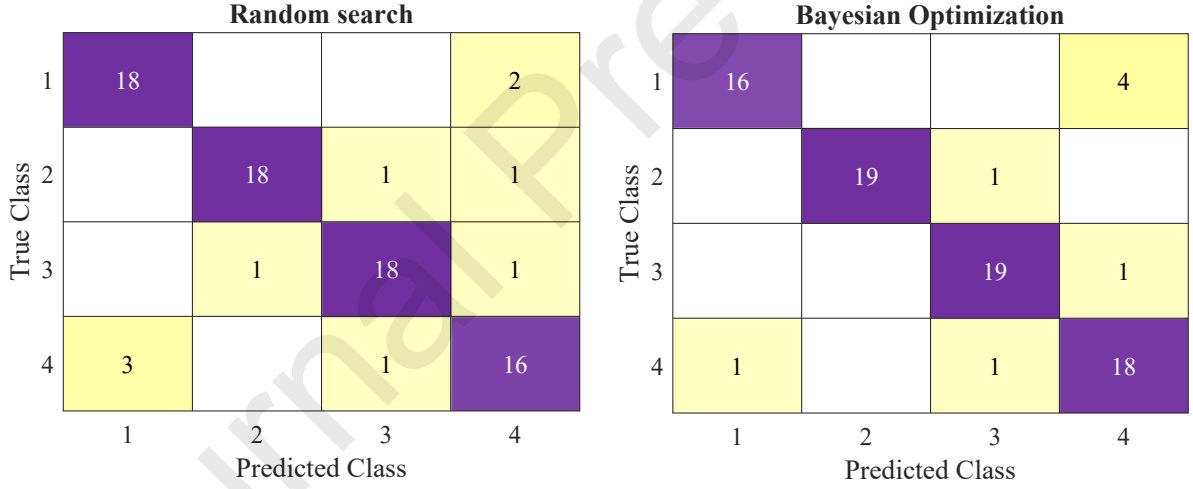


Figure 23: Confusion matrices (Test data) for mRMR Method comparing BO and RS Optimization for classification accuracy

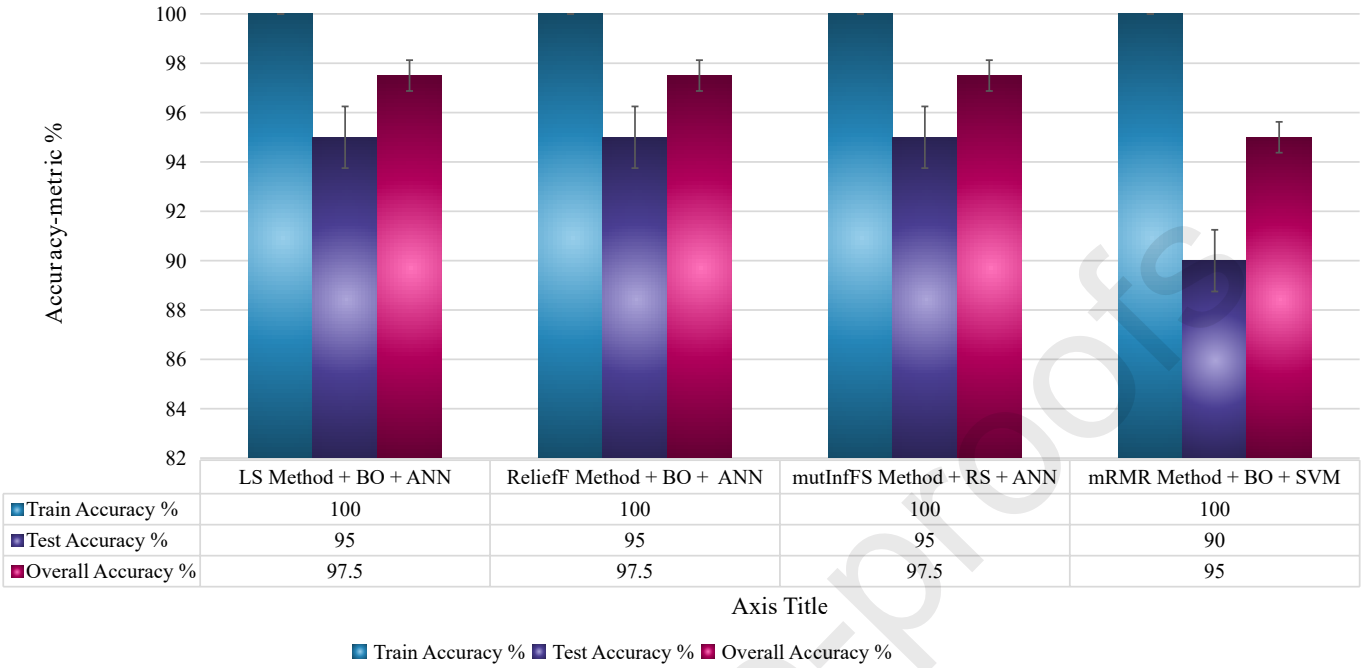
Table 25. Comparison across Feature Selection Techniques

Comparison for best optimizers with selected ML algorithms in each Feature Selection Method for DDS datasets								
			Values					
Feature Selection Method	**Best Optimizers**	**Best Classifier**	Test Precision	Test Recall	Test F1Score	Test Kappa	Training and Validation time (seconds)	Validation Loss
LS Method	Random Search	ANN	0.96	0.95	0.95	0.93	2.0602	0.059375
ReliefF Method	Bayesian	ANN	0.96	0.95	0.95	0.93	1.6483	0.072787
mutInfFS Method	Random Search	ANN	0.96	0.95	0.95	0.93	1.8929	0.071875
mRMR Method	Bayesian	SVM	0.91	0.90	0.90	0.87	1.2081	0.075962

4.4.5 Optimal Combinations and Recommendations

Table 25 evaluates the performance of the best optimizers coupled with selected ML algorithms for each FS method applied to DDS datasets. Most of the optimizers selected ANN as the best classifier to capture complex relationships in the DDS data. For the LS method, the RS ANN model is the best classifier, where ANN uses backpropagation with appropriate activation functions and optimized hyperparameters to achieve good results. However, it had the longest training and validation time at 2.0602 seconds and a validation loss of 0.059375. This longer training time is likely due to the complexity introduced by the two-layer ANN structure, which, while effective, requires more computational resources

and time to optimize. Similar trends were observed in the mutInfF method when paired with the RS ANN model. Figure 24 reveals that the LS, ReliefF, and mutInfFS methods exhibited better and comparable performance metrics. However, the ReliefF method stood out with a shorter training time due to its effective feature selection process with an effective and simple BO ANN model.



	LS Method + BO + ANN	ReliefF Method + BO + ANN	mutInfFS Method + RS + ANN	mRMR Method + BO + SVM
Train Accuracy %	100	100	100	100
Test Accuracy %	95	95	95	90
Overall Accuracy %	97.5	97.5	97.5	95

Figure 24: Bar chart (Test data) comparing the performance of best-performed FS methods and Optimizers for DDS data

In both PU and DDS-bearing datasets, the scarcity of data with real faulty characteristics makes it difficult to classify faults. However, in both cases, BO's probabilistic framework and well-informed search space combined with the ReliefF method enhance fault classification, as evident in Figure 25. Table A highlights all optimizer results in supplementary file.

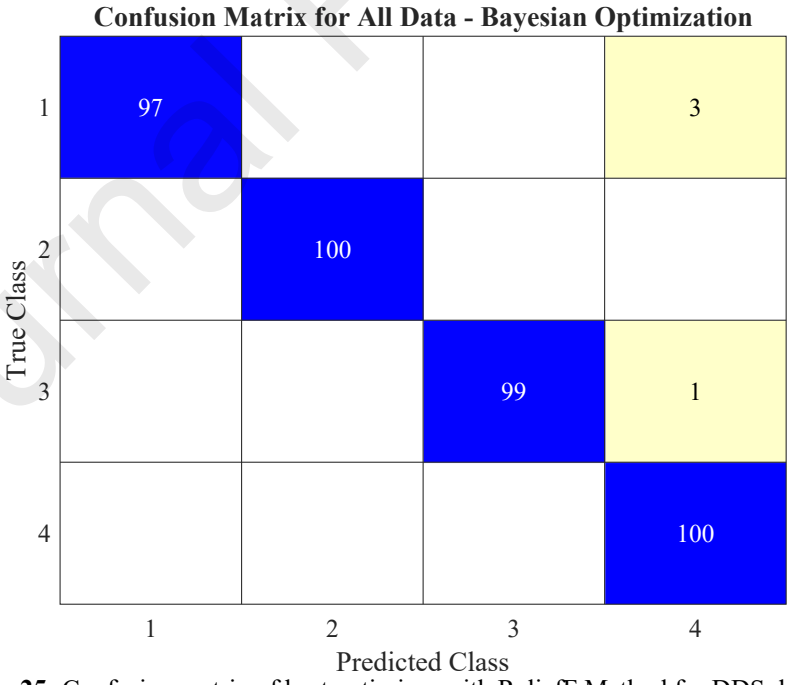


Figure 25: Confusion matrix of best optimizer with ReliefF Method for DDS datasets

5. Computational Burden

Another advantage of the ReliefF method over other FS methods is the lower computational burden with a robust feature selection process. ReliefF compares local instances with their neighbours to identify essential features and updates feature weights, which speeds up the process. The estimated time to process a 0.09-second signal (comprising 1024 data points) for the CWRU experimental platform using MVMD followed by HHT applied to 5 modes, extracting 33 features per frequency and amplitude modes, then calculating SR correlation, applying ReliefF, performing Y-J transformation and

then subsequently applying 6 ML algorithms. This processing is conducted on a personal computer equipped with an Intel Core i9-13900HX processor, which operates at a maximum turbo frequency of 5.4 GHz with 8 operators per cycle. The estimated computational burden for applying the HHT on the modes extracted by MVMD is approximately 1.70×10^{-6} seconds. As an experiment, the decomposed results obtained via MVMD and HHT were applied directly to ML algorithms. However, due to the curse of dimensionality, these algorithms underperformed with an almost 30 times increase in computational burden. Therefore, feature processing with the help of ReliefF and other methods solves these issues. As a result, the time taken for each algorithm is as follows: the ensemble took 9.134×10^{-6} seconds, SVM took 10.9×10^{-5} seconds, KNN took 10.9×10^{-5} seconds, ANN took 3.639×10^{-5} seconds, DT took 7.180×10^{-6} seconds, and NB took 6.273×10^{-6} seconds.

On a 100 MHz radiation-hardened edge processor (approximately 54× slower), MVMD + HHT and SVM steps translate to around 92 µs and 5.9 ms. Worst case scenario, total latency per segment of 6 ms for SVM model is well within the 90 ms acquisition window. As a result, leaving 80 ms for sensor tasks and confirming real-time requirements for lunar-base diagnostics with suitable margin.

Table 26. Performance Comparison with existing ML or DL methods on the CWRU and PU datasets

Performance comparison between the proposed method and other ML and DL fault classification methods					
Serial Number	**ML/DL Models**	**Signal Decomposition**	**Extracted Features**	**Datasets (Window Size)**	**Values (%)**
					Overall Accuracy
Ref [68]	CNN, Transfer Learning	Wavelet Convolution	Autonomous	CWRU datasets (2048)	99.73%
Ref [69]	ANN	VMD	Normalized Energy	CWRU datasets (1024)	99.30%
Ref [70]	Net2Net transformation	×	Autonomous	PU datasets (400)	96.24%
Ref [71]	Deep Residual Network	Input feature mappings (IFMs)	Signal-to-IFMs	PU datasets (4096)	99.70%
Proposed	ASHA-SVM	MVMD+HHT	LS	CWRU datasets (1024)	99.94%
Proposed	BO-SVM	MVMD+HHT	ReliefF	PU datasets (2048)	98.89%

6. Comparison with existing ML and DL Methods

To further validate the assertion of the proposed framework, we have added a comparison with traditional ML and DL methods on the CWRU and PU datasets. As shown in Table 26, the proposed method shows better performance in terms of overall accuracy [68-71]. For instance, the proposed method achieves 100% accuracy with 1024 data points from the vibration signal input, whereas Liao *et al.* [68] required 2048 data points to achieve an accuracy of 99.73%. The key difference between the proposed method and DL methods lies in the approach to feature extraction and signal decomposition. The proposed method depends upon conventional signal processing to decompose and extract features, while other methods use deep neural networks. This makes the proposed method less complex in terms of training and tuning, leading to improved efficiency in terms of both computational resources and time. Future work will comprehensively explore the approaches detailed in [72-75] to extend and refine our proposed framework.

Table 27. The result of the fault classification on the IMS dataset.

The result of the binary fault classification on the IMS dataset								
Test Dataset	**Class Types**	**Feature Selection Method**	**Best Optimizers**	**Best Classifier**	Values			
					Test Precision	Test Recall	Test F1Score	Test Accuracy
IMS dataset 1	(IR, OR)	ReliefF Method	BO	Ensemble	0.98	0.98	0.98	98%
IMS dataset 2	(Healthy, OR)	ReliefF Method	BO	Ensemble	0.96	0.95	0.95	96%
IMS dataset 3	(Healthy, OR)	ReliefF Method	BO	Ensemble	0.98	0.98	0.98	98%
Average	×	×	×	×	0.97	0.97	0.97	97.33%

7. Classification results using NASA Bearing Dataset

This section further conducts experiments based on a NASA-bearing dataset repository that helps ensure the proposed method's robustness [76]. The test rig setup is shown in Figure 26, and the data set is recorded by the Intelligent Maintenance System (IMS) of the National Science Foundation Industry (NSFI) at a 20 kHz sampling frequency. Three datasets were extracted, in which inner race (IR) defect occurred in bearing three and roller element defect in bearing 4 for IMS dataset 1. In contrast, outer race failure occurred in bearing 1 and bearing 3 for IMS dataset 2 and dataset 3. Table 27 displays high binary classification performance for each IMS dataset, whereas the BO optimizer outperforms ASHA and RS and selected ensemble as the best classifier. In addition, 2 dB to 6 dB noise is also added to input signals to mimic extreme conditions. However, due to the proposed denoising, robust features and adaptive ML algorithms display robust results with an average overall accuracy of 96%. Thus, the proposed framework can be integrated for autonomous fault recognition in extreme environments.

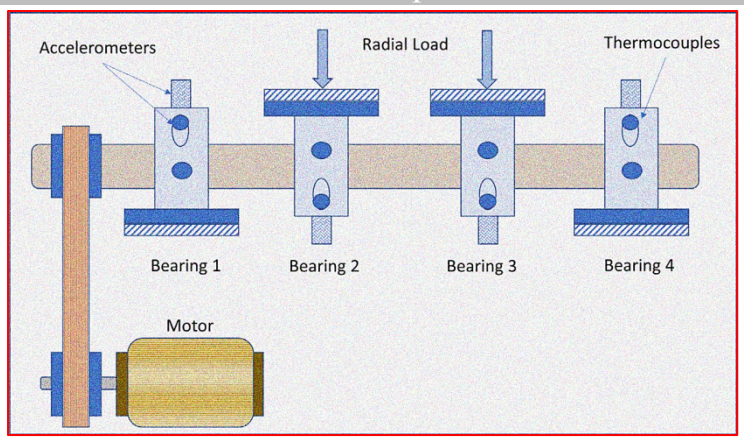


Figure 26: A complete experimental setup, including an IMS sensor placement and a bearing test rig.

Table 28. Evaluation of MVMD, MEMD, and MEWT for Multichannel Signal Decomposition.

Performance Benchmarking of MVMD, MEMD, and MEWT Techniques			
Method	**Signal 1(dB)**	**Signal 2(dB)**	**Avg SNR**
MVMD	6.72 dB	9.06 dB	7.89 dB
MEMD	10.03 dB	5.40 dB	7.71 dB
MEWT	0.91 dB	1.83 dB	1.37 dB

Table 29. Decomposition Quality Metrics (Based on IMF1)

Decomposition Quality Metrics (Based on IMF1)			
Metric	**MVMD**	**MEMD**	**MEWT**
Mode Consistency (CC)	-0.01	-0.01	-0.02
Energy Ratio (IMF1)	18%	37%	31%
Sparsity (Kurtosis)	5.83	4.01	5.64
Entropy	5.04	4.34	5.02

8. Comparison of MVMD, MEMD, and MEWT

We compared the average SNRs of two high-noise signals from the PU dataset to support the rationale for MVMD selection over MEMD and MEMT. Table 28 demonstrate that MVMD reached the highest average SNR of 7.89 dB, indicating robust ability to suppress noise and preserve information. In Table 29, MEMD recorded higher energy ratio in IMF1 at 37%, this reflects stronger concentration of signal energy in the first mode, potentially leading to mode mixing [37]. In contrast, MVMD achieved higher sparsity with a kurtosis of 5.83, showcasing better isolation of fault-related features. Entropy values further support this, with MVMD maintaining a balance between signal complexity and structure (5.04) compared to MEMD (4.34) and MEWT (5.02). Mode consistency scores were similar for MVMD and MEMD (-0.01), while MEWT had a small drop (-0.02). Overall, these results support MVMD as a more stable and effective method for signal decomposition in complex, multichannel conditions, as expected in space systems.

9. Conclusion

In this article, a novel versatile automated framework was proposed by exploring four FS methods with three advanced optimization techniques (BO, RS, and ASHA) for the improvement of fault classification in rolling bearings used in space-bound DC power systems. To address the challenge of collecting non-linear and non-stationary datasets from multichannel bearings signals, MVMD and HHT were combined for adaptive decomposition, followed by FS from multi-domain features such as time, frequency, and entropy. Multi-domain features provide a holistic view of the signal, capturing different aspects of faults which single-domain features may miss. This also helps in reducing the risk of overfitting and improves the model's ability to generalize to new, unseen data. Various performance metrics were measured to assess classifier performance, including classification accuracy, precision, recall, kappa, training and validation time, and validation loss. The critical comparisons of different FS methods and optimizers revealed the following findings using CWRU, PU, and DDS datasets:

1. For the CWRU dataset, the ASHA optimizer has been suggested due to its ability to handle a large number of samples with faster tuning and evaluation of ML algorithms. Out of 6 ML algorithms, BO, RS, and ASHA optimizers selected ANN, SVM, and KNN as best classifiers. On average, KNN showed faster training and validation, but complex ANN and SVM architectures gave better accuracy. The best combination with moderate computational burden was LS with the ASHA optimizer and SVM classifier, achieving an overall accuracy of

99.94%, with precision, recall, F1 score, and kappa nearing 1.

2. As the data scarcity starts to increase in the PU dataset, the BO optimizer with informed space search for hyperparameters optimization is suggested. However, the RS optimizer with parallel settings took less time to evaluate ML algorithms. Due to real bearing damages in PU datasets, both optimizers selected complex models with SVM and ANN as their preferred choice. The best combination was ReliefF with the BO optimizer and ANN architecture, which demonstrated an overall accuracy of 98.89%, precision of 0.98, recall of 0.98, F1 score of 98, and a kappa of 0.97, with a moderate training time of 18.6383 seconds.
3. Similar trends were observed for more scarce DDS datasets as well, where the BO optimizer performed well due to its probabilistic nature. BO again paired with the ReliefF method and ANN architecture to produce good results with lower training and validation time of only 1.6483 seconds. This suggests that the multivariate ReliefF method not only eliminates redundant features but also helps to increase classification accuracy.

The collective utilization of these existing techniques for extraterrestrial environments transforms them into a comprehensive, autonomous, and end-to-end solution that delivers a robust fault detection framework. Future research will test this framework under simulated space conditions, including vacuum, thermal cycling, and radiation, to optimize diagnostic models for bearing reliability in lunar missions. Data from these experiments will refine our proposed framework. Meanwhile, collaboration with space agencies will further cross-validate our results.

Acknowledgments

We would like to express our sincere gratitude to all those who have contributed to the completion of this work. All authors have read and agreed to the revised version of the manuscript. All authors planned the study and contributed to the idea and field of information.

References

[1] M. Smith *et al.*, "The Artemis program: an overview of NASA's activities to return humans to the moon," in *2020 IEEE aerospace conference*, 2020: IEEE, pp. 1-10.
[2] NASA. "NASA Artemis Program." https://www.nasa.gov/humans-in-space/artemis/ (accessed).
[3] E. S. Agency. "ESA Moon Village: European Space Agency, Moon Village Concept Overview, 2023." https://www.esa.int/ (accessed.
[4] C. s. S. Program. "China's International Lunar Research Station (ILRS): CNSA 2023 Plans." https://www.cnsa.gov.cn/ (accessed.
[5] A. Biswas, S. Ray, D. Dey, and S. Munshi, "Detection of simultaneous bearing faults fusing cross correlation with multikernel SVM," *IEEE Sensors Journal,* vol. 23, no. 13, pp. 14418-14427, 2023.
[6] Z. Wang, Q. Zhang, J. Xiong, M. Xiao, G. Sun, and J. He, "Fault diagnosis of a rolling bearing using wavelet packet denoising and random forests," *IEEE Sensors Journal,* vol. 17, no. 17, pp. 5581-5588, 2017.
[7] U.-P. Chong, "Signal model-based fault detection and diagnosis for induction motors using features of vibration signal in two-dimension domain," *Strojniški vestnik,* vol. 57, no. 9, pp. 655-666, 2011.
[8] M. E. H. Benbouzid, "A review of induction motors signature analysis as a medium for faults detection," *IEEE transactions on industrial electronics,* vol. 47, no. 5, pp. 984-993, 2000.
[9] S. Ding, "Model-based fault diagnosis in dynamic systems using identification techniques, Silvio Simani, Cesare Fantuzzi and Ron J. Patton, Springer: London, 2003, 282pp. ISBN 1-85233-685-4," ed: Wiley Online Library, 2005.
[10] R. Isermann, "Model-based fault-detection and diagnosis–status and applications," *Annual Reviews in control,* vol. 29, no. 1, pp. 71-85, 2005.
[11] J. Zarei, M. A. Tajeddini, and H. R. Karimi, "Vibration analysis for bearing fault detection and classification using an intelligent filter," *Mechatronics,* vol. 24, no. 2, pp. 151-157, 2014.
[12] H. Ye, B. Yang, Y. Han, and N. Chen, "State-of-the-art solar energy forecasting approaches: Critical potentials and challenges," *Frontiers in Energy Research,* vol. 10, p. 268, 2022.
[13] T. R. Mohan, J. P. Roselyn, R. A. Uthra, D. Devaraj, and K. Umachandran, "Intelligent machine learning based total productive maintenance approach for achieving zero downtime in industrial machinery," *Computers Industrial Engineering,* vol. 157, p. 107267, 2021.
[14] Z. Wang *et al.*, "A generalized fault diagnosis framework for rotating machinery based on phase entropy," *Reliability Engineering & System Safety,* vol. 256, p. 110745, 2025.
[15] Z. Wang *et al.*, "Few-shot fault diagnosis for machinery using multi-scale perception multi-level feature fusion image quadrant entropy," *Advanced Engineering Informatics,* vol. 63, p. 102972, 2025.
[16] Z. Wang *et al.*, "A High-Accuracy Fault Detection Method Using Swarm Intelligence Optimization Entropy," *IEEE Transactions on Instrumentation and Measurement,* 2024.
[17] J. B. Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, and F. Fnaiech, "Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals," *Applied Acoustics,* vol. 89, pp. 16-27, 2015.
[18] Y. Lei, Z. He, and Y. Zi, "Application of an intelligent classification method to mechanical fault diagnosis," *Expert Systems with Applications,* vol. 36, no. 6, pp. 9941-9948, 2009.

[19] W. Laala, A. Guedidi, and A. Guettaf, "Bearing faults classification based on wavelet transform and artificial neural network," *International Journal of System Assurance Engineering Management,* pp. 1-8, 2023.
[20] M. Han and J. Pan, "A fault diagnosis method combined with LMD, sample entropy and energy ratio for roller bearings," *Measurement,* vol. 76, pp. 7-19, 2015.
[21] A. Soualhi, K. Medjaher, and N. Zerhouni, "Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression," *IEEE Transactions on instrumentation measurement,* vol. 64, no. 1, pp. 52-62, 2014.
[22] L. Lu, J. Yan, and C. W. de Silva, "Dominant feature selection for the fault diagnosis of rotary machines using modified genetic algorithm and empirical mode decomposition," *Journal of Sound Vibration,* vol. 344, pp. 464-483, 2015.
[23] H. Abburi *et al.*, "A Closer Look at Bearing Fault Classification Approaches," *arXiv preprint* 2023.
[24] M. A. Jamil and S. Khanam, "Fault Classification of Rolling Element Bearing in Machine Learning Domain," *International Journal of Acoustics Vibration,* vol. 27, no. 2, 2022.
[25] Y. Yinghua, S. Guoqiang, and S. Xiang, "Fault monitoring and classification of rotating machine based on PCA and KNN," in *2018 Chinese Control And Decision Conference (CCDC)*, 2018: IEEE, pp. 1795-1800.
[26] M. Unal, M. Onat, M. Demetgul, and H. Kucuk, "Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network," *Measurement,* vol. 58, pp. 187-196, 2014.
[27] H. M. Ertunc, H. Ocak, and C. Aliustaoglu, "ANN-and ANFIS-based multi-staged decision algorithm for the detection and diagnosis of bearing faults," *Neural Computing Applications,* vol. 22, pp. 435-446, 2013.
[28] X. Yan and M. Jia, "A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing," *Neurocomputing,* vol. 313, pp. 47-64, 2018.
[29] X. Zhang, Y. Liang, and J. Zhou, "A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM," *Measurement,* vol. 69, pp. 164-179, 2015.
[30] B. Wang, W. Qiu, X. Hu, and W. Wang, "A rolling bearing fault diagnosis technique based on recurrence quantification analysis and Bayesian optimization SVM," *Applied Soft Computing,* vol. 156, p. 111506, 2024.
[31] J. Zheng, M. Gu, H. Pan, and J. Tong, "A fault classification method for rolling bearing based on multisynchrosqueezing transform and WOA-SMM," *IEEE Access,* vol. 8, pp. 215355-215364, 2020.
[32] Z. Guo, M. Liu, Y. Wang, and H. Qin, "A new fault diagnosis classifier for rolling bearing united multi-scale permutation entropy optimize VMD and cuckoo search SVM," *IEEE Access,* vol. 8, pp. 153610-153629, 2020.
[33] X. Tang, Q. He, X. Gu, C. Li, H. Zhang, and J. Lu, "A novel bearing fault diagnosis method based on GL-mRMR-SVM," *Processes,* vol. 8, no. 7, p. 784, 2020.
[34] Q. Wang, S. Wang, B. Wei, W. Chen, and Y. Zhang, "Weighted K-NN classification method of bearings fault diagnosis with multi-dimensional sensitive features," *IEEE Access,* vol. 9, pp. 45428-45440, 2021.
[35] L. Wan, K. Gong, G. Zhang, X. Yuan, C. Li, and X. J. I. A. Deng, "An efficient rolling bearing fault diagnosis method based on spark and improved random forest algorithm," *IEEE Access,* vol. 9, pp. 37866-37882, 2021.
[36] S. N. Pichika, G. Meganaa, S. G. Rajasekharan, and A. Malapati, "Multi-component fault classification of a wind turbine gearbox using integrated condition monitoring and hybrid ensemble method approach," *Applied Acoustics,* vol. 195, p. 108814, 2022.
[37] Y. Lv, R. Yuan, and G. Song, "Multivariate empirical mode decomposition and its application to fault diagnosis of rolling bearing," *Mechanical Systems Signal Processing,* vol. 81, pp. 219-234, 2016.
[38] Z. Wang, Y. Jin, S. Zhang, S. Cao, and Y. Liao, "Multivariate Empirical Wavelet Transform and Its Application to Rolling Bearings," in *International conference on the Efficiency and Performance Engineering Network*, 2024: Springer, pp. 229-235.
[39] T. Guan, S. Liu, W. Xu, Z. Li, H. Huang, and Q. Wang, "Rolling Bearing Fault Diagnosis Based on Component Screening Vector Local Characteristic-Scale Decomposition," *Shock Vibration,* vol. 2022, no. 1, p. 9925681, 2022.
[40] F. Wang, X. Liu, G. Deng, X. Yu, H. Li, and Q. Han, "Remaining life prediction method for rolling bearing based on the long short-term memory network," *Neural processing letters,* vol. 50, pp. 2437-2454, 2019.
[41] Q. Hu *et al.*, "Feature selection for monotonic classification," *IEEE Transactions on Fuzzy Systems,* vol. 20, no. 1, pp. 69-81, 2011.
[42] K. Loparo. Case western reserve university bearing data centre website [Online] Available: (11-17). http://engineering. case. edu/bearingdatacenter/download-data-file
[43] C. Lessmeier, J. K. Kimotho, D. Zimmer, and W. Sextro, "Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification," in *PHM Society European Conference*, 2016, vol. 3, no. 1.
[44] H. Pan, W. Jiao, T. Yan, A. U. Rehman, A. Wan, and S. Yang, "Combining kernel principal component analysis and spatial group-wise enhance convolutional neural network for fault recognition of rolling element bearings," *Measurement Science Technology,* vol. 34, no. 12, p. 125003, 2023.
[45] I. M. Johnstone and B. W. Silverman, "Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences," 2004.
[46] M. Robnik-Šikonja and I. J. M. l. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," vol. 53, pp. 23-69, 2003.
[47] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *Journal of biomedical informatics,* vol. 85, pp. 189-203, 2018.
[48] V. K. Garg, A. Narang, and S. Bhattacherjee, "Real-time memory efficient data redundancy removal algorithm," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010, pp. 1259-1268.
[49] M. T. Sadiq *et al.*, "Motor imagery BCI classification based on multivariate variational mode decomposition," *IEEE Transactions on Emerging Topics in Computational Intelligence,* vol. 6, no. 5, pp. 1177-1189, 2022.
[50] Z. Xu, J. Wang, and Y. Shen, "Weak faults diagnosis for rolling bearings based on variational Hilbert Huang transform," in *2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)*, 2020: IEEE, pp. 471-476.
[51] S. Mohanty, K. K. Gupta, and K. S. Raju, "Comparative study between VMD and EMD in bearing fault diagnosis," in *2014 9th*

International Conference on Industrial and Information Systems (ICIIS), 2014: IEEE, pp. 1-6.
[52] M. Zhang, J. Yin, and W. Chen, "Rolling bearing fault diagnosis based on time-frequency feature extraction and IBA-SVM," *IEEE Access,* vol. 10, pp. 85641-85654, 2022.
[53] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence,* vol. 97, no. 1-2, pp. 273-324, 1997.
[54] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research,* vol. 3, no. Mar, pp. 1157-1182, 2003.
[55] P. Bailey, and Ahmad Emad. wCorr: Weighted Correlations R package version
[56] S. Hijazi, "Semi-supervised Margin-based Feature Selection for Classification," Université du Littoral Côte d'Opale; École Doctorale des Sciences et de …, 2019.
[57] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *Cybernetics information technologies,* vol. 19, no. 1, pp. 3-26, 2019.
[58] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," *Advances in neural information processing systems,* vol. 18, 2005.
[59] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC press, 2007.
[60] M. Zaffalon and M. Hutter, "Robust feature selection using distributions of mutual information," in *Proceedings of the 18th international conference on uncertainty in artificial intelligence (UAI-2002)*, 2002: San Francisco, CA, pp. 577-584.
[61] Y. Libing, T. Nguyen-Thoi, and T.-T. Tran, "Predicting the friction angle of clays using a multi-layer perceptron neural network enhanced by Yeo-Johnson transformation and coral reefs optimization," *Journal of Rock Mechanics Geotechnical Engineering,* 2024.
[62] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, and M. Data, "Practical machine learning tools and techniques," in *Data mining*, 2005, vol. 2, no. 4: Elsevier Amsterdam, The Netherlands, pp. 403-413.
[63] L. Li *et al.*, "A system for massively parallel hyperparameter tuning," *Proceedings of Machine Learning Systems,* vol. 2, pp. 230-246, 2020.
[64] Matlab, "Hyperparameter Optimization in Classification Learner App, 2022, https://uk.mathworks.com/help/stats/hyperparameter-optimization-in-classification-learner-app.html," ed.
[65] W. Pannakkong, K. Thiwa-Anont, K. Singthong, P. Parthanadee, and J. Buddhakulsomsiri, "Hyperparameter tuning of machine learning algorithms using response surface methodology: a case study of ANN, SVM, and DBN," *Mathematical problems in engineering,* vol. 2022, no. 1, p. 8513719, 2022.
[66] M. Z. Yousaf, S. Khalid, M. F. Tahir, A. Tzes, and A. Raza, "A novel dc fault protection scheme based on intelligent network for meshed dc grids," *International Journal of Electrical Power Energy Systems,* vol. 154, p. 109423, 2023.
[67] L. Cui, Y. Zhang, R. Zhang, Q. H. J. I. T. o. A. Liu, and Propagation, "A modified efficient KNN method for antenna optimization and design," *IEEE Transactions on Antennas Propagation,* vol. 68, no. 10, pp. 6858-6866, 2020.
[68] M. Liao, C. Liu, C. Wang, and J. Yang, "Research on a rolling bearing fault detection method with wavelet convolution deep transfer learning," *IEEE Access,* vol. 9, pp. 45175-45188, 2021.
[69] X. Liang, J. Yao, W. Zhang, and Y. Wang, "A novel fault diagnosis of a rolling bearing method based on variational mode decomposition and an artificial neural network," *Applied Sciences,* vol. 13, no. 6, p. 3413, 2023.
[70] A. K. Sharma and N. K. Verma, "Quick learning mechanism with cross-domain adaptation for intelligent fault diagnosis," *IEEE Transactions on Artificial Intelligence,* vol. 3, no. 3, pp. 381-390, 2021.
[71] L. Hou, R. Jiang, Y. Tan, and J. Zhang, "Input feature mappings-based deep residual networks for fault diagnosis of rolling element bearing with complicated dataset," *IEEE Access,* vol. 8, pp. 180967-180976, 2020.
[72] W. Zhang, H. Hao, and Y. Zhang, "State of charge prediction of lithium-ion batteries for electric aircraft with swin transformer," *IEEE/CAA Journal of Automatica Sinica,* vol. 12, no. 3, pp. 645-647, 2024.
[73] W. Zhang, M. Xu, H. Yang, X. Wang, S. Zheng, and X. Li, "Data-driven deep learning approach for thrust prediction of solid rocket motors," *Measurement,* vol. 225, p. 114051, 2024.
[74] X. Li, S. Yu, Y. Lei, N. Li, and B. Yang, "Dynamic vision-based machinery fault diagnosis with cross-modality feature alignment," *IEEE/CAA Journal of Automatica Sinica,* vol. 11, no. 10, pp. 2068-2081, 2024.
[75] X. Li, W. Zhang, X. Li, and H. Hao, "Partial domain adaptation in remaining useful life prediction with incomplete target data," *IEEE/ASME Transactions on Mechatronics,* vol. 29, no. 3, pp. 1903-1913, 2023.
[76] H. Qiu, J. Lee, J. Lin, and G. Yu, "Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics," *Journal of sound and vibration,* vol. 289, no. 4-5, pp. 1066-1090, 2006.

Declaration of interests

☒ 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.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: