Egyptian Informatics Journal 35 (2026) 101038

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

Feature-engineered ensemble learning methods for insulator fault detection based on ultrasound signals

Laio Oriel Seman [a], Stefano Frizzo Stefenon [b,c,d], Valderi Reis Quietinho Leithardt [e,f,c], Juan Francisco De Paz Santana [f,*], Kin-Choong Yow [d]

[a] Department of Automation and Systems Engineering, Federal University of Santa Catarina (UFSC), Florianópolis, SC, Brazil
[b] Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Rua Conselheiro Emídio Navarro 1, 1959-007 Lisboa, Portugal
[c] Center of Technology and Systems (CTS-UNINOVA), Associated Laboratory of Intelligent Systems (LASI), 2829-516, Lisboa, Portugal
[d] Faculty of Engineering and Applied Sciences, University of Regina, Saskatchewan, S4S 0A2, Canada
[e] Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Avenida das Forças Armadas, 1649-026, Lisboa, Portugal
[f] Expert Systems and Applications Lab, Faculty of Science, University of Salamanca, 37008 Salamanca, Spain

ARTICLE INFO

Keywords:
Ensemble learning approach
Reinforcement learning
Signal augmentation
Time series analysis

ABSTRACT

This paper presents a feature-engineered ensemble method for fault detection in medium-voltage insulator ultrasound signals. We develop a feature extraction framework that captures time-domain, frequency-domain, wavelet-domain, and envelope characteristics of ultrasonic emissions from porcelain insulators under various conditions: normal operation, artificial contamination, and simulated lightning damage. A robust signal augmentation methodology enhances classification robustness while preserving class-discriminative properties. We evaluate standard ensemble approaches (random forest, extreme gradient boosting, and light gradient boosting) and introduce two novel algorithms: EvoBagging, which employs evolutionary operators to optimize bootstrap samples, and proximal policy optimization bagging (PPOBagging), which leverages reinforcement learning through proximal policy optimization to construct ensembles via sequential decision-making. Experimental results demonstrate that our feature engineering approach significantly outperforms raw signal classification (91.67% vs. 55.00% accuracy). The proposed PPOBagging algorithm achieves the highest performance with 94.17% accuracy and a macro-F1 score of 0.9412, surpassing conventional methods. Comparative analysis against convolutional kernel transforms (standard, mini, and multi random convolutional kernels) reveals that while these methods offer competitive accuracy (up to 88.33%), they fall short of our feature-engineered ensemble approach in both performance and interpretability. Comparison against deep learning methods also reaffirms the usefulness of the proposed feature engineering technique. This work contributes practical tools for non-destructive monitoring of power distribution systems, enabling proactive maintenance strategies to prevent failures.

1. Introduction

The detection and classification of faults in power grid insulators constitute a fundamental requirement to keep the power transmission and distribution systems running [1]. Insulator failures result in service disruptions, economic losses, and safety risks that affect grid reliability [2]. Traditional inspection methods involve visual assessment or offline testing [3], which are time-consuming, costly, and potentially hazardous [4]. Ultrasonic signal analysis has emerged as a non-destructive technique for real-time monitoring of insulator conditions, capable of detecting partial discharges and contamination before they progress to failure [5].

In electrical power systems, insulators are subjected to various environmental stressors, including pollution [6], moisture [7], and thermal stress [8]. These factors can lead to different types of faults, such as surface contamination, internal cracks, and physical damage [9]. Each fault type may generate distinctive acoustic emissions in the ultrasonic frequency range (20–110 kHz), creating a challenging classification task [10].

Advances in signal processing and machine learning (ML) have enhanced the capabilities of ultrasonic fault detection systems [11]. While deep learning (DL) approaches have shown effectiveness in this domain [12], they demand substantial computational resources and

* Corresponding author.
E-mail addresses: laio.seman@ufsc.br (L.O. Seman), stefano.stefenon@isel.pt (S.F. Stefenon), valderi.leithardt@iscte-iul.pt (V.R.Q. Leithardt), fcofds@usal.es (J. F. De Paz Santana), kin-choong.yow@uregina.ca (K.-C. Yow).

https://doi.org/10.1016/j.eij.2026.101038
Received 21 April 2025; Received in revised form 13 April 2026; Accepted 2 August 2026
Available online 11 August 2026
1110-8665/© 2026 The Authors. Published by Elsevier B.V. on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

lack interpretability [13]. Ensemble methods present an alternative, offering classification performance with improved efficiency and transparency. The combination of multiple base learners through bagging, boosting, or hybrid techniques has demonstrated robust performance in classification tasks, particularly when applied to structured feature representations [14].

Based on these challenges, this paper investigates the application of feature-engineered ensemble methods for fault detection in insulator ultrasound signals. We propose a feature extraction methodology that captures time-domain, frequency-domain, wavelet-domain, and envelope characteristics to provide discriminative representations of ultrasound signals. The extracted features include statistical moments, spectral power distribution, wavelet coefficients, and instantaneous frequency metrics, which collectively characterize the complex patterns associated with different insulator conditions.

We examine the effectiveness of various ensemble learning strategies, including conventional approaches such as random forest (RandomForest) [15], extreme gradient boosting (XGBoost) [16], and light gradient boosting (LightGBM) [17], alongside novel evolutionary algorithms developed for this application. We introduce two specialized ensemble methods: EvoBagging [18], which employs evolutionary operators to optimize bootstrap sample composition, and PPOBagging [19], which utilizes reinforcement learning through proximal policy optimization to construct and refine ensembles via sequential decision-making.

The research methodology employs a controlled dataset using pin-type porcelain insulators under various conditions: normal operational state, artificially contaminated according to standardized protocols, and physically damaged to simulate lightning strikes. The objectives of this research are to: establish a feature engineering pipeline for ultrasound signal classification, evaluate the performance of diverse ensemble learning methods in this context, develop and validate novel ensemble algorithms for improving classification accuracy, and compare the proposed methodology against modern convolutional kernel transforms classifiers such as random convolutional kernels (ROCKET) [20], MiniROCKET [21], and MultiROCKET [22] algorithms.

This work contributes to developing reliable, interpretable fault detection systems for power infrastructure monitoring. The proposed methodology enables early detection of insulator degradation, potentially reducing maintenance costs and preventing catastrophic failures in distribution networks. The feature engineering approach also provides interpretable insights into fault characteristics, addressing a key limitation of end-to-end DL methods currently employed in this domain.

In this paper, errors and bias in ultrasonic signals are mitigated through a combination of robust feature engineering and data augmentation. The feature extraction process captures discriminative patterns across time, frequency, wavelet, and envelope domains, reducing sensitivity to random noise and measurement inconsistencies. A comprehensive augmentation strategy introduces temporal shifts, amplitude scaling, noise injection, and frequency masking, enhancing model robustness and minimizing overfitting. Selective augmentation is also applied to balance class distributions, reducing bias toward majority classes.

The paper is structured as follows: Section 2 reviews the relevant literature on fault detection in electrical systems, with emphasis on ultrasonic signal analysis and machine learning approaches. Section 3 describes the dataset and experimental setup, detailing the acquisition of ultrasound signals under different insulator conditions. Section 4 presents the proposed methodology, including the feature extraction process, data augmentation strategies, and ensemble learning framework. Section 5 reports the experimental results and provides a comprehensive comparative analysis of the proposed approach against existing methods. Finally, Section 6 concludes the paper by summarizing the main findings and outlining potential directions for future research.

2. Related works

There is a growing trend toward data analysis using ML techniques to improve electrical system inspection [23]. Several authors have proposed modifications to detection models to improve fault identification capabilities [24–26]. For instance, Zhao et al. [27] used a region-based convolutional neural network (R-CNN), Wang et al. [28] used a mask R-CNN, and He et al. [29] proposed a modified object detection method for insulator fault identification.

Alshalawi et al. [30] presented the development and evaluation of three convolutional neural network (CNN) models designed to detect and classify four types of partial discharge faults, arcing, corona discharge, tracking, and looseness, as well as distinguish healthy electrical equipment. The models demonstrated strong performance, achieving over 94% accuracy even when tested on noisy real-world data collected from field failures. This high accuracy highlights the robustness and practical applicability of the proposed models in diagnosing partial discharge-related issues in electrical equipment, despite the complexity and variability of real-world conditions.

The use of non-destructive techniques is becoming increasingly popular for carrying out inspections on electrical power grids, due to their flexibility and the fact that they do not require direct contact with the network [31]. In addition to techniques based on convolutional networks, other models have been used to improve the identification of faults in the electrical system, such as through the use of specific equipment like infrared imaging [32], ultraviolet cameras [33], or ultrasound.

Ultrasound signal analysis has been adopted for detecting faults in high and medium-voltage insulators, being a promising approach to assist in inspections of the electrical system, thus improving its reliability. In [34] a hybrid wavelet-long short-term memory model was proposed to predict faults in contaminated insulators using ultrasonic emissions, achieving superior results over other DL techniques. In [35], neural networks were considered to classify the condition of insulators based on ultrasound recordings, validating the feasibility of non-destructive monitoring.

Ferreira et al. [36] proposed a method for estimating pollution levels on electrical insulators by analyzing the ultrasonic noise they emit when energized. The approach leverages the Spectral subband centroid energy vectors algorithm to extract key frequency features from the noise signals, compressing the spectral data to highlight the most relevant information. These processed signals form a dataset used to train an artificial neural network, which can accurately distinguish between different pollution levels. To validate the method, field tests were conducted at the Campina Grande substation, where the technique demonstrated high reliability, achieving success rates above 80% in classifying insulator contamination levels.

The integration of ML techniques into ultrasound signal processing has contributed to advancing both qualitative and quantitative analysis in diagnostic systems. Luijten et al. [37] provided a foundational survey on the evolution of ultrasound signal processing from classical model-based frameworks to data-driven approaches, emphasizing the increasing adoption of DL for robust feature representation. In a recent effort, Song et al. [38] highlighted the challenges associated with conventional signal processing pipelines and advocated for hybrid architectures combining model-based insights with deep neural networks to enhance performance across diverse scenarios.

Wang et al. [39] presented a nondestructive method to detect micron-sized kissing defects in composite insulators used in power grids. By establishing a cylindrical adhesive contact model, the researchers analyze how these defects distort ultrasonic wave propagation at the polymeric interface. Using nonlinear ultrasonic testing, they measure the ultrasonic nonlinearity parameter in both defective and intact samples. The experimental results confirm the theoretical model, demonstrating that kissing defects as small as 0.7 μm can be identified using this method.

Table 1
Summary of reviewed studies on ML-based fault detection in electrical systems.

Study	Method	Data modality	Main findings
[27]	Faster R-CNN	Visual (images)	Improved insulator recognition and fault detection in transmission lines under varying scales and occlusions.
[28]	Mask R-CNN	Infrared/Visual	Effective automatic extraction and diagnosis of insulator defects with high accuracy and speed.
[29]	Modified object detection	Visual (images)	Enhanced multi-fault detection in complex backgrounds.
[30]	Three CNN models	Partial discharge signals	Over 94% accuracy in classifying arcing, corona, tracking, looseness vs. healthy; robust on noisy field data.
[34]	Hybrid wavelet-LSTM	Ultrasonic emissions	Superior fault prediction in contaminated insulators compared to other DL techniques.
[35]	ANN (Perceptrons)	Ultrasound recordings	Feasible non-destructive classification of insulator conditions.
[36]	Spectral subband centroid + ANN	Ultrasonic noise	Over 80% success in classifying pollution levels; validated in field substation tests.
[39]	Nonlinear ultrasonic testing + contact model	Ultrasonic waves	Detection of micron-sized kissing defects ($\geq 0.7\,\mu m$) in composite insulators.
[40]	Wavelet + LDA + Sparrow opt. + BPNN	Ultrasonic signals	Improved sensitivity and categorization of internal defects (bubbles, cracks).
[45]	Ensemble of 7 pretrained CNNs (soft voting)	Visual (images)	Outperforms individual CNNs for robust insulator condition monitoring.
[46]	Three-plus-one stacking ensemble	Thermal/unsteady-state data	Superior prediction accuracy and generalization of heat transfer coefficient.

Liu et al. [40] introduced an ultrasonic testing platform designed to detect internal defects, such as bubbles and cracks, that often form during the production of insulation components. The study addresses the challenge of isolating weak acoustic signals, which are crucial for accurately distinguishing different defect types. This is achieved by applying discrete wavelet decomposition to extract both time and frequency domain information, followed by using linear discriminant analysis to reduce feature dimensions. Based on these features, a multi-stage diagnostic model is proposed that synergistically integrates a sparrow optimization algorithm with a backpropagation neural network. This approach enhances the sensitivity of the ultrasonic testing method and also robustly categorizes various defect types and morphological characteristics. DL has emerged as a natural fit for multi-label learning problems, primarily due to its capability to model intricate label correlations and extract high-level abstractions [41], currently being used by several authors [42–44].

Beyond DL, ensemble learning methods have also been explored for fault detection. Moshrefi and Nabki [47] demonstrated the effectiveness of combining ultrasonic signal features with ensemble classifiers (e.g., gradient boosting) for improved industrial fault detection accuracy. Innovative signal processing strategies, such as ultrasonic guided waves, have been proposed by Zhang et al. [48] for detecting internal discharge cracks in composite insulators using pulse-echo and reflection analysis. In [49], a wavelet neuro-fuzzy system was introduced for insulator fault forecasting, offering interpretability along with robust predictive power.

Given that insulators are prone to failure due to environmental pollution and operational stress, ensemble CNNs may support regular image-based inspections to prevent outages. Serikbay et al. [45] proposed an ensemble classifier by combining the outputs of a selected subset of seven pretrained CNN architectures through soft voting. The ensemble leverages the diverse feature extraction capabilities of each model, leading to improved classification performance. Numerical experiments confirm that this ensemble outperforms individual models, highlighting its potential for robust condition monitoring in power systems.

Li et al. [46] presented a method for detecting the thermal insulation performance of vacuum glass using ensemble learning techniques. The authors propose a three-plus-one stacking model based on unsteady-state data, using temperature-related variables and the heat transfer coefficient as inputs and targets. By combining multiple machine learning models through cross-validation and Spearman correlation analysis, the approach improves prediction accuracy and model generalization. Experimental validation involving ten single models and eleven ensemble models demonstrates that the proposed stacking model outperforms all others in predicting the heat transfer coefficient.

Table 1 summarizes the key reviewed studies, highlighting their primary methods, data modalities, and main findings for easier comparison. Most vision-based studies rely on CNNs or object detectors for surface-level defects, while ultrasound-based works employ either basic neural networks, simple wavelet-artificial neural network (ANN) pipelines, or model-based nonlinear testing, typically limited to single-domain features or raw signals. Although ensemble techniques have shown promise in other domains (e.g., image ensembles or thermal insulation), no prior study has combined comprehensive multi-domain feature engineering (time, frequency, wavelet, and envelope domains) with advanced ensemble learning specifically tailored to ultrasonic emissions from medium-voltage porcelain insulators.

Evolutionary optimization of bootstrap samples (EvoBagging) and reinforcement-learning-driven ensemble construction (PPOBagging) have not been explored in this context. This work addresses these gaps by proposing a complete feature-engineered ensemble framework that extracts 50 discriminative features from ultrasound signals under realistic operating conditions (normal, artificially contaminated, and lightning-damaged insulators).

3. Dataset

The dataset utilized in this study consists of ultrasonic recordings from medium voltage insulators under various conditions. The recordings were obtained from a controlled laboratory experiment where pin-type porcelain insulators (class 15 kV, manufactured by Germer) were subjected to a phase-to-ground voltage of 7.95 kV, equivalent to 13.8 kV phase-to-phase in distribution power systems. This condition is equivalent to how insulators are found in the field, in power distribution grids in interior regions of southern Brazil [50]. Fig. 1(a) presents the medium-voltage experiment setup used in this paper.

In this paper, three distinct insulator conditions were analyzed: insulators in good condition, artificially contaminated insulators, and drilled insulators simulating damage from lightning strikes. The contamination procedure followed the IEC-60507 protocols for artificial pollution tests on power grid insulators [51], utilizing kaolin and sodium chloride as contaminants. The drilling was performed using a bench drill to replicate perforations typically caused by electrical discharges (see Fig. 1(b).)

Data acquisition was conducted in an acrylic chamber to minimize external noise interference, creating an artificially contaminated environment. The ultrasound detector (Petterson M500 model) was used for recording the signal generated by the insulators. The detector was

positioned at a fixed distance of 0.4 m from the insulators. This detector captures noise signals with frequencies up to 500 kHz, ensuring comprehensive coverage of the partial discharge frequency range. To ensure sufficient data for analysis, recordings were conducted for 50 s per sample, yielding approximately 100,000 data points per recording, considering a sampling period of 5×10^{-4} seconds.

The experimental setup employed non-conductive materials for mounting to prevent interference from partial discharges at connection points. During testing, a water spray with conductivity of 56 kg/m^3 was applied to simulate medium-high contamination levels according to IEC standards [51]. This dataset provides a realistic representation of insulator conditions encountered in southern Brazilian distribution networks, where these components are exposed to both organic contamination from unpaved roads and saline contamination in coastal areas.

4. Methodology

This section presents a structured framework for ultrasound signal classification that integrates feature engineering, data augmentation, and ensemble learning. Initially, a comprehensive feature extraction process is applied to capture time-domain, frequency-domain, wavelet-based, and envelope characteristics, forming a unified and discriminative feature vector. To enhance robustness and mitigate class imbalance, a data augmentation strategy introduces controlled temporal, amplitude, spectral, and envelope transformations, combined through a compositional approach with validation mechanisms. The framework employs ensemble learning methods, including bagging, boosting, and advanced approaches such as EvoBagging and PPOBagging, to improve classification accuracy and generalization, resulting in a robust and interpretable fault detection pipeline.

4.1. Feature extraction methodology

The feature extraction process for signal analysis encompasses a set of time-domain, frequency-domain [52], wavelet-domain [53], and envelope characteristics. Given a discrete ultrasound signal window $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ sampled at frequency f_s, we extract a multidimensional feature vector $\mathbf{F}$ through the following methodology.

4.1.1. Time-domain feature extraction

The initial phase of our feature engineering involves the computation of statistical moments and time-domain characteristics to capture the amplitude distribution and temporal dynamics of the ultrasound signal:

$$\begin{aligned}
\mu_{\mathbf{x}} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\
\sigma_{\mathbf{x}} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_{\mathbf{x}})^2} \\
\gamma_{\mathbf{x}} &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}\right)^3 \\
\kappa_{\mathbf{x}} &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}\right)^4 - 3
\end{aligned} \tag{1}$$

where $\mu_{\mathbf{x}}$, $\sigma_{\mathbf{x}}$, $\gamma_{\mathbf{x}}$, and $\kappa_{\mathbf{x}}$ represent the mean, standard deviation, skewness, and kurtosis of the signal window, respectively.

Additional statistical descriptors are computed to characterize the amplitude distribution:

$$\begin{aligned}
x_{\max} &= \max(\mathbf{x}) \\
x_{\min} &= \min(\mathbf{x}) \\
x_{\text{med}} &= \text{median}(\mathbf{x}) \\
x_{25} &= \mathcal{P}_{25}(\mathbf{x}) \\
x_{75} &= \mathcal{P}_{75}(\mathbf{x})
\end{aligned} \tag{2}$$

where $\mathcal{P}_q(\mathbf{x})$ denotes the qth percentile of the signal values.

To characterize signal variability and energy content, we compute:

$$\begin{aligned}
\Delta_{\mathbf{x}} &= \frac{1}{n-1}\sum_{i=1}^{n-1}|x_{i+1}-x_i| \\
\text{RMS}_{\mathbf{x}} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2} \\
E_{\mathbf{x}} &= \sum_{i=1}^{n}|x_i|
\end{aligned} \tag{3}$$

where $\Delta_{\mathbf{x}}$ represents the mean absolute change, $\text{RMS}_{\mathbf{x}}$ is the root mean square value, and $E_{\mathbf{x}}$ denotes the signal energy.

4.1.2. Signal transition analysis

To characterize signal oscillations and structural properties, we analyze signal transitions and peak characteristics:

$$\text{ZCR}_{\mathbf{x}} = \frac{1}{n-1}\sum_{i=1}^{n-1}\mathbb{1}\{x_i \cdot x_{i+1} < 0\} \tag{4}$$

where $\text{ZCR}_{\mathbf{x}}$ is the zero-crossing rate and $\mathbb{1}\{\cdot\}$ is the indicator function that equals 1 when the condition is satisfied and 0 otherwise.

For peak analysis, we identify the set of peak indices $\mathcal{P}$ defined as:

$$\mathcal{P} = \{i \in \{2,3,\ldots,n-1\} : x_i > x_{i-1} \wedge x_i > x_{i+1}\}. \tag{5}$$

From this set, we derive:

$$\begin{aligned}
N_{\text{peaks}} &= |\mathcal{P}| \\
\mu_{\text{peak}} &= \begin{cases} \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} x_i & \text{if } |\mathcal{P}| > 0 \\ 0 & \text{otherwise} \end{cases} \\
\sigma_{\text{peak}} &= \begin{cases} \sqrt{\frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(x_i-\mu_{\text{peak}})^2} & \text{if } |\mathcal{P}| > 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{6}$$

where N_{peaks}, μ_{peak}, and σ_{peak} represent the number of peaks, mean peak amplitude, and standard deviation of peak amplitudes, respectively.

4.1.3. Frequency-domain analysis

For spectral characterization, we employ Welch's method [54] to estimate the power spectral density (PSD) [55] of the signal:

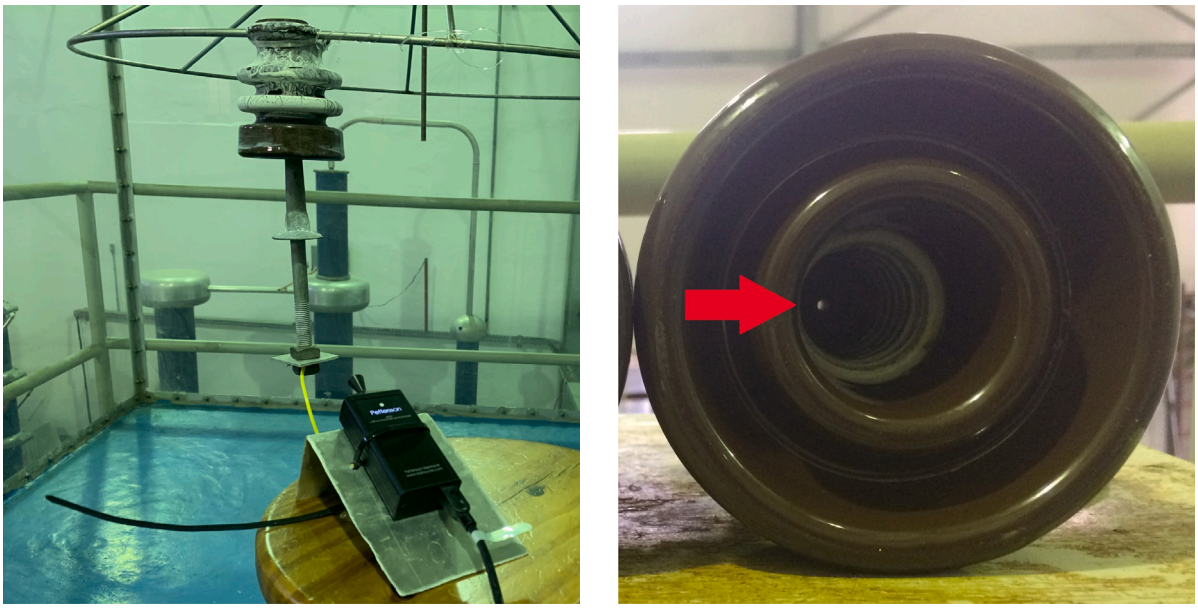
$$\hat{P}_{xx}(f) = \frac{1}{KL}\sum_{i=0}^{K-1}\left|\sum_{n=0}^{L-1} w(n)x_{i\Delta+n}e^{-j2\pi fn/f_s}\right|^2 \tag{7}$$

where $w(n)$ is the window function, L is the segment length, K is the number of segments, and Δ is the segment offset. We partition the frequency domain into M distinct bands $\{(f_i^{\text{low}}, f_i^{\text{high}})\}_{i=1}^{M}$ that encompass key ultrasound frequency ranges: {(10 kHz, 30 kHz), (30 kHz, 50 kHz), (50 kHz, 75 kHz), (75 kHz, 100 kHz), (100 kHz, 150 kHz), and (150 kHz, 210 kHz)}.

For each frequency band i, we compute the band power P_i and relative band power R_i:

$$\begin{aligned}
P_i &= \int_{f_i^{\text{low}}}^{f_i^{\text{high}}} \hat{P}_{xx}(f)\,df \\
&\approx \sum_{f_i^{\text{low}} \le f_k \le f_i^{\text{high}}} \hat{P}_{xx}(f_k)\Delta f \\
R_i &= \frac{P_i}{\int_0^{f_s/2}\hat{P}_{xx}(f)\,df} = \frac{P_i}{\sum_k \hat{P}_{xx}(f_k)\Delta f}
\end{aligned} \tag{8}$$

where Δf is the frequency resolution of the PSD estimate, and the summation is performed over all discrete frequency bins f_k that fall within the specified band.



(a) (b)

Fig. 1. Experimental setup and insulator comparison: (a) medium-voltage experiment setup; (b) perforated insulator.

Additional spectral features are computed to characterize the frequency distribution:

$$\begin{aligned} H_{\text{spec}} &= -\sum_k \hat{p}_{xx}(f_k)\log_2 \hat{p}_{xx}(f_k) \\ f_{\text{dom}} &= \operatorname*{argmax}_f \hat{P}_{xx}(f) \\ f_{\text{centroid}} &= \frac{\int f\cdot \hat{P}_{xx}(f)\,df}{\int \hat{P}_{xx}(f)\,df} \approx \frac{\sum_k f_k \hat{P}_{xx}(f_k)}{\sum_k \hat{P}_{xx}(f_k)} \\ \text{BW}_{\text{spec}} &= \sqrt{\frac{\int (f-f_{\text{centroid}})^2\cdot \hat{P}_{xx}(f)\,df}{\int \hat{P}_{xx}(f)\,df}} \end{aligned} \tag{9}$$

where $\hat{p}_{xx}(f_k) = \hat{P}_{xx}(f_k)/\sum_k \hat{P}_{xx}(f_k)$ is the normalized PSD, H_{spec} is the spectral entropy, f_{dom} is the dominant frequency, f_{centroid} is the spectral centroid, and BW_{spec} is the spectral bandwidth.

4.1.4. Wavelet decomposition

To capture multi-resolution time-frequency characteristics, we apply a wavelet decomposition using the Daubechies wavelet [56] of order 4 (db4) to a maximum decomposition level $J = 4$:

$$\mathbf{x} = \sum_k a_{J,k}\phi_{J,k}(t) + \sum_{j=1}^{J}\sum_k d_{j,k}\psi_{j,k}(t) \tag{10}$$

where $\phi_{j,k}(t)$ and $\psi_{j,k}(t)$ are the scaling and wavelet functions, respectively, and $a_{J,k}$ and $d_{j,k}$ are the approximation and detail coefficients.

For each decomposition level $j \in \{1, 2, \ldots, J\}$ and the final approximation coefficients, we compute:

$$\begin{aligned} \mu_{\text{wavelet},j} &= \frac{1}{N_j}\sum_k c_{j,k} \\ \sigma_{\text{wavelet},j} &= \sqrt{\frac{1}{N_j}\sum_k (c_{j,k}-\mu_{\text{wavelet},j})^2} \\ H_{\text{wavelet},j} &= -\sum_k \hat{c}_{j,k}\log_2 \hat{c}_{j,k} \end{aligned} \tag{11}$$

where $c_{j,k}$ represents either approximation coefficients $a_{J,k}$ (when $j = J+1$) or detail coefficients $d_{J+1-j,k}$ (when $j \le J$), N_j is the number of coefficients at level j, and $\hat{c}_{j,k} = |c_{j,k}|/\sum_k |c_{j,k}|$ is the normalized absolute coefficient value.

4.1.5. Hilbert transform analysis

To analyze the envelope and instantaneous properties of the signal, we apply the Hilbert transform [57] to obtain the analytic signal $z(t)$:

$$z(t) = x(t) + j\mathcal{H}\{x(t)\} = A(t)e^{j\phi(t)} \tag{12}$$

where $\mathcal{H}\{x(t)\}$ is the Hilbert transform of $x(t)$, $A(t)$ is the instantaneous amplitude (envelope), and $\phi(t)$ is the instantaneous phase.

From the analytic signal, we derive the following features:

$$\begin{aligned} \mu_A &= \frac{1}{n}\sum_{i=1}^{n} A_i \\ \sigma_A &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A_i-\mu_A)^2} \\ f_{\text{inst}}(t) &= \frac{1}{2\pi}\frac{d\phi(t)}{dt} \approx \frac{f_s}{2\pi}\cdot[\phi(t+\Delta t)-\phi(t)] \\ \mu_{f_{\text{inst}}} &= \frac{1}{n-1}\sum_{i=1}^{n-1} f_{\text{inst},i} \\ \sigma_{f_{\text{inst}}} &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1}(f_{\text{inst},i}-\mu_{f_{\text{inst}}})^2} \end{aligned} \tag{13}$$

where μ_A and σ_A are the mean and standard deviation of the amplitude envelope, and $\mu_{f_{\text{inst}}}$ and $\sigma_{f_{\text{inst}}}$ are the mean and standard deviation of the instantaneous frequency.

4.1.6. Feature vector construction

The complete feature vector $\mathbf{F}$ is constructed by concatenating all computed features:

$$\begin{aligned} \mathbf{F} = [&\mu_{\mathbf{x}}, \sigma_{\mathbf{x}}, \gamma_{\mathbf{x}}, \kappa_{\mathbf{x}}, x_{\max}, x_{\min}, x_{\text{med}}, x_{25}, x_{75}, \Delta_{\mathbf{x}}, \text{RMS}_{\mathbf{x}}, E_{\mathbf{x}}, \\ &\text{ZCR}_{\mathbf{x}}, N_{\text{peaks}}, \mu_{\text{peak}}, \sigma_{\text{peak}}, \{P_i, R_i\}_{i=1}^{M}, \\ &H_{\text{spec}}, f_{\text{dom}}, f_{\text{centroid}}, \text{BW}_{\text{spec}}, \\ &\{\mu_{\text{wavelet},j}, \sigma_{\text{wavelet},j}, H_{\text{wavelet},j}\}_{j=1}^{J+1}, \\ &\mu_A, \sigma_A, \mu_{f_{\text{inst}}}, \sigma_{f_{\text{inst}}}]. \end{aligned} \tag{14}$$

This comprehensive feature vector captures the essential time-domain, frequency-domain, time-frequency, and instantaneous characteristics of the ultrasound signal, providing a robust representation for subsequent classification and analysis tasks.

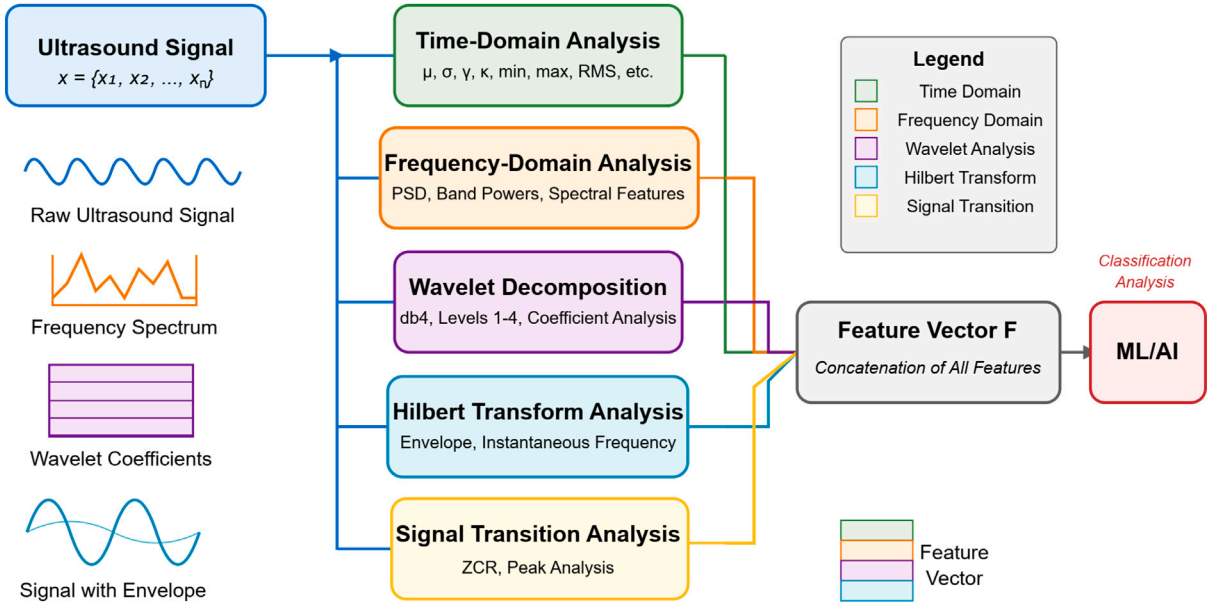


Fig. 2. Pipeline of the proposed method.

In cases where certain transforms fail due to numerical instability or insufficient data, placeholder values of zero are substituted to maintain consistent dimensionality of the feature vector across all signal windows. This approach ensures that the feature extraction process is robust to varying signal conditions while preserving the integrity of the feature space for ML applications. The complete pipeline of the proposed method is presented in Fig. 2.

4.2. Signal augmentation methodology

To enhance the robustness and generalizability of our ultrasound signal classification framework, we implement a comprehensive data augmentation strategy. This methodology systematically introduces controlled variations to the original signals while preserving their class-discriminative characteristics. Let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ denote an original ultrasound signal sampled at frequency f_s. We define a set of transformation functions $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K\}$ that generate augmented variants of the signal.

4.2.1. Temporal transformations

For the time shifting, we implement a temporal displacement transformation parameterized by a shift factor $\alpha \in [\alpha_{\min}, \alpha_{\max}]$:

$$\mathcal{T}_{\text{shift}}(\mathbf{x}, \alpha) = \mathbf{x}' \text{ where } x'_i = \begin{cases} x_{i-\delta} & \text{if } \delta \le i \le n \\ 0 & \text{if } 1 \le i < \delta \end{cases} \tag{15}$$

for $\delta = \lfloor \alpha \cdot n \rfloor$ and $\alpha \sim \mathcal{U}(\alpha_{\min}, \alpha_{\max})$. Typically, $\alpha_{\min} = -0.1$ and $\alpha_{\max} = 0.1$, allowing for shifts up to 10% of the signal length in either direction.

For time stretching, we apply a temporal scaling transformation that alters the signal's duration while preserving its amplitude characteristics:

$$\mathcal{T}_{\text{stretch}}(\mathbf{x}, \beta) = \text{resample}(\text{resample}(\mathbf{x}, \lfloor \beta \cdot n \rfloor), n) \tag{16}$$

where $\beta \sim \mathcal{U}(\beta_{\min}, \beta_{\max})$ with typical values $\beta_{\min} = 0.8$ and $\beta_{\max} = 1.2$. The nested resampling operation first stretches or compresses the signal by factor β and then restores it to the original length n.

4.2.2. Amplitude transformations

To simulate measurement noise and environmental interference, we introduce additive Gaussian noise scaled relative to the signal's energy:

$$\mathcal{T}_{\text{noise}}(\mathbf{x}, \gamma) = \mathbf{x} + \gamma \cdot \sigma_{\mathbf{x}} \cdot \epsilon \tag{17}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I}_n)$ is a vector of independent standard normal random variables, $\sigma_{\mathbf{x}}$ is the standard deviation of the original signal, and $\gamma \sim \mathcal{U}(\gamma_{\min}, \gamma_{\max})$ with typical values $\gamma_{\min} = 0.001$ and $\gamma_{\max} = 0.05$.

For colored noise addition, we incorporate spectrally shaped noise to simulate various environmental conditions:

$$\mathcal{T}_{\text{colored}}(\mathbf{x}, \eta, \nu) = \mathbf{x} + \eta \cdot \sigma_{\mathbf{x}} \cdot C_\nu(\epsilon) \tag{18}$$

where $C_\nu(\epsilon)$ denotes the spectral coloring operation with parameter ν, implemented in the frequency domain as:

$$C_\nu(\epsilon) = \mathcal{F}^{-1}\{\mathcal{F}\{\epsilon\} \cdot \mathbf{f}^\nu\} \tag{19}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the discrete Fourier transform and its inverse, $\mathbf{f}$ represents the frequency vector, and the spectral exponent ν determines the noise color: $\nu = -0.5$ for pink noise, $\nu = -1.0$ for brown noise, and $\nu = 0.5$ for blue noise. The scaling factor $\eta \sim \mathcal{U}(\eta_{\min}, \eta_{\max})$ with typical values $\eta_{\min} = 0.001$ and $\eta_{\max} = 0.05$.

Magnitude warping. We implement a non-linear amplitude modulation using a smooth warping function:

$$\mathcal{T}_{\text{warp}}(\mathbf{x}, \mathbf{c}) = \mathbf{x} \odot \mathcal{W}(\mathbf{t}; \mathbf{c}) \tag{20}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{t} = \{t_i = (i-1)/(n-1) : i \in \{1, 2, \ldots, n\}\}$ is the normalized time vector, and $\mathcal{W}(\mathbf{t}; \mathbf{c})$ is a cubic spline interpolation function with control points $\mathbf{c} = \{(p_j, q_j)\}_{j=1}^m$. The knot positions $\{p_j\}_{j=1}^m$ are uniformly spaced in $[0, 1]$, and the corresponding values $q_j \sim \mathcal{U}(q_{\min}, q_{\max})$ with typical values $q_{\min} = 0.8$ and $q_{\max} = 1.2$.

4.2.3. Spectral transformations

For bandpass filtering, we apply random bandpass filtering to simulate varying channel characteristics:

$$\mathcal{T}_{\text{filter}}(\mathbf{x}, f_{\text{low}}, f_{\text{high}}, o) = \text{filtfilt}(B_o(f_{\text{low}}, f_{\text{high}}), A_o(f_{\text{low}}, f_{\text{high}}), \mathbf{x}) \tag{21}$$

where $B_o(f_{\text{low}}, f_{\text{high}})$ and $A_o(f_{\text{low}}, f_{\text{high}})$ are the numerator and denominator filter coefficients of a Butterworth bandpass filter of order

o, with normalized cutoff frequencies $f_{\text{low}}/f_{\text{Nyquist}}$ and $f_{\text{high}}/f_{\text{Nyquist}}$. We sample $f_{\text{low}} \sim \mathcal{U}(f_{\min}, f_{\max})$, $f_{\text{high}} \sim \mathcal{U}(f_{\text{low}}/f_{\text{Nyquist}}, f_{\max})$, and $o \sim \mathcal{U}\{2,3,4\}$, with typical values $f_{\min} = 0.05 \cdot f_{\text{Nyquist}}$ and $f_{\max} = 0.95 \cdot f_{\text{Nyquist}}$.

For frequency masking, we implement spectral gaps to simulate frequency-selective fading:

$$\mathcal{T}_{\text{freq_mask}}(\mathbf{x}, \{(s_j, w_j)\}_{j=1}^{k}) = \mathcal{F}^{-1}\{\mathbf{X} \odot \mathbf{M}\} \tag{22}$$

where $\mathbf{X} = \mathcal{F}\{\mathbf{x}\}$ is the Fourier transform of the signal, and $\mathbf{M}$ is a binary masking vector defined as:

$$M_i = \begin{cases} 0 & \text{if } s_j \le i \le s_j + w_j \text{ for any } j \in \{1,2,\dots,k\} \\ 1 & \text{otherwise} \end{cases} \tag{23}$$

for $i \in \{1,2,\dots,\lfloor n/2 \rfloor + 1\}$. The starting indices $s_j \sim \mathcal{U}\{1, \lfloor n/2 \rfloor + 1 - w_j\}$ and width parameters $w_j = \lfloor \omega_j \cdot (\lfloor n/2 \rfloor + 1) \rfloor$ with $\omega_j \sim \mathcal{U}(\omega_{\min}, \omega_{\max})$. Typical values are $\omega_{\min} = 0.05$, $\omega_{\max} = 0.2$, and $k = 2$.

For phase shifting, we modify the phase characteristics of the signal while preserving its amplitude envelope:

$$\mathcal{T}_{\text{phase}}(\mathbf{x}, \phi) = \text{Re}\{z(\mathbf{x}) \cdot e^{i\phi}\} \tag{24}$$

where $z(\mathbf{x}) = \mathbf{x} + i\mathcal{H}\{\mathbf{x}\}$ is the analytic signal obtained via the Hilbert transform, $\phi \sim \mathcal{U}(\phi_{\min}, \phi_{\max})$ is the phase shift parameter with typical values $\phi_{\min} = -\pi/4$ and $\phi_{\max} = \pi/4$, and $\text{Re}\{\cdot\}$ extracts the real component.

4.2.4. Signal envelope transformations

We modify the amplitude envelope while preserving the instantaneous phase:

$$\mathcal{T}_{\text{env}}(\mathbf{x}, \rho) = A(\mathbf{x}) \cdot \rho \cdot \cos(\boldsymbol{\Phi}(\mathbf{x})) \tag{25}$$

where $A(\mathbf{x}) = |z(\mathbf{x})|$ is the instantaneous amplitude (envelope), $\boldsymbol{\Phi}(\mathbf{x}) = \arg(z(\mathbf{x}))$ is the instantaneous phase, and $\rho \sim \mathcal{U}(\rho_{\min}, \rho_{\max})$ is the envelope scaling factor with typical values $\rho_{\min} = 0.8$ and $\rho_{\max} = 1.2$.

We simulate transient signal loss by applying a temporal mask:

$$\mathcal{T}_{\text{dropout}}(\mathbf{x}, s, w, p) = \begin{cases} \mathbf{x} \odot \mathbf{m}(s, w) & \text{with probability } p \\ \mathbf{x} & \text{with probability } 1 - p \end{cases} \tag{26}$$

where $\mathbf{m}(s, w)$ is a binary masking vector defined as:

$$m_i(s, w) = \begin{cases} 0 & \text{if } s \le i \le s + w \\ 1 & \text{otherwise} \end{cases} \tag{27}$$

for $i \in \{1,2,\dots,n\}$. The starting index $s \sim \mathcal{U}\{\lfloor n/4 \rfloor, \lfloor 3n/4 \rfloor\}$, width parameter $w \sim \mathcal{U}\{\lfloor n/20 \rfloor, \lfloor n/10 \rfloor\}$, and dropout probability $p = 0.2$.

4.2.5. Compositional augmentation methodology

To generate diverse yet realistic augmented signals, we employ a compositional augmentation strategy that applies a sequence of randomly selected transformations. Given an original signal $\mathbf{x}$, the augmented signal $\mathbf{x}^{\text{aug}}$ is generated by:

$$\mathbf{x}^{\text{aug}} = (\mathcal{T}_{i_d} \circ \mathcal{T}_{i_{d-1}} \circ \cdots \circ \mathcal{T}_{i_1})(\mathbf{x}) \tag{28}$$

where $\{\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \dots, \mathcal{T}_{i_d}\}$ is a subset of d transformation functions randomly sampled without replacement from the set of available transformations. The number of transformations d is a parameter that controls the augmentation intensity, with typical values $d \in \{1,2,3\}$.

4.2.6. Dataset balancing via selective augmentation

For imbalanced datasets where certain classes $C = \{c_1, c_2, \dots, c_L\}$ have different representation counts $\{n_1, n_2, \dots, n_L\}$, we employ a selective augmentation strategy to achieve balanced class distributions. For each class c_l, we generate $\max(0, n_{\max} - n_l)$ additional synthetic samples, where $n_{\max} = \max_{1 \le l \le L} n_l$ is the count of the most frequent class. The augmentation process is defined as:

$$D^{\text{aug}} = D \cup \bigcup_{l=1}^{L} \{(\mathcal{A}(\mathbf{x}_{l,j}), c_l) : j \in S_l, |S_l| = \max(0, n_{\max} - n_l)\} \tag{29}$$

where D is the original dataset, $\mathbf{x}_{l,j}$ denotes the jth signal from class c_l, S_l is a set of indices sampled with replacement from the range $\{1,2,\dots,n_l\}$, and $\mathcal{A}$ is the augmentation function that applies a sequence of random transformations:

$$\mathcal{A}(\mathbf{x}) = \begin{cases} (\mathcal{T}_{i_d} \circ \mathcal{T}_{i_{d-1}} \circ \cdots \circ \mathcal{T}_{i_1})(\mathbf{x}) & \text{if successful} \\ \mathcal{T}_{\text{noise}}(\mathbf{x}, \gamma_{\text{low}}) & \text{as fallback 1} \\ \mathbf{x} & \text{as fallback 2} \end{cases} \tag{30}$$

where $\gamma_{\text{low}} \sim \mathcal{U}(0.001, 0.01)$ represents a conservative noise level used as a fallback augmentation strategy when more complex transformations fail.

4.2.7. On-the-fly batch augmentation

To further enhance generalization during model training, we implement an on-the-fly batch augmentation strategy that dynamically generates augmented samples during the training process. For each mini-batch $\mathcal{B} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_b, y_b)\}$, we apply stochastic augmentation:

$$\mathcal{B}^{\text{aug}} = \{(\mathbf{x}_i^{\text{aug}}, y_i) : i \in \{1,2,\dots,b\}\} \tag{31}$$

where

$$\mathbf{x}_i^{\text{aug}} = \begin{cases} \mathcal{A}(\mathbf{x}_i) & \text{with probability } p_{\text{aug}} \\ \mathbf{x}_i & \text{with probability } 1 - p_{\text{aug}}. \end{cases} \tag{32}$$

The augmentation probability p_{aug} is typically set to 0.5, and the number of transformations applied is randomly sampled from the range $\{1,2\}$ for each signal that undergoes augmentation.

4.2.8. Error handling and robustness

To ensure robustness in the augmentation pipeline, we implement a multi-tiered error-handling strategy. For each augmentation attempt, we validate the resulting signal against the following criteria:

$$\text{valid}(\mathbf{x}^{\text{aug}}) = \begin{cases} \text{true} & \text{if } \|\mathbf{x}^{\text{aug}}\|_\infty < \infty \text{ and } \sum_i |\mathbf{x}_i^{\text{aug}}| > 0 \\ \text{false} & \text{otherwise.} \end{cases} \tag{33}$$

If an augmentation operation fails or produces invalid results, we employ a series of fallback strategies with decreasing complexity until a valid augmented signal is obtained or the original signal is retained as a last resort.

The complete augmentation pipeline thus integrates temporal, amplitude, spectral, and envelope transformations with adaptive class balancing and robust error handling to generate diverse, realistic, and balanced training datasets while preserving the essential discriminative characteristics of the ultrasound signals under study. The explanation of how the data is handled regarding augmented feature methodology is presented in Fig. 3.

Algorithm 1 summarizes the proposed data augmentation methodology. The procedure combines selective offline augmentation for class balancing with stochastic on-the-fly perturbations during training. By composing temporal, amplitude, spectral, and envelope-based transformations, the method generates diverse yet physically consistent signal variations while preserving the discriminative structure required for classification. In addition, a validation and fallback mechanism is used to ensure numerical stability and robustness throughout the augmentation pipeline.

4.3. Ensemble learning methods

Ensemble learning constructs a predictive model by combining multiple base learners, often referred to as weak classifiers or regressors, to form a composite model with enhanced generalization [58]. Let $\mathcal{H} = \{h_1, h_2, \dots, h_M\}$ denote a collection of M base models. The ensemble prediction $H(x)$ is defined by an aggregation function $\mathcal{A}$:

$$H(x) = \mathcal{A}(h_1(x), h_2(x), \dots, h_M(x)). \tag{34}$$

Two widely adopted ensemble strategies are *Bagging* and *Boosting*, which differ in data resampling procedures and model dependency structures.

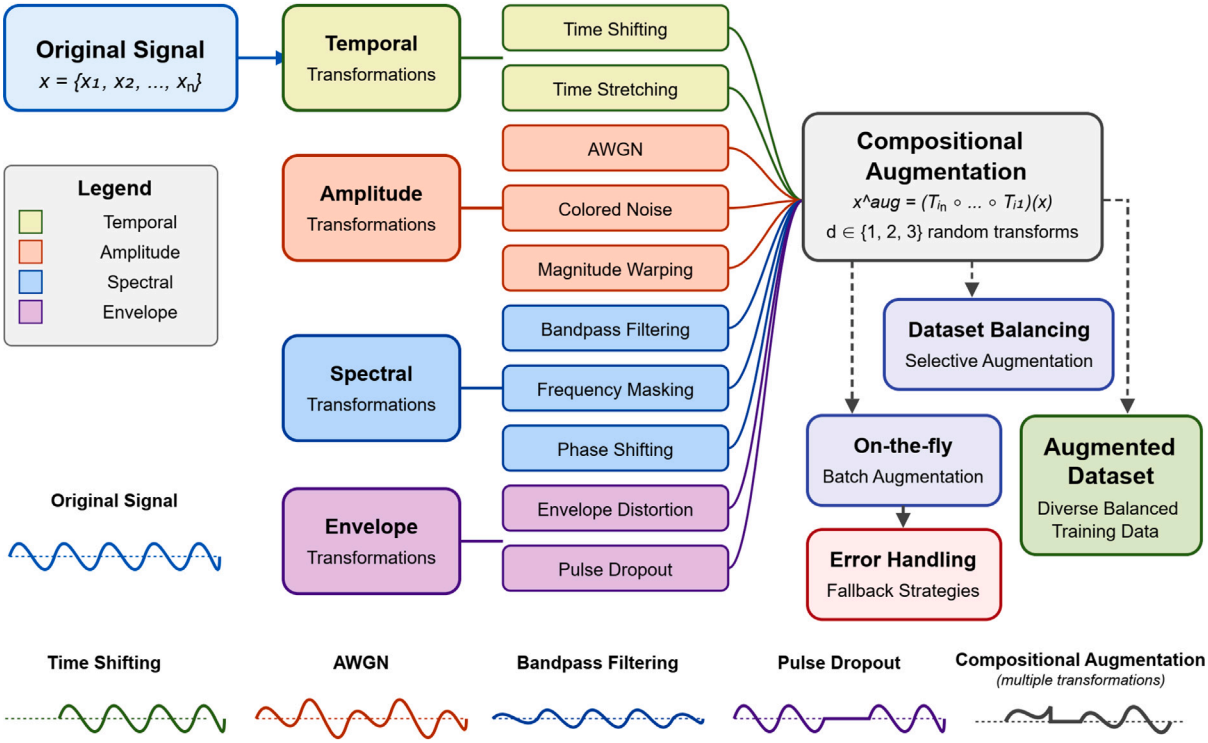


Fig. 3. Augmented feature methodology.

4.3.1. Bagging (bootstrap aggregating)

Bagging relies on the statistical principle of bootstrap sampling to generate multiple versions of the training dataset [59]. Given a labeled dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, each subset $D^{(m)}$ is sampled with replacement:

$$D^{(m)} \sim \text{Bootstrap}(D), \quad |D^{(m)}| = N. \tag{35}$$

Each base learner h_m is trained independently on its respective bootstrap sample:

$$h_m \leftarrow \mathcal{L}(D^{(m)}), \quad m = 1, \ldots, M. \tag{36}$$

The final prediction is computed for classification (majority vote) via:

$$H(x) = \text{mode}\left(h_1(x), h_2(x), \ldots, h_M(x)\right). \tag{37}$$

Bagging reduces variance and is particularly beneficial when base learners are high-variance models such as unpruned decision trees [60].

4.3.2. Boosting

Boosting constructs an ensemble in a sequential manner, where each model is trained to correct the errors made by the previous ones [61]. The ensemble output takes an additive form:

$$H(x) = \sum_{m=1}^{M} \alpha_m h_m(x) \tag{38}$$

where $\alpha_m \in \mathbb{R}$ represents the weight of the mth learner.

Boosting is effective in reducing both bias and variance [62]. Algorithms such as gradient boosting machines implement this paradigm with specific choices for loss functions $\mathcal{L}$ and optimization strategies.

4.3.3. Ensemble adaptation for multi-label classification

In multi-label settings, each instance x_i is associated with a binary label vector $\mathbf{y}_i \in \{0,1\}^L$, where L is the number of labels. Ensemble methods are adapted as binary relevance (BR), classifier chains (CC), and label powerset (LP).

The BR applies ensemble learning independently for each label $l \in \{1, \ldots, L\}$. CC incorporates label dependencies by modeling the conditional distribution $P(y_l \mid x, y_1, \ldots, y_{l-1})$. LP treats each unique label combination as a single multiclass output and applies an ensemble to this transformed space. Ensemble learning methods enhance performance in multi-label classification by capturing correlations between labels and increasing robustness to noise and class imbalance [63].

4.3.4. Employed ensemble algorithms

In this section, we provide a brief mathematical overview of the ensemble learning algorithms used in our experiments, namely RandomForest [64], XGBoost [65], LightGBM [66], and EvoBagging ensemble learning methods. These models are considered since they provide robust ensemble learning strategies that effectively handle the complexity of ultrasonic signal-based fault detection. These algorithms serve as baselines for comparing the proposed feature-engineered ensemble methods, demonstrating that classical ensemble learners can achieve high accuracy, stability, and interpretability in detecting insulator faults.

Random forest is an ensemble method based on bagging, where multiple decision trees are trained on different bootstrap samples of the dataset, and the final prediction is obtained by averaging for regression [67]. Each tree is trained by selecting a random subset of features at each node, which introduces decorrelation [68].

Given a set of trees $\{h_1(\cdot), h_2(\cdot), \ldots, h_M(\cdot)\}$, the random forest predictor is given by:

$$f_{\text{RF}}(x) = \frac{1}{M} \sum_{m=1}^{M} h_m(x) \tag{39}$$

where M is the number of trees and $h_m(x)$ denotes the prediction of the mth decision tree.

XGBoost is a gradient boosting method that builds an additive model in a forward stage-wise fashion by minimizing a regularized objective function [69]. The model is of the form:

$$f_{\text{XGB}}(x) = \sum_{m=1}^{M} \lambda_m h_m(x). \tag{40}$$

Algorithm 1: Data Augmentation Methodology Overview

Input: Original dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, transformation set $\mathcal{T}$, augmentation probability p_{aug}, maximum composition depth $d_{\max}$
Output: Balanced augmented dataset $\mathcal{D}^{\text{aug}}$ and on-the-fly augmentation operator $\mathcal{A}$

```
Initialize D^aug ← D
Compute class counts {n_l}_{l=1}^{L} and n_max ← max_{1≤l≤L} n_l
foreach class c_l ∈ C do
    m_l ← n_max − n_l
    if m_l > 0 then
        for j ← 1 to m_l do
            Sample (x, c_l) from class c_l with replacement
            Sample composition depth d ~ U{1, ..., d_max}
            Sample transformations {T_{i_1}, ..., T_{i_d}} ⊆ T without
              replacement
            Set x̃ ← x
            for k ← 1 to d do
                Apply x̃ ← T_{i_k}(x̃)
            end
            if valid(x̃) then
                D^aug ← D^aug ∪ {(x̃, c_l)}
            end
            else
                Generate fallback sample x̃ ← T_noise(x, γ_low)
                if valid(x̃) then
                    D^aug ← D^aug ∪ {(x̃, c_l)}
                end
                else
                    D^aug ← D^aug ∪ {(x, c_l)}
                end
            end
        end
    end
end
Define on-the-fly operator A for each training batch B = {(x_i, y_i)}_{i=1}^{b}
foreach (x_i, y_i) ∈ B do
    Sample u ~ U(0, 1)
    if u < p_aug then
        Sample depth d ~ U{1, 2} and compose random
          transformations from T
        Replace x_i by A(x_i)
    end
end
return D^aug and A
```

Each new function $h_m(x)$ is chosen to minimize the regularized loss:

$$\mathcal{L}^{(m)} = \sum_{i=1}^{n} \ell(y_i, f_{m-1}(x_i) + h_m(x_i)) + \Omega(h_m) \tag{41}$$

where ℓ is a differentiable loss function, and Ω is a regularization term (typically a function of tree complexity such as depth or number of leaves).

LightGBM is also based on gradient boosting, but it introduces optimization strategies such as histogram-based decision tree learning and leaf-wise tree growth with depth constraints [70]. This enhances both speed and accuracy.

The objective function is similar to that of XGBoost:

$$\mathcal{L}^{(m)} = \sum_{i=1}^{n} \ell(y_i, f_{m-1}(x_i) + h_m(x_i)) + \Omega(h_m). \tag{42}$$

However, LightGBM splits leaves based on the largest gain in loss reduction rather than level-wise expansion, leading to a more compact model with higher predictive power on some datasets [71].

4.3.5. EvoBagging

EvoBagging is an evolutionary ensemble learning algorithm that builds upon the bagging framework by evolving the contents of the bootstrap samples (bags) through evolutionary operators such as crossover, mutation, and selection [18].

Initially, N bags $\{B_1, B_2, \ldots, B_N\}$ are generated via bootstrapping. At each iteration, a fitness score is assigned to each bag based on its classification performance and size:

$$\text{fitness}(B_i) = \alpha_i \cdot \left(1 + \frac{\phi_i}{K}\right) \tag{43}$$

where α_i is the classification accuracy of the model trained on bag B_i, ϕ_i is the bag size, and K is a normalization constant encouraging larger bags.

The evolution of bags proceeds via crossover, exchanging samples between high-fitness bags, mutation randomly replacing samples in some bags, and generation gap periodically introducing new randomly generated bags to maintain diversity. The final ensemble output is given by:

$$f_{\text{EvoBag}}(x) = \frac{1}{N} \sum_{i=1}^{N} h_i(x) \tag{44}$$

where $h_i(x)$ is the decision tree trained on the ith evolved bag.

4.3.6. PPOBagging

We propose PPOBagging, an adaptive ensemble learning algorithm that evolves a population of base learners through reinforcement learning. Different than classical bagging methods that operate through static resampling, PPOBagging dynamically optimizes sample composition and inter-model diversity by casting ensemble construction as a Markov decision process. The method uses proximal policy optimization (PPO) [72] to guide actions that iteratively construct, mutate, combine, and prune decision tree classifiers, forming a robust and diverse ensemble.

Classical bagging methods operate through static resampling, without dynamic optimization of sample composition or inter-model diversity [73]. Inspired by evolutionary learning and policy-gradient reinforcement learning, PPOBagging frames the ensemble construction as a sequential decision-making problem. At each step, an agent observes a state summarizing the ensemble and chooses an action to evolve it toward higher accuracy and diversity.

The environment state at time step t, denoted $s_t \in \mathbb{R}^6$, encodes ensemble-level statistics:

$$s_t = \left[\frac{1}{B_t}\sum_{i=1}^{B_t}\frac{|D_i|}{|D_{\max}|},\ \overline{M},\ \overline{D},\ \overline{P},\ \frac{\overline{d}}{d_{\max}},\ \overline{\rho}\right] \tag{45}$$

where $|D_i|$ is the size of the ith bag with B_t being the number of bags at time t, $\overline{M}$ is the average performance metric (e.g., F1-score), $\overline{D}$ is the average pairwise diversity, $\overline{P}$ is the average payoff (fitness), $\overline{d}$ is the average tree depth, and $\overline{\rho}$ is the average error correlation among bags.

At each step, the agent chooses one of four discrete actions $a_t \in \{0, 1, 2, 3\}$:

1. `Add` ($a_t = 0$): Generate a new stratified bag $\mathcal{D}_{\text{new}}$ through sampling with replacement:

$$\mathcal{D}_{\text{new}} = \{(x_i, y_i) \sim \mathcal{D}_{\text{train}} \mid i = 1, 2, \ldots, n\} \tag{46}$$

where $n \sim \mathcal{U}(\frac{\text{max_initial_size}}{2}, \text{max_initial_size})$

2. `Crossover` ($a_t = 1$): Mix correct/wrong samples from two parent bags:

$$\mathcal{D}_{\text{child}_1} = \mathcal{D}_{p1}^{\text{correct}} \cup \mathcal{D}_{p2}^{\text{wrong}} \tag{47}$$

$$\mathcal{D}_{\text{child}_2} = \mathcal{D}_{p2}^{\text{correct}} \cup \mathcal{D}_{p1}^{\text{wrong}} \tag{48}$$

where $\mathcal{D}_{pi}^{\text{correct}} = \{(x, y) \in \mathcal{D}_{pi} \mid h_i(x) = y\}$ and $\mathcal{D}_{pi}^{\text{wrong}} = \{(x, y) \in \mathcal{D}_{pi} \mid h_i(x) \neq y\}$, with h_i being the classifier trained on bag i

3. `Mutate` ($a_t = 2$): Replace correctly classified instances with hard negatives:

$$D_i' = D_i^{\text{wrong}} \cup S \cup \mathcal{N} \tag{49}$$

where $D_i^{\text{wrong}} = \{(x, y) \in D_i \mid h_i(x) \neq y\}$, $S \subset D_i^{\text{correct}}$ is a random subset of correctly classified examples, and $\mathcal{N} = \{(x, y) \sim D_{\text{train}} \mid (x, y) \notin D_i^{\text{wrong}} \cup S\}$

4. `Prune` ($a_t = 3$): Remove the least fit bag:

$$B_{t+1} = B_t \setminus \{\arg\min_{i \in B_t} \pi_i\} \tag{50}$$

where π_i is the bag's payoff.

The ensemble's reward at time t is the change in validation performance:

$$r_t = \mathcal{M}(B_t) - \mathcal{M}(B_{t-1}) \tag{51}$$

where $\mathcal{M}(B)$ is the ensemble's weighted voting score (e.g., F1 or accuracy). Each bag $i \in B$ contributes a payoff:

$$\pi_i = M_i \cdot \left(1 + \lambda_D D_i\right) \cdot \left(1 - \lambda_\rho \rho_i\right) \cdot \frac{\text{size_coef} + |D_i|}{\text{size_coef}} \tag{52}$$

where M_i is its standalone metric, D_i is its diversity measure, ρ_i is the mean error correlation, and the rightmost term is a size weight factor. The diversity D_i for a bag is calculated as

$$D_i = 1 - \frac{1}{|B| - 1} \sum_{j \in B, j \neq i} \frac{|\{x \in \mathcal{X} \mid h_i(x) = h_j(x)\}|}{|\mathcal{X}|}, \tag{53}$$

while the error correlation ρ_i is defined as

$$\rho_i = \frac{1}{|B| - 1} \sum_{j \in B, j \neq i} \frac{|\{x \in \mathcal{X} \mid h_i(x) \neq y(x) \wedge h_j(x) \neq y(x)\}|}{|\{x \in \mathcal{X} \mid h_i(x) \neq y(x) \vee h_j(x) \neq y(x)\}|}. \tag{54}$$

The ensemble prediction uses weighted voting:

$$\hat{y}(x) = \arg\max_{c \in C} \sum_{i=1}^{|B|} w_i \cdot P_i(y = c \mid x), \quad w_i = \pi_i \tag{55}$$

PPOBagging uses the PPO algorithm to update the policy network parameters θ. The policy network $\pi_\theta(a|s)$ maps states to action probabilities and has a dual-head architecture that outputs both action probabilities and state value estimates. The PPO objective function uses a clipped surrogate loss:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t\right)\right] \tag{56}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the estimated advantage function, and ϵ is the clipping parameter (typically 0.2). The full loss function is:

$$L(\theta) = L^{\text{CLIP}}(\theta) - c_1 L^{VF}(\theta) + c_2 S[\pi_\theta](s_t) \tag{57}$$

where $L^{VF}(\theta) = (V_\theta(s_t) - V_t^{\text{target}})^2$ is the value function loss, $S[\pi_\theta](s_t) = -\sum_a \pi_\theta(a|s_t) \log \pi_\theta(a|s_t)$ is the entropy bonus, and c_1, c_2 are coefficients.

PPOBagging employs generalized advantage estimation (GAE) [74] to compute advantages:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t-1}\delta_{T-1} \tag{58}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the TD error, γ is the discount factor, and λ is the GAE parameter. The complete algorithm is presented in Algorithm 2.

The policy network is implemented as a multi-layer perceptron with layer normalization:

$$\begin{aligned} h_1 &= \text{ReLU}(\text{LayerNorm}(W_1 s_t + b_1)) \\ h_2 &= \text{ReLU}(\text{LayerNorm}(W_2 h_1 + b_2)) \\ h_3 &= \text{ReLU}(\text{LayerNorm}(W_3 h_2 + b_3)) \\ \pi(a|s_t) &= \text{Softmax}(W_\pi h_3 + b_\pi) \\ V(s_t) &= W_V h_3 + b_V \end{aligned} \tag{59}$$

Algorithm 2: PPOBagging: Ensemble evolution via reinforcement learning.

```
Data: Training set D_train, Validation set D_val
Result: Final ensemble B*
Initialize bag population B_0 ← {b_1, ..., b_k};
Initialize PPO policy network π_θ(a_t | s_t);
for episode = 1 to N do
    for step = 1 to T do
        Encode state s_t ← f(B_t);
        Sample action a_t ~ π_θ(· | s_t);
        Apply action a_t to obtain new population B_{t+1};
        Evaluate reward: r_t = M(B_{t+1}) − M(B_t);
        Store transition (s_t, a_t, r_t);
    end
    Compute returns and advantages using GAE;
    for epoch = 1 to K do
        Sample mini-batch of transitions;
        Compute PPO objective L(θ);
        Update θ using gradient descent;
        Break if KL divergence exceeds threshold;
    end
    Update best ensemble B* if M(B_t) > M(B*);
end
return B*
```

The implementation uses a batch actor-critic architecture [75], where the actor outputs action probabilities, and the critic estimates state values. A key feature of PPOBagging is its adaptive ensemble construction, which maintains computational efficiency by pruning suboptimal bags and bounding the maximum ensemble size. The training cost is $C_{\text{train}} = \sum_i T(D_i)$, where T is the cost of training a tree, and the voting cost is $C_{\text{vote}} = \mathcal{O}(B \cdot C)$ for ensemble voting.

5. Results

This section presents the evaluation of our signal classification methodology, as illustrated in Fig. 4. Our approach follows four main analytical paths: raw signal window processing, enhanced feature engineering, ROCKET-family transformations and deep learning. All experiments were conducted on a high-performance workstation running Gentoo Linux. The computational environment includes an Intel(R) Core(TM) i9-14900K and an NVIDIA GeForce RTX 4090 GPU. The system is equipped with 32 GiB of RAM and multiple high-capacity storage volumes, ensuring efficient data handling and model training.

As shown in Fig. 4, we begin with raw electrical signals and process them through three distinct pipelines. The first pipeline uses raw signal windows directly as model inputs. The second pipeline extracts domain-specific enhanced features, which are further refined into augmented features with additional statistical and frequency-domain descriptors. Those are then compared with ROCKET-family and deep learning models.

Through 5-fold cross-validation, we evaluate each approach using ensemble-based classifiers (RandomForest, XGBoost, LightGBM) and a support vector machine (SVM) as a baseline. We also analyze the performance of specialized ensemble methods, including EvoBagging and PPOBagging algorithms. Our experiments measure classification accuracy, precision, recall, and F1 scores, along with computational efficiency. We identify challenging samples and examine class-specific performance patterns to understand model behavior and limitations.

The following sections detail each component of our experimental framework, comparing the performance gains achieved through feature engineering against raw signal processing, and examining the trade-offs between handcrafted features and convolutional approaches.

Table 2
Cross-validation performance using raw signal windows.

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1 (Macro)
RandomForest	0.5500 ± 0.0408	0.5668 ± 0.0277	0.5632 ± 0.0240	0.5489 ± 0.0322
XGBoost	0.5050 ± 0.0549	0.5088 ± 0.0462	0.5063 ± 0.0400	0.4999 ± 0.0450
LightGBM	0.5150 ± 0.0322	0.5187 ± 0.0203	0.5163 ± 0.0197	0.5117 ± 0.0215
SVM	0.4583 ± 0.0279	0.3811 ± 0.0361	0.4946 ± 0.0095	0.3861 ± 0.0096

Table 3
Cross-validation performance using extracted features.

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1 (Macro)
RandomForest	0.9100 ± 0.0170	0.9156 ± 0.0193	0.9117 ± 0.0178	0.9102 ± 0.0186
XGBoost	0.9017 ± 0.0220	0.9085 ± 0.0220	0.9048 ± 0.0204	0.9025 ± 0.0240
LightGBM	0.9017 ± 0.0207	0.9068 ± 0.0213	0.9046 ± 0.0210	0.9016 ± 0.0240
SVM	0.8517 ± 0.0433	0.8661 ± 0.0447	0.8547 ± 0.0418	0.8517 ± 0.0413

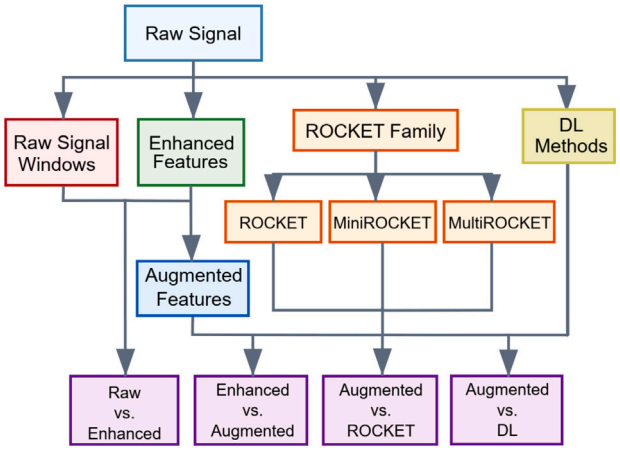


Fig. 4. Methodological framework for signal classification comparing three approaches: raw signal windows, enhanced/augmented features, and ROCKET-family transformations. The diagram illustrates the processing pipeline from raw signals to classification results, with comparative evaluation between different feature extraction techniques.

5.1. Evaluating the enhanced feature method

In this section, we present the classification results obtained using raw signal windows and compare them to those obtained from the proposed feature engineering approach. Raw signal windows of length 1000 samples were directly used as input to the classifiers. Table 2 summarizes the average and standard deviation across the folds for each evaluation metric.

We then evaluated the models using the 50-dimensional handcrafted features extracted from each signal window. The results are presented in Table 3.

The feature engineering approach significantly outperformed the raw input method across all models and metrics. Notably, the RandomForest model trained on engineered features achieved an average accuracy of 91.0% compared to only 55.0% when trained on raw windows. Improvements were similarly observed in macro-precision, macro-recall, and macro-F1, indicating that the extracted features provided more discriminative information for classification.

The performance boost can be attributed to factors as: The engineered features incorporate domain-specific signal characteristics such as frequency content, statistical moments, and signal shape metrics; The raw signal, while rich, introduces high dimensionality and noise, potentially obscuring discriminative patterns; Classical models like SVMs and ensemble trees benefit substantially from compact, meaningful representations.

Fig. 5 presents the top features as ranked by the RandomForest classifier trained on the enhanced feature set. The ranking is based on the gain metric, which reflects each feature's relative contribution to model decisions.

From those observations, some conclusions can be drawn from the feature ranking: Features such as `low_band_power`, `low_rel_power`, and `mid_high_band_power` highlight the importance of frequency-domain information, particularly in the lower and mid-frequency ranges, as they capture energy concentrations that may be associated with fault signatures in electrical components. The standard deviation of the wavelet coefficients, including `wavelet_0_std` and `wavelet_1_std`, further emphasizes the relevance of localized time-frequency structures, indicating their strong discriminative capability.

Statistical descriptors such as `percentile_25`, `percentile_75`, `rms`, `std`, and `energy` reveal that amplitude distribution and energy dispersion within each window play a meaningful role in distinguishing between classes. Measures related to entropy and spectral characteristics, namely `spectral_entropy` and `spectral_centroid`, suggest that signal complexity and variations in spectral shape are also significant contributors to classification performance. Finally, temporal transition features, including `zero_crossing_rate` and `hilbert_inst_freq_mean`, capture rapid oscillatory behavior and phase-related information, which can be indicative of transient fault patterns.

5.1.1. Feature ablation study

To evaluate the contribution of different feature groups to classification performance, we conducted an ablation study using 5-fold cross-validation with XGBoost as the classifier. The enhanced feature set was divided into seven groups: time-domain features (mean, standard deviation, kurtosis, extrema, percentiles, energy metrics), zero-crossing rate, peak analysis (peak count and amplitude statistics), frequency band power analysis across six ultrasonic frequency ranges (10–210 kHz), spectral features (entropy, centroid, bandwidth, dominant frequency), wavelet decomposition coefficients using Daubechies-4 wavelets, and Hilbert transform features (envelope and instantaneous frequency analysis).

Table 4 presents the performance of individual feature groups, while Table 5 shows the top-performing feature combinations. The results show that frequency band features achieve the highest individual performance (87.83% accuracy), while the combination of time-domain, frequency bands, and spectral features yields the optimal accuracy (90.67%). The enhanced feature approach outperforms raw signal classification (48.00% accuracy with 1000 features) by 42.67 percentage points while using 95% fewer features.

The ablation analysis identifies three primary findings. Frequency band features appear in eight of the ten highest-performing combinations. The combination of time-domain, frequency bands, and spectral features (90.67% accuracy) outperforms the complete 50-feature set (89.50% accuracy) by 1.17 percentage points. Raw signal classification (48.00% accuracy, 1000 features) underperforms all enhanced feature

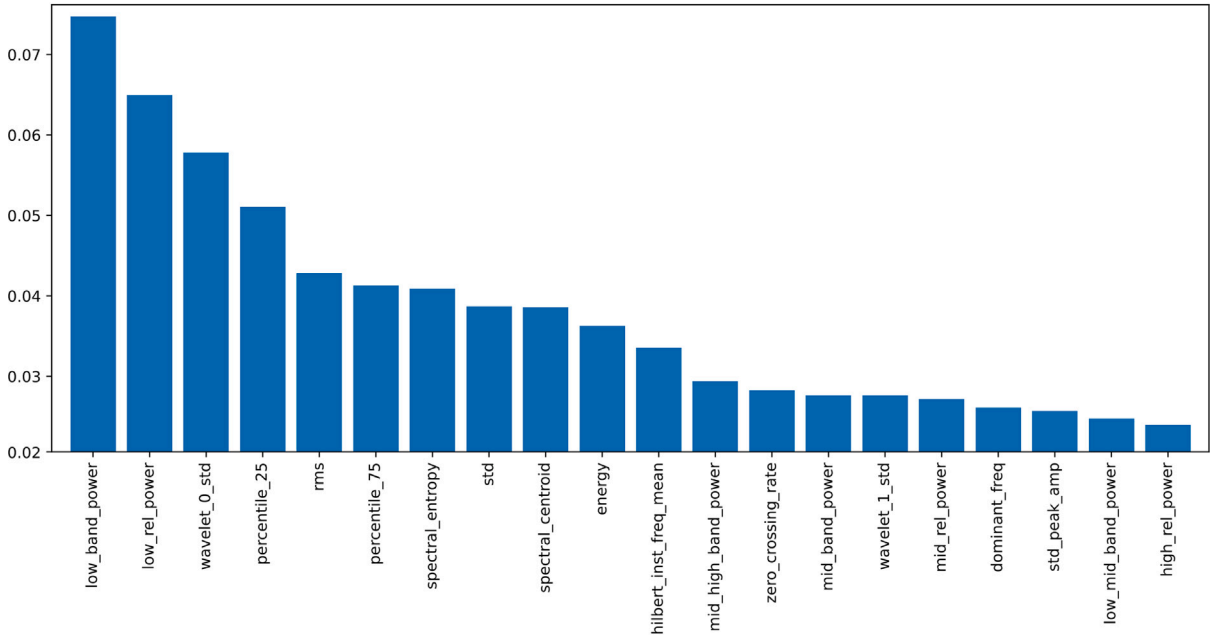


Fig. 5. Top feature importances extracted from the enhanced feature set using RandomForest.

Table 4
Classification accuracy of individual feature groups using 5-fold cross-validation. Features extracted from 1000-sample ultrasonic signal windows at 2 MHz sampling rate.

Feature group	Features	Accuracy (%)
Frequency bands	12	87.83 ± 3.36
Spectral	4	80.83 ± 3.91
Wavelet	15	79.33 ± 3.05
Hilbert transform	4	70.83 ± 4.12
Time-domain	11	67.33 ± 3.31
Peak analysis	3	59.67 ± 3.56
Zero-crossing rate	1	58.00 ± 5.13

Table 5
Top-performing feature combinations ranked by classification accuracy. Results obtained using XGBoost classifier with 5-fold cross-validation on ultrasonic signal classification task.

Feature combination	Features	Accuracy (%)
Time-domain + Frequency bands + Spectral	27	90.67 ± 2.49
Time-domain + Frequency bands	23	90.00 ± 2.84
All enhanced features	50	89.50 ± 2.21
Frequency bands + Hilbert	16	89.17 ± 2.30
Frequency bands + Spectral + Wavelet	31	89.00 ± 2.20
Frequency bands + Spectral	16	88.83 ± 2.96
Frequency bands + Wavelet	27	88.50 ± 2.32
Raw signal + All enhanced features	1050	88.50 ± 2.66
Peak analysis + Frequency bands	15	87.83 ± 2.92
Frequency bands only	12	87.83 ± 3.36

combinations. Time-domain features achieve 67.33% accuracy individually but contribute to the optimal 90.67% accuracy when combined with frequency-based features.

Fig. 6 presents the ablation study results across four analytical perspectives. The feature combination ranking (top-left) shows time-domain and frequency bands achieving the highest accuracy. The accuracy versus feature count relationship (top-right) indicates that raw signals (1000 features) produce lower classification accuracy than optimized enhanced feature combinations (16–50 features). The individual feature group analysis (bottom-left) ranks frequency bands first among single feature types. The error bar comparison (bottom-right) shows

Table 6
Difficult samples analysis.

ID	True Class	Most Pred. As	Agreement	Miscl. Rate	Entropy	Model predictions		
						RF	XGB	LGBM
6	C	D	0.67	0.67	0.64	C	D	D
21	C	D	1.00	1.00	0.00	D	D	D
50	C	B	0.67	0.67	0.64	B	C	B
51	B	C	1.00	1.00	0.00	C	C	C
52	C	D	1.00	1.00	0.00	D	D	D
59	A	D	1.00	1.00	0.00	D	D	D
61	C	B	0.67	0.67	0.64	B	B	C
67	C	A	1.00	1.00	0.00	A	A	A
71	C	B	1.00	1.00	0.00	B	B	B
87	B	C	1.00	1.00	0.00	C	C	C
96	A	D	1.00	1.00	0.00	D	D	D
108	A	D	1.00	1.00	0.00	D	D	D

cross-validation standard deviations ranging from 2.20% to 2.96% for the top-performing combinations.

5.1.2. *Difficult samples analysis*

Table 6 presents 12 samples that were challenging for the tested ensemble models. These samples exhibited high misclassification rates, with the majority of models failing to predict the correct class.

The examination of misclassification patterns reveals a significant finding: in 75% of difficult samples (9 out of 12), all three models unanimously agreed on the same incorrect prediction (agreement = 1.00), with corresponding entropy values of 0.00. This consensus on misclassification indicates potential limitations in the feature representation rather than model-specific factors. The remaining samples with partial agreement (agreement = 0.67) exhibited entropy values of 0.64, reflecting predictions distributed across two different classes. Fig. 7 depicts this bimodal distribution of entropy values, suggesting two distinct types of difficult samples: those where all models confidently make the same incorrect prediction and those where models show some disagreement.

Class distribution analysis reveals that samples from class C accounted for 58.3% of difficult samples (7 out of 12), as shown in Fig. 8. These were frequently confused with either class B or class D. The class-specific performance is further quantified in Fig. 9, which displays

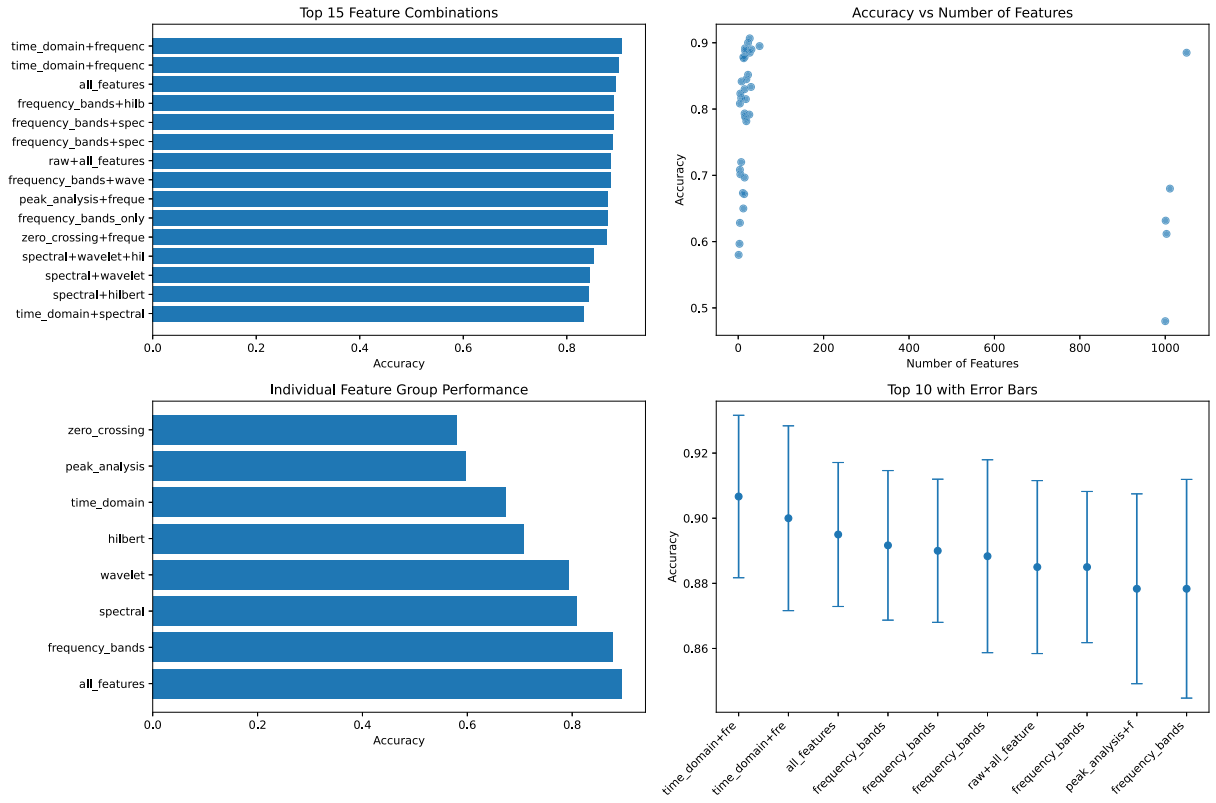


Fig. 6. Ablation study results visualization showing (a) top 15 feature combinations ranked by accuracy, (b) relationship between number of features and classification performance, (c) individual feature group performance comparison, and (d) top 10 combinations with error bars indicating cross-validation variability.

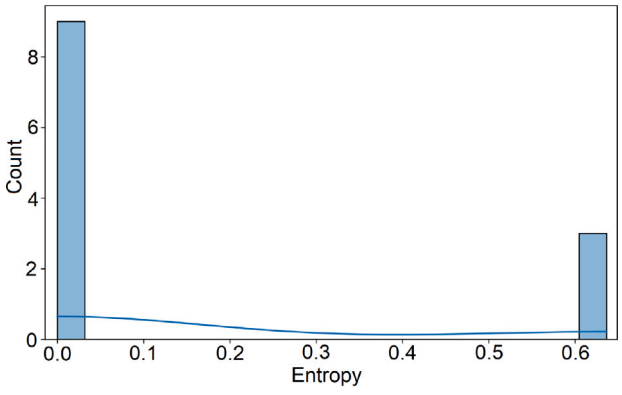


Fig. 7. Histogram of prediction entropy values across difficult samples. The bimodal distribution shows two distinct groups: samples with zero entropy (complete model agreement on incorrect predictions) and samples with moderate entropy (partial agreement).

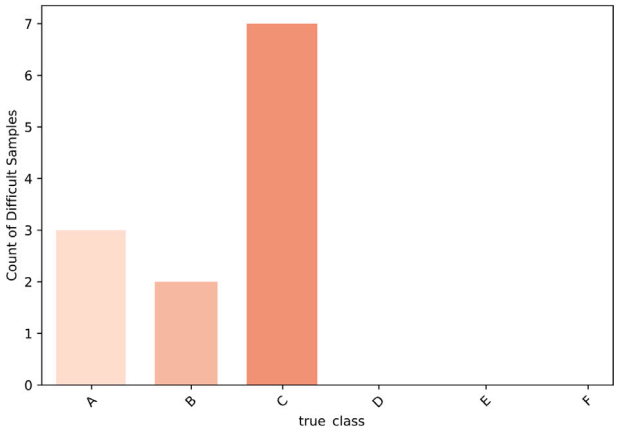


Fig. 8. Distribution of difficult samples by true class. Class C exhibits the highest number of difficult samples, followed by class A and class B. Classes E and F show no difficult samples.

the F1 scores for each class across the three models. Classes E and F achieved perfect classification (F1 = 1.00) across all models, while class C demonstrated the lowest performance (F1 = 0.74–0.78). Classes A, B, and D showed intermediate performance with F1 scores ranging from 0.82 to 0.90. The consistent pattern of F1 scores across different models provides evidence that the classification challenges are class-specific rather than model-specific.

Examining the specific confusion patterns, we identified several recurring misclassifications: All three samples from class A (samples 59, 96, and 108) were consistently misclassified as class D; A bidirectional confusion exists between classes B and C, with two B samples (51 and 87) misclassified as C, and three C samples (50, 61, and 71) misclassified as B; Three samples from class C (6, 21, and 52) were consistently misclassified as class D; One sample from class C (67) was unanimously misclassified as class A.

These patterns are further explained by the t-SNE visualization in Fig. 10, which provides insight into the spatial distribution of samples in the feature space. Difficult samples (marked with crosses) typically occur in regions where different classes overlap or at class boundaries. Classes E and F show clear separation in the feature space, corresponding to their perfect classification performance. In contrast, classes A, B, C, and D exhibit areas of overlap, explaining the observed confusion patterns.

The observed confusion patterns have implications for model refinement. The C-D confusion (3 samples), A-D confusion (3 samples), and

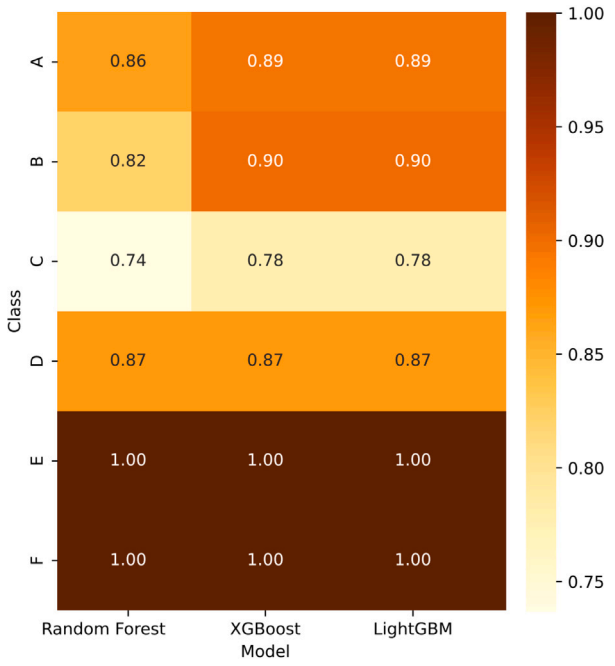


Fig. 9. Per-class F1 scores across different models. Classes E and F achieve perfect classification with all models, while class C consistently shows the lowest performance. The similarity in F1 patterns across models suggests class-inherent feature representation challenges.

B-C confusion (5 samples) represent 91.7% of all difficult cases, suggesting these particular class boundaries require additional attention. The t-SNE visualization indicates that the current feature representation does not adequately separate these overlapping classes. Additional domain-specific features may help address these confusions, particularly for class C that appears in the majority of difficult samples.

The distribution of entropy values and the consistent F1 score patterns across models suggest that the current ensemble approach does not overcome the fundamental limitations in distinguishing certain classes. Alternative feature engineering techniques or specialized classification approaches for the most problematic class pairs may yield improvements in overall system performance.

5.1.3. EvoBagging performance on enhanced features

We evaluated the EvoBagging algorithm using the handcrafted enhanced features derived from time, frequency, and wavelet domains. On the fifth cross-validation fold, EvoBagging achieved a strong classification performance, with an overall accuracy of 90.83% and a macro-averaged F1 score of 0.908. These results affirm the algorithm's ability to form competitive ensembles via evolutionary operations such as mutation, crossover, and fitness-based selection.

Table 7 summarizes the per-class metrics. The classifier exhibits robust precision and recall across most categories. Notably, it achieves perfect scores on E and F, which have well-defined feature distributions. Meanwhile, more ambiguous classes like B and C show slightly lower precision, likely due to overlapping spectral characteristics with other contamination modes.

EvoBagging demonstrates strong generalization on the enhanced features (see Fig. 11). Compared to other methods in our benchmark, EvoBagging maintains a favorable trade-off between ensemble diversity and accuracy and speed, with the added benefit of population interpretability.

The importance analysis identifies features 15, 19, and 10 as the most influential for classification. The final payoff distribution centers around 1.5, indicating convergence to solutions of similar quality.

Table 7
Per-class performance of EvoBagging on enhanced features (Fold 5).

Class	Precision	Recall	F1-score
A	0.90	0.95	0.93
B	0.84	0.80	0.82
C	0.80	0.80	0.80
D	0.90	0.90	0.90
E	1.00	1.00	1.00
F	1.00	1.00	1.00

Table 8
Per-class performance of PPOBagging on enhanced features (Fold 1).

Class	Precision	Recall	F1-score
A	0.95	0.95	0.95
B	0.89	0.85	0.87
C	0.85	0.85	0.85
D	0.95	1.00	0.98
E	1.00	1.00	1.00
F	1.00	1.00	1.00

Sample frequency analysis reveals that certain training instances (particularly samples 0–3) appear more frequently in high-performing bags, suggesting their informative value for the classification task.

Fig. 12 displays the decision tree structures from the four highest-performing bags. All trees achieved perfect classification scores (1.0) with payoff values between 1.521 and 1.528. Despite this performance parity, the trees exhibit structural diversity in their decision boundaries.

Bag 4 utilizes feature 14 for its primary split, while Bags 38, 28, and 34 use features 30, 17, and 29, respectively. All trees consistently employ features 15, 19, and 10 in their decision paths, aligning with the feature importance analysis. This structural diversity while maintaining identical performance metrics demonstrates the algorithm's capacity to discover multiple viable classification strategies.

The evolutionary process converged after 7 rounds, with stable performance metrics indicating efficient exploration of the solution space. This convergence efficiency, combined with the structural diversity of top-performing classifiers, demonstrates the effectiveness of the evolutionary bagging approach for classification tasks.

5.1.4. PPOBagging performance on enhanced features

We evaluated the proposed PPOBagging algorithm on the same enhanced feature set used in previous experiments. The proposed model, which dynamically constructs and evolves bagged ensembles via reinforcement learning, demonstrated superior performance during its first cross-validation fold. It achieved a validation accuracy of 94.17% and a macro-averaged F1 score of 0.9412, outperforming all previously tested ensemble methods.

The algorithm was configured to run for 50 episodes with 10 decision steps per episode. Early stopping was triggered at episode 7 once no further gains in ensemble performance were detected. The reinforcement learning trajectory showed rapid convergence, with the ensemble achieving its best performance (94.12% weighted F1) as early as episode 2.

Table 8 summarizes the per-class precision, recall, and F1-scores. PPOBagging maintains strong generalization across all six classes. Notably, it achieves perfect scores on E and F, and high scores even in previously difficult classes such as B and C.

Compared to EvoBagging, PPOBagging exhibits higher accuracy by leveraging a reward-driven optimization loop. The improvement in difficult-to-classify classes highlights the benefit of PPO's policy-guided bag composition and mutation operations. Additionally, the perfect recall in classes with distinct signal features underscores PPOBagging's ability to preserve high-precision learners while promoting useful diversity.

PPOBagging presents a compelling method in the family of adaptive ensemble construction. Its dynamic exploration of the ensemble design

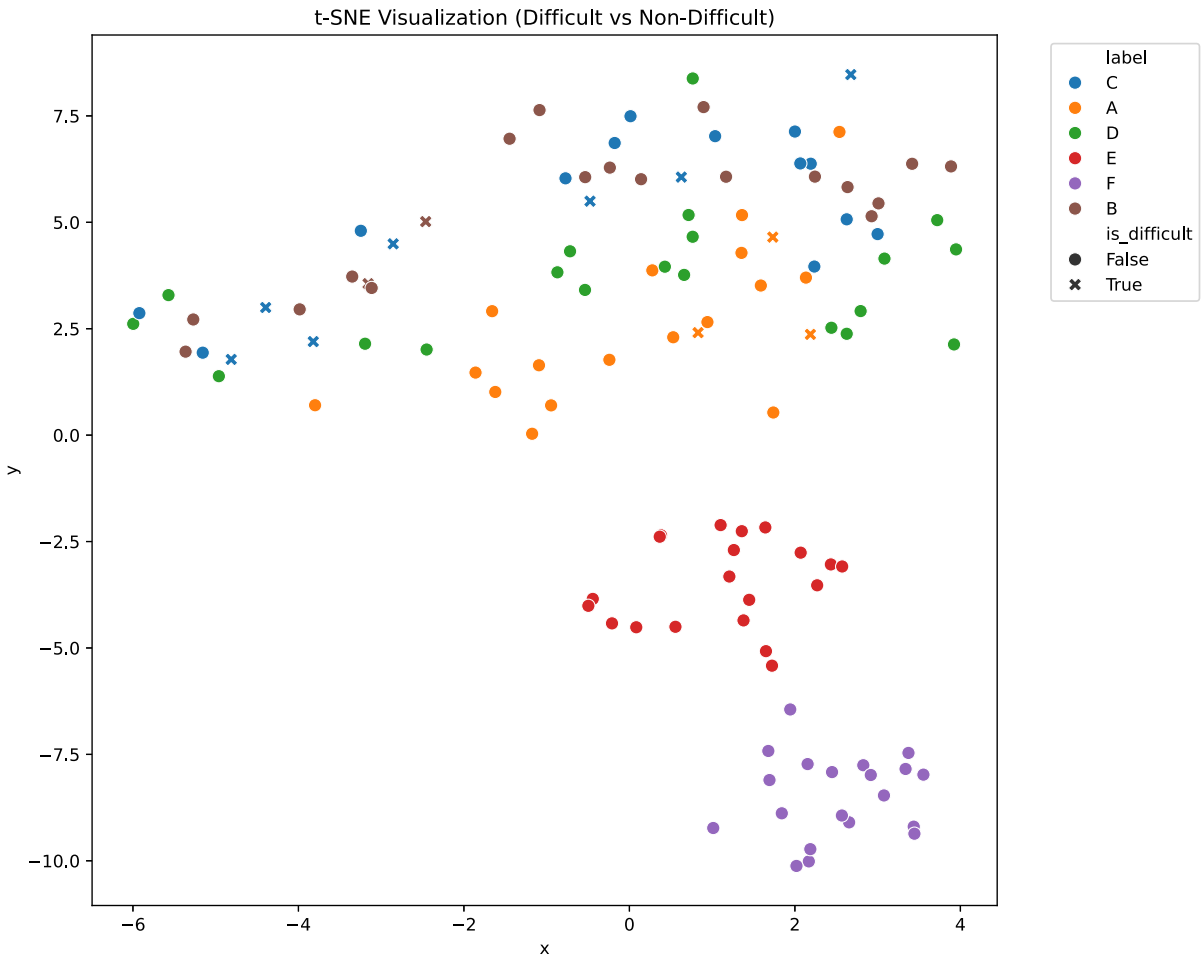


Fig. 10. t-SNE visualization of the feature space, distinguishing between correctly classified (circles) and difficult samples (crosses). Difficult samples tend to occur at class boundaries or in regions of class overlap. Classes E and F demonstrate clear separation, while classes A, B, C, and D show overlapping regions.

space, backed by policy gradient updates and structured state, action representations, enables it to surpass traditional static or evolutionary ensemble strategies in both accuracy and robustness.

The method exhibits distinct performance patterns as depicted in Fig. 13. The score evolution (top-left panel) demonstrates rapid convergence to optimal performance within the first 4 episodes, with the ensemble score (blue line) consistently outperforming the average individual classifier score (red line). This convergence pattern differs from traditional evolutionary approaches, showing more rapid initial improvement.

Diversity among bags (top-right panel) stabilizes at approximately 0.17 after initial exploration, a notably lower value than typically observed in purely evolutionary approaches. This suggests that the reinforcement learning policy converges toward a more specialized set of complementary classification strategies. The bag size evolution (middle-left panel) shows a consistent downward trend from 95 to 80 samples, indicating the agent's preference for more compact, specialized classifiers.

The tree depth evolution (middle-right panel) reveals a gradual increase in complexity, stabilizing at approximately 6 levels. This controlled growth in tree complexity suggests the RL policy learns to balance model expressiveness against potential overfitting. The policy and value loss curves (bottom-left panel) exhibit the characteristic convergence pattern of PPO training, with rapid initial decrease followed by stabilization, indicating effective policy optimization.

The action distribution analysis (bottom-right panel) provides unique insights into the agent's learned strategy. The policy shows a preference for crossover operations in early episodes, transitioning to a more balanced approach utilizing mutation and selective removal in later episodes. This adaptive action selection demonstrates the advantage of the RL approach in dynamically adjusting ensemble construction strategies based on current performance.

The decision tree structures of the four highest-performing bags generated by the PPORLBagging method are displayed in Fig. 14. All four trees achieve perfect classification scores (1.0) with similar payoff values ranging from 1.348 to 1.381. The trees maintain diversity in their structure despite their equivalent performance, demonstrating the algorithm's ability to discover multiple viable classification strategies.

Analysis of the top-left tree (Bag 0) reveals a primary split on feature 10, with subsequent splits on features 15, 48, and 28. This tree achieves a diversity score of 0.160 with a depth of 5 levels. Bag 1 (top-right) employs a different structure with the primary split on feature 10 but utilizes features 27, 15, and 32 in key decision points, achieving slightly higher diversity (0.166) with greater depth (7 levels).

Bags 18 and 22 (bottom row) demonstrate additional structural variations while maintaining perfect classification. Bag 18 uses feature 48 for its primary split and achieves a diversity score of 0.170, while Bag 22 uses feature 7 with a diversity score of 0.179. The consistent use of features 10 and 15 across multiple trees aligns with the feature importance analysis (bottom-left panel in Fig. 13), which identifies these as among the most significant features. The diversity in tree structures maintained by the PPO approach differs from traditional bagging methods, which typically produce more homogeneous ensembles. This structural diversity, combined with the controlled tree depth, contributes to the ensemble's robust performance.

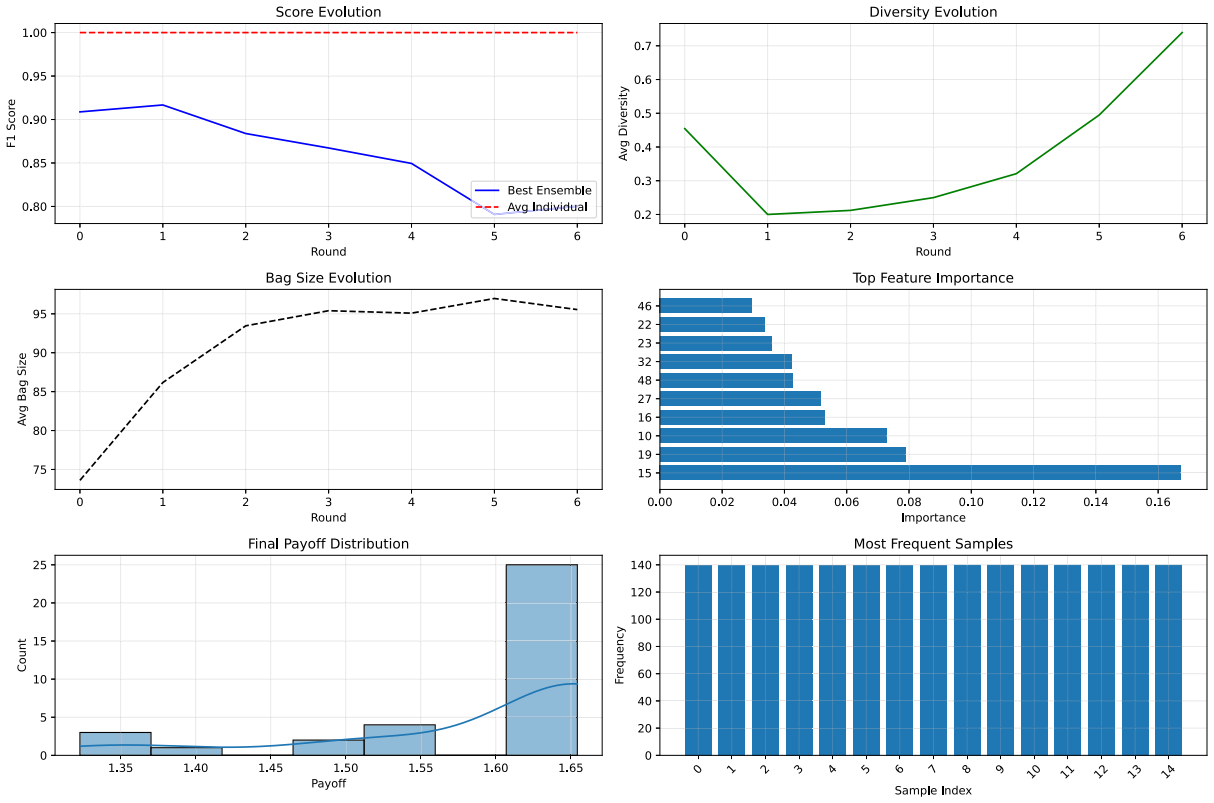


Fig. 11. Performance metrics of the EvoBagging algorithm across training rounds. Top: F1 score evolution and bag diversity. Middle: Bag size evolution and feature importance. Bottom: Payoff distribution and sample selection frequency.

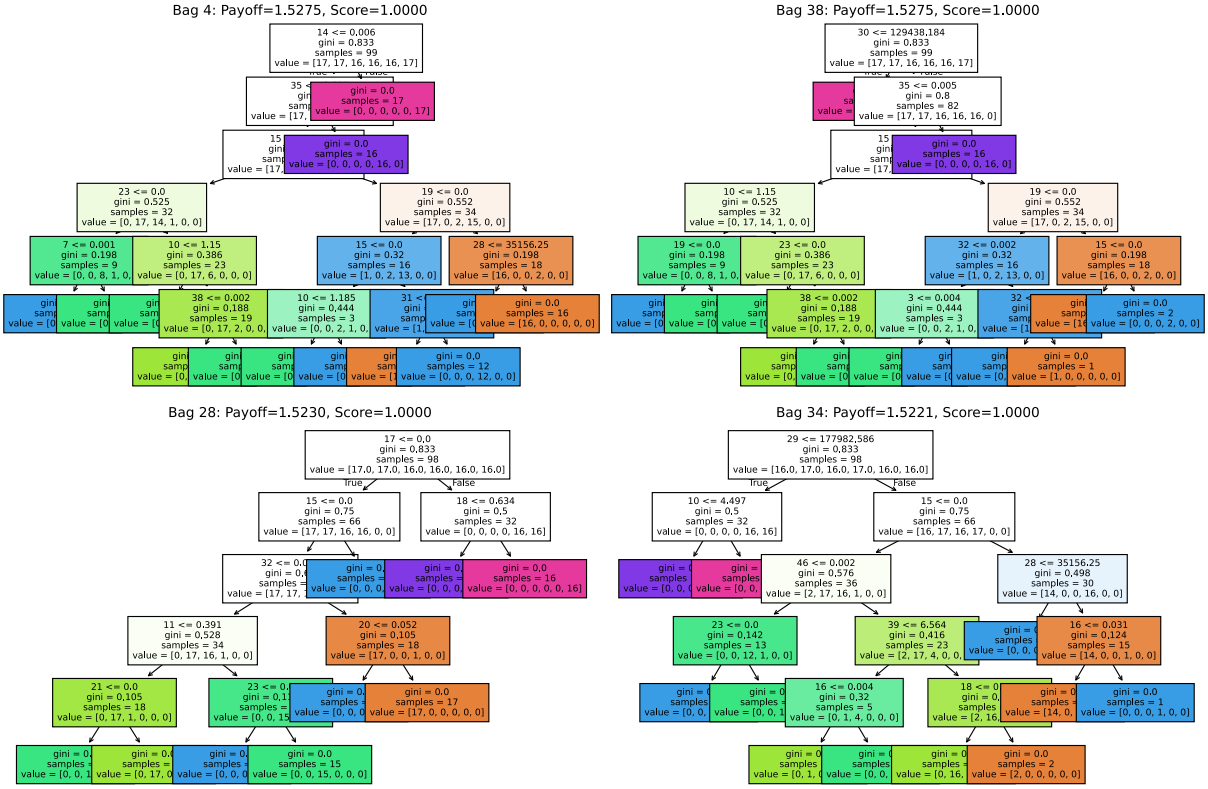


Fig. 12. Decision tree structures of the four highest-performing bags. Each tree achieved perfect classification (score = 1.0) through different decision boundaries, illustrating ensemble diversity.

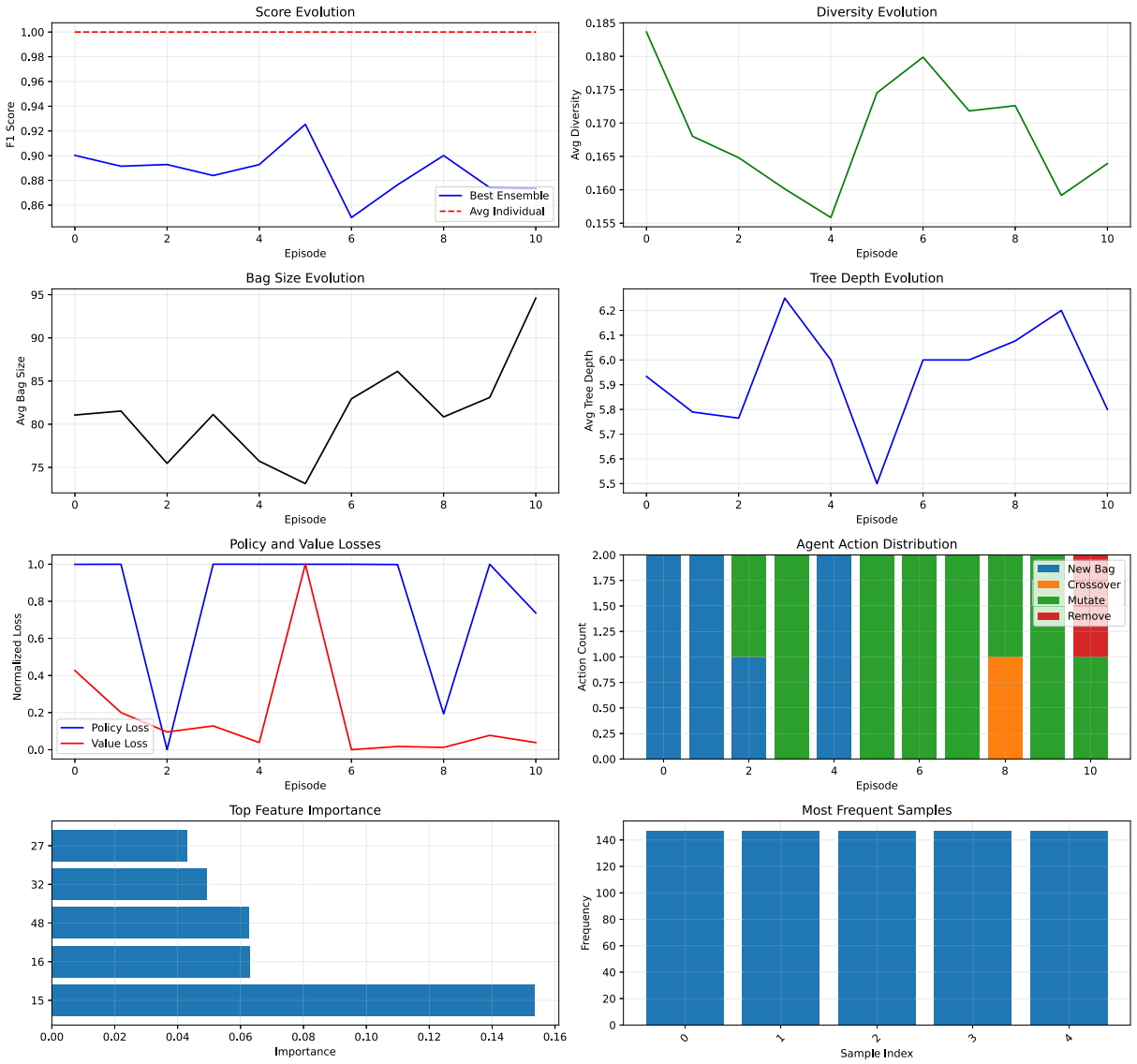


Fig. 13. Performance metrics of the PPORLBagging algorithm across training episodes. Top: F1 score evolution and bag diversity. Middle: Bag size evolution and tree depth evolution. Bottom: Policy/value losses and agent action distribution.

The reinforcement learning approach to bagging demonstrates several distinct characteristics compared to evolutionary methods. While both approaches achieve comparable final classification performance, the PPO-based method shows faster initial convergence and produces trees with more controlled complexity. The average tree depth in PPORLBagging (approximately 6 levels) is notably lower than typically observed in evolutionary approaches while maintaining equivalent classification performance.

The reinforcement learning policy's ability to dynamically select different ensemble manipulation strategies (as evidenced in the action distribution analysis) represents a key advantage over fixed-strategy evolutionary approaches. This adaptive behavior allows the PPO method to escape local optima more efficiently and explore promising regions of the solution space in a directed manner.

Additionally, the diversity values observed in PPO-generated bags (0.160–0.179) are generally lower than those seen in evolutionary approaches, yet the ensemble maintains optimal performance. This suggests that the RL policy identifies a more efficient set of complementary classifiers, requiring less explicit diversity to achieve robust ensemble prediction.

Table 9
Comparison of classification accuracy with and without feature augmentation.

Model	Standard features	Augmented features	Improvement
RandomForest	0.8833	0.9083	+0.0250
XGBoost	0.9083	0.9167	+0.0083
SVM	0.8667	0.8583	−0.0083
LightGBM	0.8667	0.9167	+0.0500

As a drawback, the necessary time to train PPOBagging is higher than the average for the usual ensemble bagging methods, which is not a desirable feature.

5.2. Comparison between standard and augmented feature engineering

To further enhance classification performance, we explored the impact of an augmented feature extraction pipeline. In this variant, additional statistical and frequency-domain features were incorporated. Table 9 presents the accuracy of each model trained using the standard and augmented feature sets.

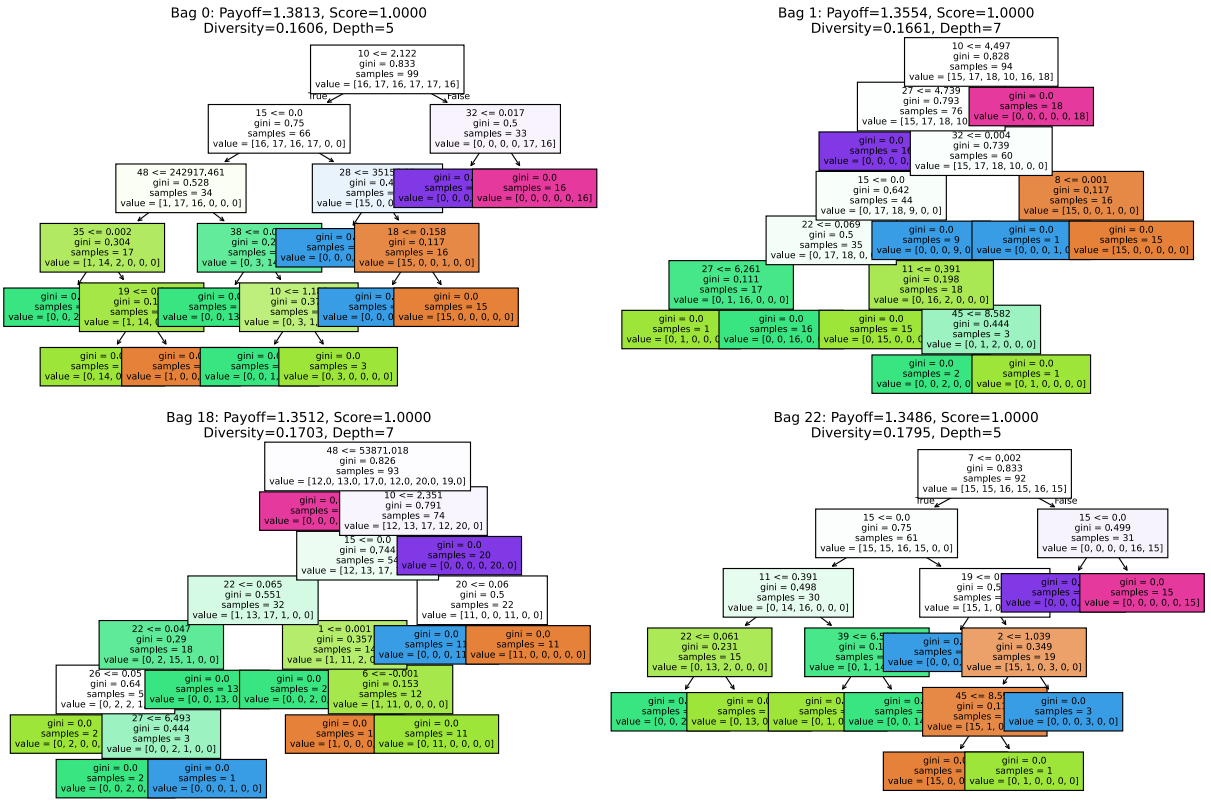


Fig. 14. Decision tree structures of the four highest-performing bags generated by the PPORLBagging algorithm. Each tree achieved perfect classification (score = 1.0) with diversity scores ranging from 0.160 to 0.179 and tree depths between 5 and 7 levels.

The results show a consistent performance gain for most ensemble models with the augmented features. LightGBM, in particular, exhibited the highest absolute improvement, achieving a final accuracy of 91.67%. SVM, however, showed a minor drop in performance, possibly due to overfitting or increased feature dimensionality not aligning well with the kernel's capacity.

The LightGBM classifier trained with the augmented feature set yielded the best overall performance. Table 10 reports the detailed classification metrics for each class.

Table 10
LightGBM classification report (augmented features).

Class	Precision	Recall	F1-Score	Support
A	0.95	0.90	0.92	20
B	0.90	0.90	0.90	20
C	0.88	0.70	0.78	20
D	0.80	1.00	0.89	20
E	1.00	1.00	1.00	20
F	1.00	1.00	1.00	20
Accuracy	0.9167			
Macro Avg	0.92	0.92	0.91	120
Weighted Avg	0.92	0.92	0.91	120

5.3. Enhanced feature method against ROCKET, MiniROCKET, and MultiROCKET

To assess the efficacy of recent convolutional feature generation methods for time series classification, we applied ROCKET, MiniROCKET, and MultiROCKET transforms to raw signal windows. These transforms generate high-dimensional representations through randomized convolutional kernels, which are subsequently classified using RandomForest and XGBoost models.

Table 11 reports the transformation time, classifier time, and total runtime. MiniROCKET proved to be the most efficient, completing the transformation in 0.61 s, compared to 16.39 s for ROCKET and 9.75 s for MultiROCKET.

Table 12 summarizes the classification accuracy across all six pipelines. MiniROCKET+XGBoost and MultiROCKET+XGBoost tied for best performance at 88.33%, outperforming ROCKET-based methods. MiniROCKET+RandomForest provided the best balance between speed and accuracy.

Across all ROCKET variants, classes E and F achieved perfect precision, recall, and F1-scores. However, classification of C remained challenging, with F1-scores consistently below 0.70. MiniROCKET + XGBoost achieved the highest F1-score for class A (0.9524) and strong overall balance. MultiROCKET + XGBoost slightly improved class C (F1 = 0.7000), compared to MiniROCKET (F1 = 0.6857). ROCKET + RandomForest offered a good speed-to-accuracy ratio, with consistent performance across mid-range classes.

Table 11
Execution time for ROCKET-based transformations and classifiers.

Pipeline	Transform time (s)	Classifier time (s)	Total time (s)
ROCKET + RandomForest	16.39	1.23	17.63
ROCKET + XGBoost	16.39	27.41	43.81
MiniROCKET + RandomForest	0.61	0.64	1.25
MiniROCKET + XGBoost	0.61	32.41	33.02
MultiROCKET + RandomForest	9.75	1.83	11.58
MultiROCKET + XGBoost	9.75	49.25	59.00

The ROCKET-family transforms, particularly MiniROCKET and MultiROCKET, offered strong classification performance with minimal preprocessing. However, While only slightly behind the augmented feature-engineering pipeline in accuracy (91.67%), MiniROCKET-based methods significantly increased training time.

Table 12
Classification accuracy using convolutional kernel transforms.

Pipeline	Accuracy
MiniROCKET + XGBoost	0.8833
MultiROCKET + XGBoost	0.8833
ROCKET + RandomForest	0.8667
ROCKET + XGBoost	0.8667
MultiROCKET + RandomForest	0.8667
MiniROCKET + RandomForest	0.8500

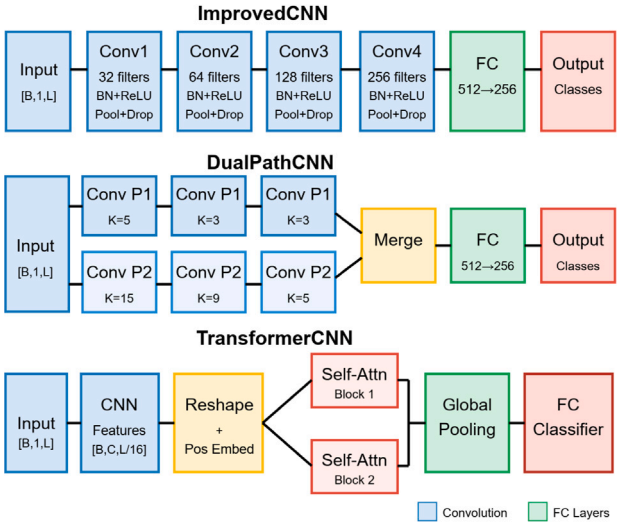


Fig. 15. Tested deep learning architectures.

Handcrafted features enabled better interpretability and superior accuracy on minority classes, which remains valuable for reliability-critical systems such as fault detection in power electronics.

5.4. Comparing enhanced features with deep learning approaches

To compare traditional feature engineering with modern deep learning approaches, we evaluated three neural network architectures directly on raw signal windows: ImprovedCNN, DualPathCNN, and TransformerCNN. These models were trained with advanced techniques including mixup data augmentation, learning rate scheduling, and early stopping to maximize generalization performance. An overview of the architectures is presented in Fig. 15.

The employed neural architectures are designed to process 1D temporal signals and can be summarized as follows: The ImprovedCNN is a standard deep convolutional model with four blocks combining normalization, activation, pooling, and dropout, using progressively larger filters and ending with fully connected layers for classification. The DualPathCNN processes the input through two parallel branches with small and large kernels to capture both local and long range patterns, merging their features before classification. The TransformerCNN integrates convolutional layers with transformer blocks, leveraging multi head attention to model long range and complex temporal dependencies beyond traditional CNNs.

All models were applied directly to raw signal windows of length 1000 samples, without any manual feature extraction. The networks were trained and evaluated using the same 5-fold cross-validation methodology employed in our previous experiments, with an NVIDIA GeForce RTX 4090 GPU accelerating the training process. Table 13 presents the classification accuracy achieved by each deep learning model on raw signal inputs, compared to our best ensemble model with enhanced features.

The TransformerCNN architecture achieved higher performance (79.17%) compared to other deep learning models, outperforming the DualPathCNN (70.83%) and ImprovedCNN (58.33%) variants. This pattern suggests that the ability to model long-range dependencies through self-attention mechanisms provides benefits for this signal classification task.

Table 13
Classification accuracy of deep learning models on raw signal inputs.

Model	Accuracy (%)
ImprovedCNN	58.33
DualPathCNN	70.83
TransformerCNN	79.17
LightGBM (Augmented Features)	91.67

Table 14
Performance comparison of deep learning models versus traditional approaches.

Approach	Accuracy	Input type
RandomForest	0.5500	Raw Signal
ImprovedCNN	0.5833	Raw Signal
DualPathCNN	0.7083	Raw Signal
TransformerCNN	0.7917	Raw Signal
RandomForest	0.9100	Enhanced Features
LightGBM	0.9167	Augmented Features

All deep learning models outperformed traditional machine learning algorithms when both were applied to raw signal windows. The TransformerCNN's accuracy of 79.17% on raw signals represents a 24.17 percentage point improvement over the best traditional model on raw signals (RandomForest at 55.00%). However, the deep learning models operating on raw signals did not match the performance of ensemble methods working with enhanced features. The LightGBM classifier with augmented features achieved 91.67% accuracy, outperforming the TransformerCNN by 12.50 percentage points.

Table 14 compares the performance of different modeling approaches on both raw signals and engineered features.

While deep learning models extracted meaningful patterns directly from raw signals, several practical considerations favor the enhanced feature approach. The feature engineering pipeline provides a higher degree of interpretability by revealing discriminative signal characteristics, as evidenced by the feature importance rankings presented in Fig. 5. In contrast, deep learning models, particularly those based on transformer architectures, tend to operate as black boxes, offering limited transparency in their decision-making process.

In terms of computational requirements, deep learning models demand significantly more resources during training. For instance, the TransformerCNN configuration includes approximately 2.4 million parameters and relies on GPU acceleration, whereas ensemble methods based on engineered features achieve superior performance with substantially lower computational overhead.

Regarding sample efficiency, deep learning approaches generally require large volumes of training data to achieve strong generalization. The results demonstrate that, given the limited dataset size, feature engineering methods retain a clear advantage in classification performance.

The integration of domain knowledge represents a key strength of the enhanced feature approach, as it incorporates expertise from signal processing and electrical systems, including relevant spectral bands and characteristic signal patterns for fault detection. In contrast, neural networks must infer these relationships directly from the data without explicit guidance.

The results show that while deep learning approaches process raw signal data effectively, feature engineering paired with ensemble methods remains the most effective classification strategy for this application. The 12.50 percentage point advantage of enhanced features over the best deep learning model demonstrates the value of domain knowledge and targeted feature extraction in signal classification tasks.

For reliability-critical applications in electrical systems, the combination of enhanced feature engineering with ensemble methods provides a balance of performance, interpretability, and computational efficiency. Future work may explore hybrid approaches that combine the strengths of both methods, potentially using deep learning to augment handcrafted features or incorporating domain knowledge into neural network architectures.

6. Conclusion

This study presented an approach to multi-label fault detection in insulator ultrasound signals through feature-engineered ensemble methods. Our research has demonstrated that carefully designed feature extraction, combined with advanced ensemble learning techniques, significantly outperforms raw signal classification methods.

The experimental results showed that engineered features derived from time-domain, frequency-domain, wavelet-domain, and envelope characteristics provided rich, discriminative representations that improved classification accuracy from around 55% with raw signals to over 90% with the proposed feature set. Among the ensemble methods evaluated, our novel PPOBagging algorithm achieved the highest performance with a 94.17% accuracy and a macro-F1 score of 0.9412, surpassing conventional methods including RandomForest, XGBoost, and LightGBM.

We also established that feature augmentation provides consistent performance improvements for most ensemble models, with LightGBM exhibiting the most significant gain (+5.00%) when trained with the augmented feature set. Comparative analysis against modern convolutional kernel transforms (ROCKET, MiniROCKET, and MultiROCKET) revealed that while these methods offer competitive accuracy (up to 88.33%), they fall short of our feature engineering approach in terms of both performance and interpretability. The same happens when our approach is compared to deep learning methods; while the deep learning methods have a better ability to handle raw signals, they still fall short when compared to the feature engineering approach.

The proposed signal augmentation methodology further enhances the robustness of the models by introducing controlled variations that preserve class-discriminative characteristics. This approach is particularly valuable for handling class imbalance and improving generalization in real-world deployment scenarios where signal characteristics may vary due to environmental factors. From an application perspective, this research provides practical tools for non-destructive monitoring of medium-voltage insulators in power distribution systems. The ability to accurately detect and classify different types of faults, including contamination, moisture, and physical damage, enables proactive maintenance strategies that can prevent catastrophic failures and improve overall system reliability.

Future work should explore the integration of DL architectures with the proposed feature engineering approach, potentially through attention mechanisms that can leverage domain-specific signal attributes. Additionally, investigating transfer learning techniques could enable the adaptation of these models to different insulator types and environmental conditions with minimal retraining.

CRediT authorship contribution statement

Laio Oriel Seman: Writing – review & editing, Writing – original draft, Software, Resources, Methodology, Investigation, Data curation, Conceptualization. **Stefano Frizzo Stefenon:** Writing – review & editing, Visualization, Validation, Project administration. **Valderi Reis Quietinho Leithardt:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration. **Juan Francisco De Paz Santana:** Writing – review & editing, Validation, Supervision, Funding acquisition. **Kin-Choong Yow:** Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Methodology.

Declaration of competing interest

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

Acknowledgments

Colaboración Consejería de educación de la Junta de Castilla y León grupo de investigación ESAL-EXPERT SYSTEM AND APPLICATIONS LAB (ESALAB).

References

[1] Stefenon SF, Seman LO, Singh G, Yow KC. Enhanced insulator fault detection using optimized ensemble of deep learning models based on weighted boxes fusion. Int J Electr Power Energy Syst 2025;168:110682. http://dx.doi.org/10.1016/j.ijepes.2025.110682.

[2] Deng F, Xie Z, Mao W, Li B, Shan Y, Wei B, Zeng H. Research on edge intelligent recognition method oriented to transmission line insulator fault detection. Int J Electr Power Energy Syst 2022;139:108054. http://dx.doi.org/10.1016/j.ijepes.2022.108054.

[3] Stefenon SF, Singh G, Souza BJ, Freire RZ, Yow KC. Optimized hybrid YOLOu-Quasi-ProtoPNet for insulators classification. IET Gener Transm Distrib 2023;17(15):3501–11. http://dx.doi.org/10.1049/gtd2.12886.

[4] Corso MP, Stefenon SF, Singh G, Matsuo MV, Perez FL, Leithardt VRQ. Evaluation of visible contamination on power grid insulators using convolutional neural networks. Electr Eng 2023;105:3881–94. http://dx.doi.org/10.1007/s00202-023-01915-2.

[5] Stefenon SF, Bruns R, Sartori A, Meyer LH, Ovejero RG, Leithardt VRQ. Analysis of the ultrasonic signal in polymeric contaminated insulators through ensemble learning methods. IEEE Access 2022;10:33980–91. http://dx.doi.org/10.1109/ACCESS.2022.3161506.

[6] Carvalho JPM, Stefenon SF, Yow KC, Ovejero RG, Leithardt VRQ. N-BEATS neural network applied for insulator fault prediction considering EMD methods. In: International conference on electrical, computer, communications and mechatronics engineering, vol. 5, Zanzibar, Tanzania: IEEE; 2025, p. 1–6. http://dx.doi.org/10.1109/ICECCME64568.2025.11277686.

[7] Guo K, Sun W, Lan R, Wei Y, Li G. Insulation performance and harmonic characteristics of insulation materials of the cable under moisture conditions. Electr Eng 2025;107(2):1671–82. http://dx.doi.org/10.1007/s00202-024-02616-0.

[8] Akbari M, Shayegani-Akmal AA. Experimental investigation on the accelerated aging of silicone rubber insulators based on thermal stress. Int J Electr Power Energy Syst 2023;149:109049. http://dx.doi.org/10.1016/j.ijepes.2023.109049.

[9] Seman LO, Buratto WG, Gonzalez GV, Leithardt VRQ, Nied A, Stefenon SF. Differentiable neural search architecture with zero-cost metrics for insulator fault prediction. Results Eng 2026;29:109716. http://dx.doi.org/10.1016/j.rineng.2026.109716.

[10] Massalkhi H, Lutfi A, El-Hag A. Classification of defects in outdoor ceramic insulators using machine learning. MEPCON, In: International middle east power system conference, vol. 25, Cairo, Egypt: IEEE; 2024, p. 1–4. http://dx.doi.org/10.1109/MEPCON63025.2024.10850157.

[11] Hao Y, Liang Z, Shen Z, Zhang Y, Huang S, Yang L. Application of ultrasound in fracture toughness measurement of GIS epoxy insulators. IEEE Trans Instrum Meas 2024;73:1–8. http://dx.doi.org/10.1109/TIM.2024.3375417.

[12] Jalal M, Khalil IU, ul Haq A. Deep learning approaches for visual faults diagnosis of photovoltaic systems: State-of-the-art review. Results Eng 2024;23:102622. http://dx.doi.org/10.1016/j.rineng.2024.102622.

[13] Stefenon SF, Seman LO, Klaar ACR, Ovejero RG, Leithardt VRQ. Hypertuned-YOLO for interpretable distribution power grid fault location based on EigenCAM. Ain Shams Eng J 2024;15(6):102722. http://dx.doi.org/10.1016/j.asej.2024.102722.

[14] Khan AA, Chaudhari O, Chandra R. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. Expert Syst Appl 2024;244:122778. http://dx.doi.org/10.1016/j.eswa.2023.122778.

[15] Yang Z, Li Z, Du X, Wang F, Qiu Y, Wei J. Intelligent prediction model of a polymer fracture grouting effect based on an information acquisition optimizer-optimized random forest. Results Eng 2025;25:107382. http://dx.doi.org/10.1016/j.rineng.2025.107382.

[16] Zeng Z, Song B, Wu S, Li Y, Nie D, Wang L. Prediction of breakdown voltage of long air gaps under switching impulse voltage based on the ISSA-XGBoost model. Energies 2025;18(7):1800. http://dx.doi.org/10.3390/en18071800.

[17] Singh NK, Nagahara M. LightGBM-, SHAP-, and correlation-matrix-heatmap-based approaches for analyzing household energy data: Towards electricity self-sufficient houses. Energies 2024;17(17):4518. http://dx.doi.org/10.3390/en17174518.

[18] Ngo G, Beard R, Chandra R. Evolutionary bagging for ensemble learning. Neurocomputing 2022;510:1–14. http://dx.doi.org/10.1016/j.neucom.2022.08.055.

[19] Zhang Y, Song F, Zhang C, Zhang T, Deng W. Alleviating local optima in bridge bidding via diverse PPO ensembling. IEEE Access 2025;13:59285–94. http://dx.doi.org/10.1109/ACCESS.2025.3555303.

[20] Bier A, Jastrzebska A, Olszewski P. Variable-length multivariate time series classification using ROCKET: A case study of incident detection. IEEE Access 2022;10:95701–15. http://dx.doi.org/10.1109/ACCESS.2022.3203523.

[21] Kuo RJ, Xu ZX. Predictive maintenance for wire drawing machine using MiniRocket and GA-based ensemble method. Int J Adv Manuf Technol 2024;134(3):1661–76. http://dx.doi.org/10.1007/s00170-024-14225-z.

[22] Tan CW, Dempster A, Bergmeir C, Webb GI. MultiRocket: Multiple pooling operators and transformations for fast and effective time series classification. Data Min Knowl Discov 2022;36(5):1623–46. http://dx.doi.org/10.1007/s10618-022-00844-1.

[23] Serikbay A, Akhmetov Y, Nurmanova V, Zollanvari A, Bagheri M. YOLO-based detection and classification of high-voltage insulator surface contamination. IEEE Trans Dielectr Electr Insul 2026;33(1):623–32. http://dx.doi.org/10.1109/TDEI.2025.3631380.

[24] Mehmood B, Atef Ghunem R, El-Hag A, Tay LL. UHF detection of the eroding DC dry-band arcing on silicone rubber insulation. IEEE Trans Dielectr Electr Insul 2025;32(3):1380–6. http://dx.doi.org/10.1109/TDEI.2024.3510222.

[25] Elmancy A, Hamdi A, Shaban K, El-Hag A. Real-time insulator defect detection and damage assessment on edge for UAV-based inspection. In: 2025 26th international middle east power systems conference. MEPCON, 2025, p. 1–7. http://dx.doi.org/10.1109/MEPCON66918.2026.11360113.

[26] Lutfi A, El-Hag A, Shaban K. Nonintrusive ultrasonic sensing and deep learning for outdoor ceramic insulator assessment. IEEE Trans Dielectr Electr Insul 2024;31(6):2993–3000. http://dx.doi.org/10.1109/TDEI.2024.3403072.

[27] Zhao W, Xu M, Cheng X, Zhao Z. An insulator in transmission lines recognition and fault detection model based on improved faster RCNN. IEEE Trans Instrum Meas 2021;70:1–8. http://dx.doi.org/10.1109/TIM.2021.3112227.

[28] Wang B, Dong M, Ren M, Wu Z, Guo C, Zhuang T, Pischler O, Xie J. Automatic fault diagnosis of infrared insulator images based on image instance segmentation and temperature analysis. IEEE Trans Instrum Meas 2020;69(8):5345–55. http://dx.doi.org/10.1109/TIM.2020.2965635.

[29] He M, Qin L, Deng X, Liu K. MFI-YOLO: Multi-fault insulator detection based on an improved YOLOv8. IEEE Trans Power Deliv 2024;39(1):168–79. http://dx.doi.org/10.1109/TPWRD.2023.3328178.

[30] Alshalawi AH, Al-Ismail FS. Partial discharge detection based on ultrasound using optimized deep learning approach. IEEE Access 2024;12:5151–62. http://dx.doi.org/10.1109/ACCESS.2024.3350555.

[31] de Barros Bezerra JM, Lima AMN, Deep GS, da Costa EG. An evaluation of alternative techniques for monitoring insulator pollution. IEEE Trans Power Deliv 2009;24(4):1773–80. http://dx.doi.org/10.1109/TPWRD.2009.2016628.

[32] Shafique U, Muhammad Alam S, Rashid U, Javed W, Anwaar H, Shah Zeb M, Ahmad T, Imtiaz U, Nzanywayingoma F. Infrared thermography-based insulator fault classification via unsupervised clustering and semi-supervised learning. IEEE Access 2024;12:180781–91. http://dx.doi.org/10.1109/ACCESS.2024.3404930.

[33] Wang S, Zhou M, Li N. Discharge ultraviolet image characteristics and insulation state evaluation of polluted and wet suspension porcelain insulator strings. IEEE Trans Instrum Meas 2024;73:1–10. http://dx.doi.org/10.1109/TIM.2024.3417545.

[34] Stefenon SF, Freire RZ, Meyer LH, Corso MP, Sartori A, Nied A, Klaar ACR, Yow KC. Fault detection in insulators based on ultrasonic signal processing using a hybrid deep learning technique. IET Sci Meas Technol 2020;14(10):953–61. http://dx.doi.org/10.1049/iet-smt.2020.0083.

[35] Stefenon SF, Silva MC, Bertol DW, Meyer LH, Nied A. Fault diagnosis of insulators from ultrasound detection using neural networks. J Intell Fuzzy Systems 2019;37(5):6655–64. http://dx.doi.org/10.3233/JIFS-190013.

[36] Ferreira TV, Germano AD, da Costa EG. Ultrasound and artificial intelligence applied to the pollution estimation in insulations. IEEE Trans Power Deliv 2012;27(2):583–9. http://dx.doi.org/10.1109/TPWRD.2011.2178042.

[37] Luijten B, et al. Ultrasound signal processing: From models to deep learning. Ultrasound Med Biol 2022;48(3):497–513. http://dx.doi.org/10.1016/j.ultrasmedbio.2022.11.003.

[38] Song K, Feng J, Chen D. A survey on deep learning in medical ultrasound imaging. Front Phys 2024;12:1398393. http://dx.doi.org/10.3389/fphy.2024.1398393.

[39] Wang H, Cheng L, Liao R, Zhang S, Yang L. Nonlinear ultrasonic nondestructive detection and modelling of kissing defects in high voltage composite insulators. IEEE Trans Dielectr Electr Insul 2020;27(3):924–31. http://dx.doi.org/10.1109/TDEI.2019.008523.

[40] Liu P, Li J, Ye Z, Wu Z, Qiu N, Cui B, Wei J, Peng Z. Ultrasonic detection and diagnostic method for the internal defects in epoxy composite. IEEE Trans Dielectr Electr Insul 2024;31(6):3221–30. http://dx.doi.org/10.1109/TDEI.2024.3425316.

[41] Gorgoz I, Cebeci M. Analysis of electric field and temperature distributions of non-uniformly contaminated silicone composite insulators using deep learning. IEEE Access 2025;13:94721–39. http://dx.doi.org/10.1109/ACCESS.2025.3574493.

[42] Chinchu M, Vennila H, Bibin GS. Improved YOLO v8: An efficient real-time fault detection for high-voltage insulators for autonomous aerial vehicles (AAVs) images using deep learning techniques. IEEE Access 2026;14:24110–26. http://dx.doi.org/10.1109/ACCESS.2026.3660918.

[43] Akash R, Sarathi R, Haddad M. Deep learning approach for noninvasive tracking detection in silicone rubber insulators using fluorescent fiber sensing. IEEE Sens. Lett 2025;9(11):1–4. http://dx.doi.org/10.1109/LSENS.2025.3621294.

[44] Basharan V, Shanmugam G. Explainable AI-based coating defect recognition in HV porcelain insulators using improved GAN and multi-modal PD feature fusion under imbalanced data. IEEE Trans Dielectr Electr Insul 2026;33(1):595–603. http://dx.doi.org/10.1109/TDEI.2025.3574648.

[45] Serikbay A, Bagheri M, Zollanvari A, Phung B. Ensemble pretrained convolutional neural networks for the classification of insulator surface conditions. Energies 2024;17(22):5595. http://dx.doi.org/10.3390/en17225595.

[46] Li X, Liu S, Wang Y, Zhou F, Wang L. Rapid detection method for insulation performance of vacuum glass based on ensemble learning. Eng Appl Artif Intell 2024;133:108106. http://dx.doi.org/10.1016/j.engappai.2024.108106.

[47] Moshrefi A, Nabki F. Advanced industrial fault detection: A comparative analysis of ultrasonic signal processing and ensemble machine learning techniques. Appl Sci 2024;14(15):6397. http://dx.doi.org/10.3390/app14156397.

[48] Zhang W, Li M, Chen J. Ultrasonic guided wave based discharge crack detection for composite insulators. In: Proceedings of the 2023 international conference on advanced materials and intelligent manufacturing. ICAMIM, Springer; 2023, p. 345–56. http://dx.doi.org/10.1007/978-981-97-8780-7_27.

[49] Stefenon SF, Meyer LH, Nied A, Klaar ACR, Yow KC. Electrical insulator fault forecasting based on a wavelet neuro-fuzzy system. Energies 2020;13(2):484. http://dx.doi.org/10.3390/en13020484.

[50] Stefenon SF, Americo JP, Meyer LH, Grebogi RB, Nied A. Analysis of the electric field in porcelain pin-type insulators via finite elements software. IEEE Lat Am Trans 2018;16(10):2505–12. http://dx.doi.org/10.1109/TLA.2018.8795129.

[51] IEC. Artificial pollution tests on high-voltage ceramic and glass insulators to be used on A.C. systems. In: International electrotechnical commission (IEC): 60507, vol. 3, Geneva, Switzerland: IEC; 2013, p. 1–77.

[52] Lu Y, Jin Q, Zhao N, Tian C, Cao Q. Research on load design for large containership based on nonlinear springing analysis in frequency domain. Results Eng 2025;27:106582. http://dx.doi.org/10.1016/j.rineng.2025.106582.

[53] Li M, Zhao F, Wang H, Yang Y, Dong X, et al. Visual recognition of icing risk on overhead transmission lines based on improved fuzzy wavelet domain. Results Eng 2025;28:107126. http://dx.doi.org/10.1016/j.rineng.2025.107126.

[54] Jwo DJ, Chang WY, Wu IH. Windowing techniques, the welch method for improvement of power spectrum estimation. Comput Mater Contin 2021;67(3):3983. http://dx.doi.org/10.32604/cmc.2021.014752.

[55] Yang Z, Yang G, Yang L, Zhang Q. A reconstruction method for graph signals based on the power spectral density estimation. Digit Signal Process 2022;122:103347. http://dx.doi.org/10.1016/j.dsp.2021.103347.

[56] Bazi S, Benzid R, Bazi Y, Said Nait-Said M, Mahmoud Al Rahhal M. Detection of induction motor incipient short circuit faults based on continuous wavelet transform with higher-order statistics. IEEE Access 2025;13:182705–16. http://dx.doi.org/10.1109/ACCESS.2025.3620538.

[57] Esquivel-Hernández J, Martínez-Manuel R, Maldonado-Hurtado D, Valentín-Coronado LM, Barrera D, Sales S. Interferometric temperature fiber sensor based on Hilbert transform. IEEE Photonics Technol Lett 2025;37(1):41–4. http://dx.doi.org/10.1109/LPT.2024.3504299.

[58] Mian Z, Deng X, Dong X, Tian Y, Cao T, Chen K, Al Jaber T. A literature review of fault diagnosis based on ensemble learning. Eng Appl Artif Intell 2024;127:107357. http://dx.doi.org/10.1016/j.engappai.2023.107357.

[59] Kim H, Lim Y. Bootstrap aggregated classification for sparse functional data. J Appl Stat 2022;49(8):2052–63. http://dx.doi.org/10.1080/02664763.2021.1889997.

[60] Ribeiro MHDM, da Silva RG, Moreno SR, Canton C, Larcher JHK, Stefenon SF, Mariani VC, dos Santos Coelho L. Variational mode decomposition and bagging extreme learning machine with multi-objective optimization for wind power forecasting. Appl Intell 2024;54:3119–34. http://dx.doi.org/10.1007/s10489-024-05331-2.

[61] Sibindi R, Mwangi RW, Waititu AG. A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices. Eng Rep 2023;5(4):e12599. http://dx.doi.org/10.1002/eng2.12599.

[62] Mienye ID, Sun Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. IEEE Access 2022;10:99129–49. http://dx.doi.org/10.1109/ACCESS.2022.3207287.

[63] Swarna K, Vinayagam A, Ananth MBJ, Kumar PV, Veerasamy V, Radhakrishnan P. A KNN based random subspace ensemble classifier for detection and discrimination of high impedance fault in PV integrated power network. Measurement 2022;187:110333. http://dx.doi.org/10.1016/j.measurement.2021.110333.

[64] Osman H, Azab A, AlGhazi A, Duffuaa S, Baki F. Forecasting electric vehicle charging loads using random forest and gene expression programming ensemble models. Results Eng 2025;28:108369. http://dx.doi.org/10.1016/j.rineng.2025.108369.

[65] Che Z, Peng C, Wang C, Wang J. A novel integrated TDLAVOA-XGBoost model for tool wear prediction in lathe and milling operations. Results Eng 2025;27:105984. http://dx.doi.org/10.1016/j.rineng.2025.105984.

[66] Zou P, Zhang Z, Yang S, Wang J, Wang K, Kou L, Nie H. A fault diagnosis method for converter transformers of UHVDC converter station using LightGBM-BTCN-MCAM with dissolved gas analysis. Results Eng 2025;28:108362. http://dx.doi.org/10.1016/j.rineng.2025.108362.

[67] Biswas P, Samanta T. Anomaly detection using ensemble random forest in wireless sensor network. Int J Inf Technol 2021;13(5):2043–52. http://dx.doi.org/10.1007/s41870-021-00717-8.

[68] Ji S, Liu K, Ling B, Li J, Wang J, Ma X. A study on improved random forest-based anomaly detection of regional tariff data under distributed photovoltaic access. Results Eng 2025;27:105920. http://dx.doi.org/10.1016/j.rineng.2025.105920.

[69] Trizoglou P, Liu X, Lin Z. Fault detection by an ensemble framework of extreme gradient boosting (XGBoost) in the operation of offshore wind turbines. Renew Energy 2021;179:945–62. http://dx.doi.org/10.1016/j.renene.2021.07.085.

[70] Gu Q, Sun W, Li X, Jiang S, Tian J. A new ensemble classification approach based on rotation forest and LightGBM. Neural Comput Appl 2023;35(15):11287–308. http://dx.doi.org/10.1007/s00521-023-08297-3.

[71] Amiri-Ramsheh B, Larestani A, Atashrouz S, Nasirzadeh E, Essakhraoui M, Abedi A, Ostadhassan M, Mohaddespour A, Hemmati-Sarapardeh A. Toward accurate prediction of carbon dioxide (CO2) compressibility factor using tree-based intelligent schemes (XGBoost and LightGBM) and equations of state. Results Eng 2025;25:104035. http://dx.doi.org/10.1016/j.rineng.2025.104035.

[72] Niu Y, Fan X, Xue R. A proximal policy optimisation algorithm-based algorithm for cardiovascular disorders detection. J Med Eng Technol 2025;49:45–64. http://dx.doi.org/10.1080/03091902.2025.2471332.

[73] Klaar ACR, Stefenon SF, Seman LO, Mariani VC, Coelho LS. Structure optimization of ensemble learning methods and seasonal decomposition approaches to energy price forecasting in Latin America: A case study about Mexico. Energies 2023;16(7):3184. http://dx.doi.org/10.3390/en16073184.

[74] Jacinto E, Martinez F, Martinez F. Navigation of autonomous vehicles using reinforcement learning with generalized advantage estimation. Int J Adv Comput Sci Appl 2023;14(1):954–9. http://dx.doi.org/10.14569/IJACSA.2023.01401103.

[75] Joshi T, Kodamana H, Kandath H, Kaisare N. TASAC: A twin-actor reinforcement learning framework with a stochastic policy with an application to batch process control. Control Eng Pract 2023;134:105462. http://dx.doi.org/10.1016/j.conengprac.2023.105462.