A novel Neutrosophic Transformer for handling imbalanced data

WU, Xingtao, DING, Yunfei, WANG, Lina, DING, Dong and ZHANG, Hongwei <http://orcid.org/0000-0002-7718-021X>

Journal of King Saud University Computer and Information Sciences https://doi.org/10.1007/s44443-026-00664-z

Article in Press

A novel Neutrosophic Transformer for handling imbalanced data

Received: 03 Jan 2026

Accepted: 10 Mar 2026

Published online: 23 March 2026

Cite this article as: Wu, X., Ding, Y., Wang, L. *et al.* A novel Neutrosophic Transformer for handling imbalanced data. *J. King Saud Univ. Comput. Inf. Sci.* (2026). https://doi.org/10.1007/s44443-026-00664-z

Xingtao Wu, Yunfei Ding, Lina Wang, Dong Ding & Hongwei Zhang

A Novel Neutrosophic Transformer for Handling Imbalanced Data

Abstract

In complex engineering environments, the majority of collected data predominantly comprises samples from normal operating conditions, leading to highly imbalanced datasets characterized by a scarcity of minority-class samples. Meanwhile, these data are inevitably contaminated by noise that masks subtle characteristic patterns, further complicating classification and recognition tasks. Traditional Transformer models exhibit limitations in capturing local features, sensitivity to data volume, difficulty in extracting hierarchical features, and susceptibility to overfitting. To overcome these limitations, this work proposes a novel Neutrosophic Transformer. By integrating a neutrosophic input representation and a neutrosophic feature extraction head, the model facilitates effective information exchange among diverse data components, including noise versus signal and minority vs majority samples. Consequently, the proposed architecture not only captures global contextual features but also preserves local and fine-grained details, enabling precise feature extraction. Furthermore, we introduce a Kernel Neutrosophic K-Nearest Neighbors (KNKNN) model designed to handle nonlinear data distributions in the extracted feature space. It is integrated with the proposed Neutrosophic transformer to form a unified framework that delivers robust and accurate classification. Comparative experiments on three real-world imbalanced engineering datasets demonstrate that the proposed method surpasses comparative approaches in feature extraction capability, classification accuracy, robustness, and generalization. Moreover, it achieves significant improvements in overall analytical performance.

Keywords: Neutrosophic Transformer, Deep Learning, Kernel Neutrosophic K-Nearest Neighbors, Fault diagnosis, Dynamic Threshold Mechanism, Attention Residual Refinement Mechanism

1. Introduction

Data-driven condition monitoring and fault classification has emerged as the dominant paradigm in modern engineering applications, leveraging sensor data to identify anomalies without resorting to expensive physical models[1, 2]. In complex industrial settings, this task typically comprises two main phases: feature extraction and classification. Traditional approaches rely on signal processing techniques for feature extraction, such as time-domain statistical features, frequency analysis, and time-frequency transformations, followed by conventional classifiers like support vector machines and decision trees. However, these methods often struggle with the inherent challenges of real-world operational data, including severe class imbalance where normal operation samples vastly outnumber anomalous instances, and persistent noise contamination from environmental interference and sensor inaccuracies.

The advent of deep learning has revolutionized this domain by enabling end-to-end learning from raw data. Deep neural networks can automatically learn discriminative features, alleviating the need for manual feature engineering. Nevertheless, most deep learning models optimize for overall accuracy under the implicit assumption of balanced distributions. Under severe imbalance, the cross-entropy loss is dominated by the majority-class, driving the decision boundary away from the minority support region and producing deceptively high accuracy while minority recall deteriorates[3]. Although techniques such as resampling, cost-sensitive reweighting and synthetic data augmentation have been widely adopted[4], their efficacy plateaus when dealing with complex noise patterns and difficult boundary cases[5].

Recent studies have explicitly addressed noise, class overlap, and boundary ambiguity in imbalanced datasets. For instance, Newton's cooling law based weighted oversampling (NCLWO) has been proposed for imbalanced datasets with feature noise [6], and a Newton cooling theorem based local overlapping regions cleaning and oversampling approach has been introduced in [7]. Other methods include a local entropy-adversarial oversampling method[8] and a semi-supervised local entropy-decayed oversampling approach[9]. These advanced sampling-based methods demonstrate improved handling of noise and boundary ambiguity. However, they primarily operate at the data level,

requiring careful tuning of sampling strategies and cost matrices, and may not fully capture the inherent uncertainty and indeterminacy in noisy engineering environments from a model-level perspective. Moreover, as data-level approaches, they may not be optimal for engineering applications requiring real-time processing and adaptive noise handling. To address these limitations, we propose a fundamentally different approach that operates at the model architecture level. Our method introduces model-level uncertainty modeling through neutrosophic theory, which directly embeds uncertainty representation within the deep learning architecture itself. This distinction is crucial for engineering applications where real-time processing and adaptive noise handling are required, enabling our approach to address imbalance and noise adaptively during both feature extraction and classification, without external data preprocessing.

Recently, Transformer[10] architectures have demonstrated remarkable success in feature extraction across various domains, owing to their self-attention mechanisms that capture long-range dependencies and global contextual information. In diagnosis applications, Transformers have shown superior capability in extracting discriminative features from time-series sensor data, effectively modeling temporal relationships that are crucial for identifying incipient faults. Notable examples include a time frequency Transformer framework [11], a hybrid model based on convolutional neural networks and Transformer for diagnosing unbalanced faults in rotating machinery [12], a dual-perspective time series Transformer for space-power-system diagnosis [13], and Transformer-based load forecasting [14]. However, standard Transformer models lack explicit mechanisms to handle the uncertainty and indeterminacy inherent in noisy engineering environments, and their performance remains susceptible to class imbalance and data quality issues [15].

To address the uncertainty modeling challenge, neutrosophic theory has been integrated with deep learning architectures. Sert et al. [16] incorporated the neutrosophic triplet into a neutrosophic entropy loss function for image classification tasks. Özyurt et al.[17] proposed a neutrosophic set - expert maximum fuzzy-sure entropy approach for brain tumor image classification. Moreover, neutrosophic Transformers have emerged, such as NTrans Net which employs multi-scale neutrosophic uncertainty guidance for indoor depth completion [18]. While these approaches incorporate uncertainty modeling, they are primarily designed for specific domains (e.g., image processing) and lack explicit mechanisms to handle severe class imbalance and complex noise patterns in industrial condition monitoring. They also do not integrate dynamic threshold adjustment or attention residual refinement tailored for imbalanced data distributions.

The k-nearest neighbour (KNN) algorithm stands out as a remarkably versatile and straightforward classification technique with extensive applications across various domains[19, 20], including fault detection in induction motors[21], rotating rectifier diagnosis[22], and integration with methods such as SVM[23] and deep neural networks[24]. However, KNN faces challenges with imbalanced data and noisy environments, i.e. majority voting biases toward dominant classes and outliers can degrade performance[25, 26]. To address these challenges, the Fuzzy K-Nearest Neighbour (FKNN) algorithm[27] was developed, incorporating fuzzy set theory to enable samples to associate with multiple classes at different membership levels[28], thereby providing some robustness to noise and class overlap[29–31]. FKNN still struggles with highly imbalanced data distributions and complex noise patterns, as it lacks explicit mechanisms to handle the inherent indeterminacy and inconsistency present in real-world engineering contexts. Neutrosophic K-Nearest Neighbour (NKNN) algorithm[32] extends FKNN by incorporating truth, falsity, and indeterminacy memberships, making it well suited for uncertain, imprecise, and imbalanced fault diagnosis scenarios[33]. However, current NKNN approaches still struggle to model complex non-linear relationships.

To address these gaps, we enhance Transformer performance while leveraging neutrosophic logic to manage uncertainty.

First, the issue of data uncertainty and noise is prevalent in real-world engineering data. The original Transformer model struggles to effectively handle such uncertain and noisy inputs, which can lead to inaccurate feature extraction and subsequent misclassification[34]. Therefore, refined neutrosophic parameters are introduced. These parameters facilitate a more sophisticated depiction of data uncertainty. Unlike traditional approaches that may oversimplify uncertain information, this refined parameterization allows the model to distinguish between different degrees of uncertainty with greater precision, thereby enhancing feature quality and improving classification reliability.

A second major challenge arises from severe class imbalance, where the number of majority-class (e.g., normal operation) samples often far exceeds that of minority-class samples. The standard Transformer model often exhibits a predisposition favoring the majority-class, resulting in poor recognition of minority class instances, which are crucial for identifying critical states[35]. A dynamic threshold mechanism is introduced to counteract this bias. It adaptively

adjusts the attention weights based on the uncertainty in the data. Therefore, the model can place greater emphasis on the features that are more reliable and informative, even if they belong to the minority class. This dynamic adjustment prevents the model from being misled by the noisy or irrelevant features that may dominate in imbalanced datasets, ultimately improving the model's robustness and accuracy for minority class recognition.

Third, the Transformer model can sometimes suffer from the loss of important fine-grained information during the attention process. This is particularly problematic when subtle variations in the data are indicative of significant state changes[36]. To mitigate this issue, an attention residual refinement mechanism is incorporated. This mechanism retains residual connections in the attention process, ensuring that important information is preserved and not lost. By combining the refined attention maps with the original input, the model can utilize both the detailed attention information and the raw feature characteristics. This fusion of information results in a more comprehensive and robust feature representation, which is crucial for accurate classification under complex conditions.

These three improvements collectively enhance the Neutrosophic Transformer model's ability to handle the key challenges of imbalanced and noisy real-world data, providing a more reliable and efficient solution compared to the original Transformer architecture. In contrast to these existing neutrosophic deep learning approaches, our Neutrosophic Transformer introduces several key innovations: (1) a novel neutrosophic input representation specifically designed for imbalanced time-series sensor data, (2) a dynamic threshold mechanism that adaptively adjusts classification boundaries based on sample uncertainty, (3) an attention residual refinement module that preserves fine-grained minority class features, and (4) integration with a kernelized neutrosophic KNN classifier to handle nonlinear distributions. Unlike other methods focus on depth completion, our framework is specifically tailored for industrial fault diagnosis under severe class imbalance and noise.

Furthermore, to enhance the NKNN's capability in handling non-linear data, the Kernel Neutrosophic K-Nearest Neighbour (KNKNN) algorithm is proposed in this study. By integrating kernel methods into the NKNN framework, KNKNN excels at modeling complex, non-linear patterns within datasets. The kernel function maps the original data into a high-dimensional feature space where non-linear relationships become more linearly separable[37]. This integration not only improves classification precision but also strengthens the model's resilience to noise. The neutrosophic parameters within KNKNN retain the capability to model uncertain and imprecise information, making the algorithm particularly suited for real-world classification tasks characterized by highly variable data quality.

The study's contributions are as follows:

(i). To address the limitations of Transformer classifiers when handling imbalanced datasets, a novel deep learning model "Neutrosophic Transformer" is proposed.

(ii). A Kernel Neutrosophic KNN model is presented by incorporating kernel method into the NKNN model, which has effectively enhanced the model's classification performance and robustness against noise for non-linear data.

(iii). A hybrid model based on the Neutrosophic Transformer and the Kernel Neutrosophic K-Nearest Neighbor is developed and subsequently applied to handle the real-world challenge of imbalanced data.

The remainder of this paper is organized as follows. Section 2 reviews neutrosophic sets, Transformer theory, and neutrosophic KNN. Section 3 details the proposed Neutrosophic Transformer architecture, including the dynamic threshold mechanism and attention residual refinement, followed by the Kernel Neutrosophic KNN classifier and the overall fault diagnosis framework. Section 4 presents the experimental setup, results on three real-world datasets, ablation studies, complexity analysis, and noise injection experiments. Finally, Section 5 concludes and outlines directions for future work.

2. Basic theories

The technical terms and symbols used in this paper are as shown in the Table 1.

2.1. Neutrosophic theory

It is a general truth that the natural order of events encompasses inherent variability. Classical methods exclude real-world uncertainty, thereby sometimes lacking rationality. Fuzzy sets are widely adopted by scientists to tackle issues associated with uncertainty.

Table 1: Summary of key model notations

Symbol	Description	Symbol	Description
$\mathbf{x}_i$	Feature vector of sample i	τ_i	Dynamic threshold for sample i
y_i	Class label of sample i	base	Global base threshold
d_{model}	Embedding dimension	λ	Dynamic amplitude factor
L	Number of Transformer blocks	c_i	Confidence score
$\mathbf{Q}, \mathbf{K}, \mathbf{V}$	Attention matrices	k, M	Neighbors and classes count
$\mathbf{h}_{\text{CLS}}$	CLS token hidden state	δ, m, σ	Noise, weight, kernel parameters
$\hat{\mathbf{p}}$	Predicted probabilities	c_j	Class centroid
P	Posterior probability	c_{imax}^{-2}	Nearest neutrosophic point
T_i, I_i, F_i	Neutrosophic triple	$K(d)$	Kernel function
T_i^n, I_i^n, F_i^n	Normalized triple	$\mathbf{w} = [w_T, w_F, w_I]^T$	Attention weights
P_{adj}	Adjusted probability	b	Scalar bias
α, β, γ	Neutrosophic weights	$\boldsymbol{\theta}$, Θ	Parameter sets

Definition 1. [38] Suppose X is given, then the fuzzy set $\tilde{A}$ in X is a set of ordered pairs $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}$, where $\mu_{\tilde{A}}(x)$ is the membership function of the fuzzy set $\tilde{A}$. The membership function maps each member of X to a degree of membership that is a value between 0 and 1.

Although fuzzy sets demonstrate a degree of effectiveness in handling the uncertainty arising from fuzziness within samples, They remain ill-equipped to address uncertainties extensively encountered in real-world problems, especially under conditions of scarce information. To tackle this issue, Atanasov[39] developed an expanded framework for fuzzy sets, known as intuitionistic fuzzy sets.

Definition 2. [39] An intuitionistic fuzzy set in U is a set like A in which two degrees are attributed to each member $u \in U$; one is the "degree of membership," and the other is the "degree of non-membership". If the membership function is $\mu(A) : X \to [0, 1]$ and the non-membership function is $\nu(A) : X \to [0, 1]$, then the indeterminacy value of A is $1 - \mu(A) - \nu(A)$. The condition $0 \leq \mu(A) + \nu(A) \leq 1$ is always satisfied.

The approach defines membership and non-membership degrees while also accounting for uncertainty and contradiction. Consequently, the functions $T(x)$, $I(x)$, and $F(x)$, which represent true, uncertain, and false membership degrees respectively, are all integrated into consideration.

Definition 3. [40] Let X be a universe of discourse. A neutrosophic set A in X is characterized by three membership functions: truth-membership $T_A(x)$, indeterminacy-membership $I_A(x)$, and falsity-membership $F_A(x)$, such that for each $x \in X$:

$$T_A(x) \in [0, 1], \quad I_A(x) \in [0, 1], \quad F_A(x) \in [0, 1]$$

and

$$0 \leq T_A(x) + I_A(x) + F_A(x) \leq 3.$$

The triplet $(T_A(x), I_A(x), F_A(x))$ is called a neutrosophic component, representing the degree of truth, uncertainty, and falsity of x belonging to A, respectively.

The original T, F, I are computed as Eq.(1-3):

$$T = \frac{P - P_{min}}{P_{max} - P_{min}}, \tag{1}$$

$$F = 1 - T, \tag{2}$$

$$I = \sqrt{T^2 + F^2}. \tag{3}$$

where P denotes the prediction matrix output by the model.

2.2. Transformer

Transformer[10] architectures have been successfully adapted to diagnostic classification tasks. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with $\mathbf{x}_i \in \mathbb{R}^d$ and label $y_i \in \{0, 1, \ldots, K-1\}$, the input $\mathbf{x}_i$ is first projected into a d_{model}-dimensional embedding space through Eq.(4)

$$\mathbf{E}_i = \mathbf{W}_{\text{emb}}\, \mathbf{x}_i + \mathbf{b}_{\text{emb}}, \tag{4}$$

where $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{d_{\text{model}} \times d}$ and $\mathbf{b}_{\text{emb}} \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters. A positional encoding $\mathbf{P}$ is then added to $\mathbf{E}_i$ to retain the order information, resulting in Eq.(5):

$$\mathbf{H}_i = \mathbf{E}_i + \mathbf{P}_{\text{pos}(i)}. \tag{5}$$

The encoded features traverse L Transformer blocks, each integrating multi-head self-attention and feed-forward networks. The self-attention mechanism is defined as Eq.(6):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}. \tag{6}$$

Each Transformer block includes a feed-forward network with residual connections to stabilize training:

$$\text{FFN}(\mathbf{X}) = \max(0, \mathbf{X}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{7}$$

For classification, a [CLS] token is added to the input sequence. Its final hidden state is used to predict class probabilities via a softmax layer shown in Eq.(8):

$$\hat{\mathbf{p}} = \text{softmax}(\mathbf{W}_{\text{cls}}\mathbf{h}_{\text{CLS}} + \mathbf{b}_{\text{cls}}) \tag{8}$$

During training, the standard cross-entropy loss is defined as Eq.(9):

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{k=0}^{K-1} \mathbf{1}_{\{y_i=k\}} \log \hat{p}_{i,k} \tag{9}$$

It is minimized using the AdamW optimizer. The inference process is efficient, with a time complexity of $O(Ld_{\text{model}}^2)$ per sample, typically completing in less than 1 ms on modern hardware.

2.3. Neutrosophic K-Nearest Neighbor

At the close of the twentieth century, research innovatively developed the theory of Neutrosophy from a philosophical perspective. Neutrosophy offers a more comprehensive means of expressing uncertain information within decision-making problems[40]. The superiority of Neutrosophy lies in its capacity for more precise articulation of uncertainty, a characteristic that aligns more closely with human modes of thought.

The fundamental tenet of neutrosophy is that any proposition decomposes into three elements: ”truth” (T), ”uncertainty” (I), and ”falsehood” (F). In concrete analysis, neutrosophic logic extends beyond the confines of fuzzy logic—where an entity classified as type A receives an absolute ”true” or ”false” judgement. It further explores the grey zone between the extremes of ”true” and ”false”, introducing a continuous and neutral assessment dimension. This extension enables the logical system to describe the complexity and uncertainty of the real world more precisely, thereby providing more refined and contextually relevant decision support and model construction.

By incorporating the theory of neutrosophy entities, the FKNN algorithm refines membership information into three tiers: ”true”, ”false”, and ”uncertainty”. This innovation not only enables the algorithm to adjust sample feature weights with greater precision but also quantifies and emphasises the varying significance of features in driving classification outcomes. It effectively addresses the limitations inherent in traditional KNN algorithms when handling feature weight allocation. The first step in the training phase is to determine the class centroids of the sample set. These centroids serve as the foundation for subsequent membership degree calculations. Then, the ”true” and ”false”

membership degrees T_i and F_i, along with the uncertainty degree I_i, are computed using the following Eq.(10-12):

$$T_{ij} = \frac{\left(x_i - c_j\right)^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^{M}\left(x_i - c_j\right)^{-\left(\frac{2}{m-1}\right)} + \left(x_i - c_{imax}^{-}\right)^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \tag{10}$$

$$F_i = \frac{(\delta)^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^{M}\left(x_i - c_j\right)^{-\left(\frac{2}{m-1}\right)} + \left(x_i - c_{imax}^{-}\right)^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \tag{11}$$

$$I_i = \frac{\left(x_i - c_{imax}^{-}\right)^{-\left(\frac{2}{m-1}\right)}}{\sum_{j=1}^{M}\left(x_i - c_j\right)^{-\left(\frac{2}{m-1}\right)} + \left(x_i - c_{imax}^{-}\right)^{-\left(\frac{2}{m-1}\right)} + \delta^{-\left(\frac{2}{m-1}\right)}} \tag{12}$$

where δ is a parameter used to adjust the noise level, c_j signifies the centroid of the class samples, and M represents the number of classes. The distance from a sample to the nearest neutrosophic point (the midpoint between two class centroids) is denoted as c_{imax}^{-}, while m serves as the weighting coefficient.

In addition, the k value is set, and the Manhattan distance, known for its stability, is used to compute the similarity. The formula for calculating similarity is as follows:

$$d_i = \left[\frac{1}{\sum_{i=1}^{p}\left|x_{s_i} - x_{t_i}\right|}\right]^{2/q-1} \tag{13}$$

where, p defines the number of feature dimensions, i describes the sample's attributes, d_i reflects the similarity with the i-th neighbor x_i, and q adjusts the weighting of similarity.

Finally, the class membership degrees are calculated to perform classification. The formulas are given below:

$$\mu_j(x_s) = \frac{\sum_{i=1}^{k} d_i(T_{ij} + I_i - F_i)}{\sum_{i=1}^{k} d_i} \tag{14}$$

$$c(x_s) = \arg\max\left[u_j(x_s)\right] \tag{15}$$

where, $\mu_j(x_s)$ represents the class membership degree for each class sample, and $c(x_s)$ indicates the class with the highest membership degree for the sample x_s.

3. Proposed Method

A novel diagnosis framework integrating Neutrosophic Transformer and Kernel Neutrosophic K-Nearest Neighbour is introduced. The specific modelling process is as follows:

(i) After removing the date feature from the raw data, standardisation is applied to all data.

(ii) Feature extraction employs many contrasting methods to mitigate computational overload and suboptimal training outcomes caused by excessive dimensionality.

(iii) The KNKNN model performs state classification on the reduced-dimensional dataset, with comparative experiments conducted against established models for validation.

The framework is illustrated in Figure 1.

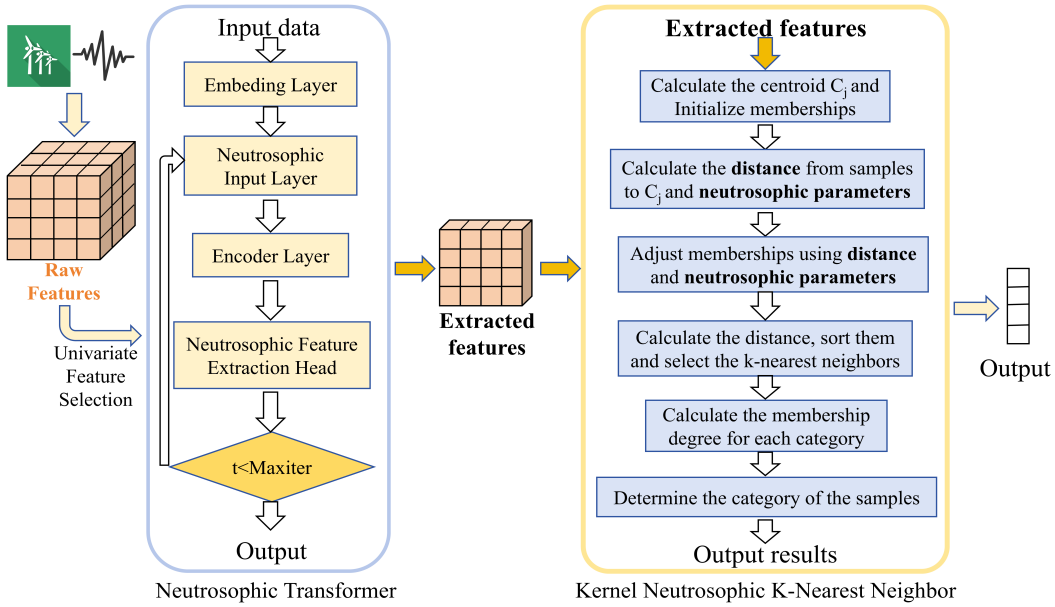

Figure 1: Structure of proposed method.

The proposed framework adopts a modular design that decouples feature extraction from classification. This design is motivated by four considerations. First, the Neutrosophic Transformer learns high-level discriminative representations from raw sensor data, capturing both global context and fine-grained details. Second, the KNKNN classifier leverages kernelized distances and neutrosophic uncertainty modeling to handle nonlinear patterns, noise, and borderline samples—challenges that a standard linear classifier may struggle with. Third, the separation allows independent optimization and replacement of each module, facilitating ablation studies to assess individual contributions. Moreover, KNKNN requires no additional training and is robust in small-sample and high-noise scenarios, complementing the data-driven feature extractor. Thus, the hybrid approach combines the strengths of deep representation learning with the robustness of kernel-based uncertainty-aware classification, which is particularly beneficial for imbalanced and noisy industrial data.

3.1. Neutrosophic Transformer

Despite their formidable capabilities, Transformer often falter in two critical scenarios: (1) When datasets are imbalanced, minority class samples tend to be overwhelmed by the majority-class; (2) In situations of cognitive uncertainty, the model's posterior probability P hovers around 0.5, rendering decisions unreliable. Therefore, the neutrosophic theory is incorporated into Transformer to address these shortcomings. The neutrosophic triplet (T, I, F) used in our Neutrosophic Transformer adheres to the formal definition in Definition 3. Specifically, we compute these components from the model's internal representations to quantify the confidence, uncertainty, and contradiction levels of each sample. The specific modelling process for the Neutrosophic Transformer is illustrated in Figure 2 below:

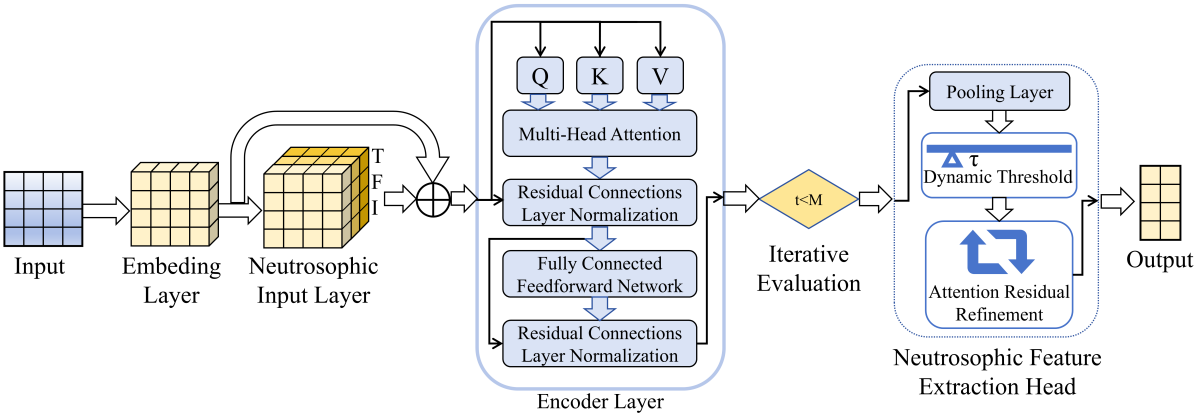

Figure 2: Structure of Neutrosophic Transformer.

The Neutrosophic input layer in the diagram is employed to generate the parameters of the Neutrosophic certainty T, falsity F, and uncertainty I. During each iteration, P is adjusted based on T, F, and I. According to Eqs (1),(2) and (3), the three quantities T, F, and I are derived through simple minimum-maximum scaling, algebraic complement coding, and Euclidean operations. Although they lie in [0,1], the definitions are purely geometric and ignore the model's posterior confidence, making the uncertainty term sensitive to extreme feature values and unable to reflect genuine prediction hesitation.

To mitigate the impact of outliers on neutrosophic features while guaranteeing numerical stability across mini-batches, a lightweight two-step normalization tailored is adopted to the neutrosophic datasets. Because the raw features are already compressed to the range [0, 1] by the built-in Min–Max operator in the original code, this mapping is simply retained as Eq.(16,17):

$$T_i^{\mathrm{n}} = \frac{T_i - \min T}{\max T - \min T}, \tag{16}$$

$$F_i^{\mathrm{n}} = \frac{F_i - \min F}{\max F - \min F}. \tag{17}$$

To suppress extreme values of I_{unc} without distorting the original scale, a smoothing saturation process is employed as Eq.(18,19):

$$I_{\mathrm{clip}}(i) = \tanh \Delta(k\, I_{\mathrm{unc}}(i)), \tag{18}$$

$$I_i^{\mathrm{n}} = \frac{I_{\mathrm{clip}}(i) + 1}{2}. \tag{19}$$

The constant $k = 2$ controls the saturation steepness and is fixed for all experiments. After clipping, all three normalized vectors $T^{\mathrm{n}}, F^{\mathrm{n}}, I^{\mathrm{n}} \in [0, 1]^{100\times 1}$ are column-wise re-aligned for subsequent dynamic-threshold computation. The conventional geometric definition captures only a priori feature distances and ignores the model's posterior confidence. To address this limitation, the uncertainty is redefined as Eq.(20) below:

$$I_{\mathrm{unc}} = 1 - 2\,|P - 0.5|\,, \tag{20}$$

where P denotes the posterior probability output by the Transformer. This formulation maps proximity to 0.5 into higher uncertainty, thereby quantifying the model's hesitation near the decision boundary and naturally constraining the term within [0, 1].

Otherwise, to overcome the disjoint optimization of a static threshold and post-processing, learnable weights α, β, and γ are introduced in stage 1. These weights inject T, F, and I_{unc} as residual terms directly into the model output as shown in Eq.(21):

$$P_{\mathrm{adj}} = P + \alpha T - \beta F - \gamma I_{\mathrm{unc}}. \tag{21}$$

The proposed design processes neutrosophic features into an end-to-end trainable residual. This structure supports

gradient back-propagation through the Transformer and influences the model's weight parameters, leading to the joint optimization of thresholds and weights.

The dynamic threshold mechanism forms part of the Neutrosophic Feature Extraction Head. It is introduced to replace the conventional fixed decision boundary $\tau = 0.5$. For each sample i, the dynamic threshold τ_i is defined as Eq.(22,23):

$$\tau_i = \text{base} + \lambda\,(0.5 - c_i), \tag{22}$$

$$c_i = \frac{T_i^{\text{n}} - F_i^{\text{n}} - I_i^{\text{n}} + 1}{2}, \tag{23}$$

where $c_i \in [0, 1]$ quantifies the model's confidence. A lower c_i raises τ_i, making the sample harder to classify as the positive class, and vice versa.

The dynamic threshold mechanism adaptively adjust the classification boundary for each sample based on its confidence. In traditional Transformers, the decision boundary is fixed at 0.5, which tends to bias toward the majority-class in imbalanced data. In our method, the confidence score c_i integrates the truth, falsity, and uncertainty degrees of each sample: the lower the confidence (i.e., the more uncertain or noise-like the sample), the higher the threshold τ_i, making the model more conservative in its prediction for that sample and avoiding misclassifying noisy or borderline samples as positive. Conversely, for high-confidence samples, the model is more inclined to assign a positive label. This mechanism enables the model to maintain stable performance on the majority-class while significantly enhancing sensitivity to minority (fault) samples.

The optimal operating point of the dynamic-threshold mechanism is determined by jointly tuning the set $\Theta = \{\text{base}, \lambda, \alpha, \beta, \gamma\}$ on the validation split. Table 2 shows the search space and step sizes.

Table 2: Search space for dynamic-threshold parameters.

Symbol	Description	Range	Step
base	Global base threshold	[0, 1]	0.02
λ	Dynamic amplitude factor	[0, 0.5]	0.05
α	Weight for T^{n}	[0, 1]	0.10
β	Weight for F^{n}	[0, 1]	0.10
γ	Weight for I^{n}	[0, 1]	0.10

The objective function is the macro F1-score:

$$F_1(\Theta) = \frac{2\,\text{TP}(\Theta)}{2\,\text{TP}(\Theta) + \text{FP}(\Theta) + \text{FN}(\Theta)}. \tag{24}$$

The optimization of hyperparameters for both the dynamic threshold mechanism and the attention residual refinement module is conducted using Bayesian optimization with a Matérn 5/2 kernel. This approach is selected due to its efficiency in exploring high-dimensional, non-convex parameter spaces and its ability to balance exploration and exploitation. The optimization process is constrained to 100 iterations for each parameter set, with each iteration requiring less than 0.3 seconds on a single CPU core. The expected-improvement acquisition function is employed to guide the search towards regions with higher expected performance gains. This systematic optimization not only ensures the reproducibility of our experimental outcomes but also provides clear guidance for practitioners aiming to apply the proposed framework to analogous engineering problems characterized by imbalanced and noisy data.

The attention residual refinement mechanism is part of the Neutrosophic Feature Extraction Head of the Neutrosophic Transformer. After the dynamic-threshold stage, a subset of samples still suffers from mis-classification. To address these errors without retraining the Transformer, a lightweight residual refinement module that utilizes the neutrosophic triplets is introduced as attention cues.

Let $\mathbf{x}_i = [T_i^{\text{n}},\ F_i^{\text{n}},\ I_i^{\text{n}}]^\top \in \mathbb{R}^3$ denote the normalized neutrosophic feature vector for sample i. The residual score is produced by a single linear layer followed by a bias term:

$$r_i = \mathbf{w}^\top \mathbf{x}_i + b, \tag{25}$$

where $\mathbf{w} = [w_T, w_F, w_I]^\top \in \mathbb{R}^3$ and $b \in \mathbb{R}$ are global learnable parameters. The refined posterior probability is then

$$\hat{p}_i^{(2)} = \sigma(\hat{p}_i^{(1)} + r_i), \tag{26}$$

with $\sigma(\cdot)$ the sigmoid function.

The attention residual refinement mechanism is inspired by the residual learning philosophy, aiming to prevent critical features from being diluted or lost during multi-layer attention propagation. In applications involving imbalanced and noisy data, certain subtle but discriminative patterns (e.g., weak characteristic signatures indicative of minority-class states) may be smoothed out by global attention. Our mechanism fine-tunes the preliminary prediction $\hat{p}_i^{(1)}$ by using the neutrosophic triplet $[T_i^n, F_i^n, I_i^n]$ as attention weights. This acts as a lightweight "feature corrector" that can selectively enhance or suppress contributions from specific feature channels according to the uncertainty structure of each sample, thereby improving the model's ability to capture subtle patterns and its overall classification robustness.

The parameters $\{\mathbf{w}, b\}$ are optimized on the validation set by maximizing the macro F1-score:

$$\mathcal{L}(\mathbf{w}, b) = \frac{2\,\mathrm{TP}(\mathbf{w}, b)}{2\,\mathrm{TP}(\mathbf{w}, b) + \mathrm{FP}(\mathbf{w}, b) + \mathrm{FN}(\mathbf{w}, b)}. \tag{27}$$

To determine the optimal attention weights and bias of the refinement module, the parameter vector is searched $\boldsymbol{\theta} = [w_T, w_F, w_I, b]^\top \in \mathbb{R}^4$ jointly on the validation set.

Table 3: Search space and step sizes for the attention residual refinement module.

Symbol	Meaning	Range	Step Size
w_T	Attention weight for T^{n}	$[-2, 2]$	0.1
w_F	Attention weight for F^{n}	$[-2, 2]$	0.1
w_I	Attention weight for I^{n}	$[-2, 2]$	0.1
b	Scalar bias term	$[-1, 1]$	0.05

The optimisation objective is the macro-F1 score in Eq.(28):

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{2\,\mathrm{TP}(\boldsymbol{\theta})}{2\,\mathrm{TP}(\boldsymbol{\theta}) + \mathrm{FP}(\boldsymbol{\theta}) + \mathrm{FN}(\boldsymbol{\theta})} \tag{28}$$

Table 3 summarises the search space and step sizes. A maximisation search is conducted using Bayesian optimisation under the conditions specified in Table 2. The Matérn 5/2 kernel function and an expectation maximisation approach are employed to obtain the policy.

The proposed Neutrosophic Transformer addresses the dual challenges of noise interference and class imbalance through its integrated architecture. Regarding noise handling, the neutrosophic triplet (T, F, I) provides a principled mechanism for quantifying uncertainty: the indeterminacy component I directly captures the ambiguity inherent in noisy samples. Formally, given an input sample $\mathbf{x}_i$, the model computes its neutrosophic representation as $\mathbf{n}_i = [T_i, F_i, I_i]^T$, where higher I_i values indicate greater uncertainty, typically corresponding to noisy or borderline instances. This representation allows the model to distinguish between reliable signal patterns and noise-contaminated features during feature extraction.

For minority-class samples, the dynamic threshold mechanism plays a crucial role. Traditional fixed-threshold classifiers tend to favor the majority-class due to its numerical dominance, causing minority samples to be misclassified as the majority-class. Our approach adaptively adjusts the decision boundary for each sample based on its confidence score $c_i = \frac{T_i^n - F_i^n - I_i^n + 1}{2}$. Samples with lower confidence (often minority-class or noisy samples) receive higher classification thresholds $\tau_i = \text{base} + \lambda(0.5 - c_i)$, making the model more conservative in assigning them to the positive class. Conversely, high-confidence samples are classified with lower thresholds, enabling more sensitive detection of minority-class patterns.

The attention residual refinement mechanism further enhances minority-class recognition by preserving fine-

grained discriminative features. In standard Transformer architectures, global attention mechanisms may dilute subtle patterns characteristic of minority-classes. Our residual correction, defined as $r_i = \mathbf{w}^\top \mathbf{n}_i + b$, leverages the neutrosophic triplet as attention weights to selectively amplify or suppress feature channels. This ensures that even weak but discriminative signatures from minority-class samples are preserved in the final feature representation $\hat{p}_i^{(2)} = \sigma(\hat{p}_i^{(1)} + r_i)$.

Collectively, these mechanisms establish an information exchange pathway between noise processing and class-imbalance handling. Noise-contaminated samples exhibit elevated I values, which in turn influence both the dynamic threshold (making classification more conservative) and the attention residual (reducing the impact of uncertain features). Minority-class samples benefit from lower confidence scores, which trigger threshold adjustments that counteract majority-class bias. This integrated approach enables simultaneous robustness to noise and sensitivity to minority-class, addressing two fundamental challenges in real-world imbalanced learning scenarios.

3.2. KNKNN

In the fields of machine learning and pattern recognition, data frequently exhibits intricate nonlinear structures, posing significant challenges to traditional models based on linear distance metrics. To enhance model performance and enable more effective capture of complex patterns and intrinsic structures within data, kernelisation methods have emerged. The essence of kernelisation is transforming original data into a high-dimensional space using kernel functions. Within this new space, data that is originally non-linearly separable may become linearly separable, thereby enabling models to better handle complex classification and regression tasks.

Distance kernelling represents a specific implementation of kernelling methods. By substituting traditional Euclidean or other linear distance metrics with kernelled distance measures, we can redefine similarity between sample points within the high-dimensional feature space. Kernelled distance metrics effectively capture complex non-linear patterns in data and boost the model's noise resistance. In our research, incorporating distance kernelling into the NKNN model aims to enhance its classification performance when handling complex datasets. This enables the model to more accurately reflect the membership relationships between sample points and class centres, thereby improving classification accuracy and reliability.

The kernel function is defined as equation below:

$$K(d) = 2 \cdot \left(1 - \exp\left(-\frac{d^2}{\sigma^2}\right)\right) \tag{29}$$

where d represents the original distance, and σ is the kernel width parameter.

The Gaussian kernel function is adopted in this study to map original distances into a high-dimensional space. This choice is motivated by two considerations: First, the Gaussian kernel possesses strong nonlinear mapping capabilities, transforming complex intertwined fault patterns in the original space into more separable structures in the high-dimensional space. Second, its exponential decay characteristic provides inherent robustness to noise and outliers, effectively suppressing the impact of impulse noise commonly found in sensor data on distance metrics. The parameter σ controls the smoothness of the mapping and can be optimized to adapt to the distribution characteristics of different datasets.

The calculation of the neutrosophic parameters after incorporating the kernel function are as follows:

$$T_i = \frac{\left(K(\|x_i - c_j\|)\right)^{-\frac{2}{m-1}}}{\sum_{j=1}^{M}\left(K(\|x_i - c_j\|)\right)^{-\frac{2}{m-1}} + \left(K(\|x_i - c_{i_{\max}}^-\|)\right)^{-\frac{2}{m-1}} + \delta^{-(\frac{2}{m-1})}} \tag{30}$$

$$I_i = \frac{\left(K(\|x_i - c_{i_{\max}}^-\|)\right)^{-\frac{2}{m-1}}}{\sum_{j=1}^{M}\left(K(\|x_i - c_j\|)\right)^{-\frac{2}{m-1}} + \left(K(\|x_i - c_{i_{\max}}^-\|)\right)^{-\frac{2}{m-1}} + \delta^{-(\frac{2}{m-1})}} \tag{31}$$

$$F_i = \frac{\delta^{-(\frac{2}{m-1})}}{\sum_{j=1}^{M}\left(K(\|x_i - c_j\|)\right)^{-\frac{2}{m-1}} + \left(K(\|x_i - c_{i_{\max}}^-\|)\right)^{-\frac{2}{m-1}} + \delta^{-(\frac{2}{m-1})}} \tag{32}$$

where c_j is the centroid of the j-th class; $c^-_{i_{\max}}$ is the distance from the sample to the nearest midpoint; m is the weighting exponent; σ is the kernel width parameter; δ is the parameter for adjusting the amount of noise; x_i is the i-th sample point.

To avoid issues arising from manual parameter tuning, the neighborhood size k in the KNKNN model is first determined using the Coati Optimization Algorithm (COA)[41] prior to experimentation. This algorithm searches for the optimal k value within the range [1, 20] by maximizing the cross-validation macro-F1 score, ensuring the model adapts to the specific characteristics of each dataset.

The pseudocode for KNKNN is shown in the Table 1.

Algorithm 1 Kernel Neutrosophic K-Nearest Neighbor (KNKNN)

Input: Training data $\mathcal{X}_{\text{train}}$, labels $\mathcal{Y}_{\text{train}}$, Test data $\mathcal{X}_{\text{test}}$, parameters k, σ, δ, m
Output: Predicted labels $\mathcal{Y}_{\text{pred}}$

Compute class centroids C_c for each class $c = 1, \ldots, M$
for each $\mathbf{x}_i \in \mathcal{X}_{\text{train}}$ **do**
    Compute T_i, I_i, F_i using Eqs. (30)-(32) with kernel distances
end for
for each $\mathbf{x}'_j \in \mathcal{X}_{\text{test}}$ **do**
    for each $\mathbf{x}_i \in \mathcal{X}_{\text{train}}$ **do**
        Compute kernel distance K_{ij} using Eq. (29)
    end for
    Select k nearest neighbors based on K_{ij}
    Compute class memberships μ_c using Eq. (14)
    Assign label $\hat{y}_j = \arg\max_c \mu_c$ {Eq. (15)}
    $\mathcal{Y}_{\text{pred}} \leftarrow \mathcal{Y}_{\text{pred}} \cup \{\hat{y}_j\}$
end for
return $\mathcal{Y}_{\text{pred}}$

4. Experiments

4.1. Case 1

4.1.1. Dataset description

The experiments utilized an AMD R7 processor with 3.20GHz speed, 16GB memory, and ran on the Windows 11 64-bit operating system. All models are done using the Matlab toolkit or programming.

The experimental data for this study, sourced from the Industrial Big Data Innovation Competition, encompasses operational data from turbines 15 and 21 between November 1, 2015, and January 1, 2016. Each turbine includes 20 features[42]. The noise in the dataset primarily consists of environmental noise and sensor noise.

The data underwent a preprocessing step where high-power samples (above 2KW) are removed. Subsequently, data alignment and aggregation are performed, including merging samples with the same timestamp and fusing blade-related metrics such as blade pitch angle, blade pitch speed, and pitch motor temperature. These steps resulted in a feature set of 20 attributes. The dataset details and attribute information are provided in Tables 4 and 5, respectively.

Table 4: The source and details of the turbine dataset

Turbine No.	**15**	**21**
Feature	20	20
Size	13607	5058
Modal	Time-series signal	
Positive Sample	2841	1274
Negative Sample	10766	3784
Imbalance Rate	1:3.79	1:2.97

Table 5: The corresponding attributes for fan SCADA data

No.	Feature name	No.	Feature name
1	Wind Speed	11	Acceleration in X Direction
2	Generator RPM	12	Acceleration in Y Direction
3	Output Power	13	Ambient Temperature
4	Wind Direction	14	Cabin Temperature
5	Wind Direction (25s)	15	1_ng5_tmp
6	Yaw Position	16	2_ng5_tmp
7	Yaw Rate	17	3_ng5_tmp
8	Average Pitch Angle	18	1_ng5_DC
9	Average Pitch Rate	19	2_ng5_DC
10	Average Pitch Motor temperature	20	3_ng5_DC

4.1.2. Feature extraction

To showcase the Neutrosophic Transformer's advantages in feature extraction, five comparative methods are employed: Random Forest (RF)[43], Principal Component Analysis (PCA)[44], Convolutional Neural Network 1D (Conv1D)[45], LSTM with Attention Mechanism (LSTM-Attention)[46] and Transformer. Finally, the datasets processed through these four feature extraction methods are experimented using the KNN model. All metrics used in this paper are macro average evaluation metrics. The specific values set of parameters during the experiment are shown in the Table 6. The training set and test set are divided in a 1:1 ratio.

For the wind turbine dataset (Case 1), each sample corresponds to a single time stamp consisting of 20 sensor readings. Although the original data are collected sequentially over time, we treat each sample independently because there is no explicit temporal dependency between adjacent samples in the constructed dataset. Consequently, all baseline models, including those capable of sequence modeling (e.g., LSTM-Attention and Transformer), process each sample as a feature vector without leveraging temporal order. The results are documented in Tables 7 and 8.

Table 6: Values set for control parameters of compared model

Model	Parameter	Value
RF	Trees	50
	Max Depth	5
	Leaf	20
PCA	Variance Retention Rate	90%
Conv1D	Kernel size	3
	num_channels	[32,64]
LSTM-Attention	hidden_dim	128
	attention_dim	64
Transformer/ Neutrosophic Transformer	MaxEpochs	50
	LearnRate	1e-5
	MiniBatchSize	32
KNN	k	3

Table 7: Results of the compared models for Fan No.15 and Fan No.21 in feature extraction experiment

Feature extraction methods	Fan No.15				Fan No.21			
	TP	FP	FN	TN	TP	FP	FN	TN
RF	183	610	1238	4773	121	303	516	1589
PCA	906	309	515	5074	475	121	162	1771
Conv1D	632	391	789	4992	324	162	313	1730
LSTM-Attention	992	172	429	5211	482	54	155	1838
Transformer	1143	134	278	5249	554	54	83	1838
Neutrosophic Transformer	1199	93	222	5290	580	28	57	1864

Table 8: Results of the compared models for Fan No.15 and Fan No.21 in feature extraction experiment

Feature extraction methods	Fan No.15				Fan No.21			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
RF	0.7284	0.5124	0.5077	0.5016	0.6762	0.5201	0.5149	0.5116
PCA	0.8789	0.8268	0.7901	0.8062	0.8881	0.8566	0.8409	0.8482
Conv1D	0.8266	0.7407	0.6861	0.7057	0.8122	0.7567	0.7115	0.7282
LSTM-Attention	0.9117	0.8881	0.8331	0.8565	0.9174	0.9107	0.8641	0.8840
Transformer	0.9394	0.9224	0.8897	0.9048	0.9458	0.9340	0.9206	0.9270
Neutrosophic Transformer	0.9537	0.9439	0.9132	0.9275	0.9664	0.9621	0.9479	0.9547

Tables 7 and 8 summarize the feature extraction results. On Fan 15, the Neutrosophic Transformer outperformed the standard Transformer across all metrics, with gains of 1.43% in accuracy, 2.15% in precision, 2.35% in recall, and 2.27% in F1-score. Even larger improvements are observed on Fan 21, where accuracy, precision, recall, and F1-score increased by 2.06%, 2.81%, 2.73%, and 2.77%, respectively.

Additionally, in Fan No. 15, the Neutrosophic Transformer outperformed the best-performing LSTM-Attention model in the comparison set by 4.20%, 5.58%, 8.01%, and 7.10% across its four evaluation metrics. For Wind Turbine No. 21, its four metrics outperformed the best-performing LSTM-Attention model in the comparison set by 4.90%, 5.14%, 8.38%, and 7.07%, respectively. Overall, the Neutrosophic Transformer exhibits distinct advantages as a feature extraction method and can be considered an effective approach for this purpose.

Through 10 repeated experiments, the Neutrosophic Transformer demonstrated statistically significant advantages over other methods. As shown in the Table 9, on both the Fan 15 and Fan 21 datasets, the Neutrosophic Transformer demonstrated highly significant (p ¡ 0.001) superiority across all four metrics: accuracy, precision, recall, and F1 score. Compared to traditional methods (Random Forest, PCA), p-values are consistently below 1e-15; compared to deep learning benchmark models (Original Transformer, Conv1D, LSTM-Attention), p-values are consistently below 1e-6. These results indicate that the improvements in feature extraction and classification performance achieved by the Neutrosophic Transformer are statistically reliable and not due to chance.

Table 9: Performance Comparison of Neutrosophic Transformer and Other Methods on the fan datasets p-value

Neutrosophic Transformer	Fan No.15				Fan No.21			
vs Comparative Methods	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
RF	8.59e-22	1.07e-17	1.55e-27	3.62e-24	5.16e-22	1.63e-25	1.21e-19	1.64e-24
PCA	3.94e-19	2.44e-21	1.53e-16	5.72e-19	4.18e-06	7.34e-06	5.17e-06	4.18e-06
Conv1D	6.42e-10	3.69e-09	1.07e-09	2.52e-09	7.04e-13	3.31e-11	1.00e-12	8.61e-12
LSTM-Attention	6.35e-12	1.74e-09	2.37e-12	1.73e-11	3.01e-07	5.77e-06	7.48e-08	2.04e-07
Transformer	5.76e-05	8.00e-05	9.62e-05	6.46e-05	4.76e-03	8.31e-03	4.18e-03	4.73e-03

Conduct an in-depth analysis of the capability of feature extraction methods in handling imbalanced data, with particular focus on the recognition performance for minority class samples. The Recall metric directly reflects the model's ability to identify minority class samples, while the F1-score, as the harmonic mean of precision and recall, provides a more comprehensive evaluation of the model's performance on imbalanced data.

Traditional methods such as Random Forest (RF) achieve Recall values of only 0.5077 and 0.5149 on wind turbine 15 and 21 datasets, indicating their near inability to effectively recognize minority class samples. In contrast, deep learning methods significantly enhance minority class recognition capability. Specifically, Transformer achieves Recall scores of 0.8897 and 0.9206, while Neutrosophic Transformer further improves these to 0.9236 and 0.9277. This improvement primarily stems from the hierarchical feature learning ability of deep models, which enables the extraction of more discriminative features from complex high-dimensional data.

Notably, Neutrosophic Transformer achieves F1-scores of 0.9355 and 0.9349 on the two wind turbine datasets, significantly outperforming other methods. The enhancement in F1-score indicates that the model effectively increases recall for minority class samples while maintaining high precision, achieving a more optimal adjustment of the decision boundary. This stands in sharp contrast to the tendency of traditional KNN methods to favor the majority class when handling imbalanced data, demonstrating the advantage of deep learning methods in refining classification decision boundaries through feature learning rather than simple distance metrics.

Based on the preceding analysis, this paper opts for the proposed Neutrosophic Transformer as the feature extraction method. Following feature extraction by the Neutrosophic Transformer for Fan No. 15 and Fan No. 21, the feature importance calculated by univariate feature selection methods is showed in the Figure 3 and Figure 4.

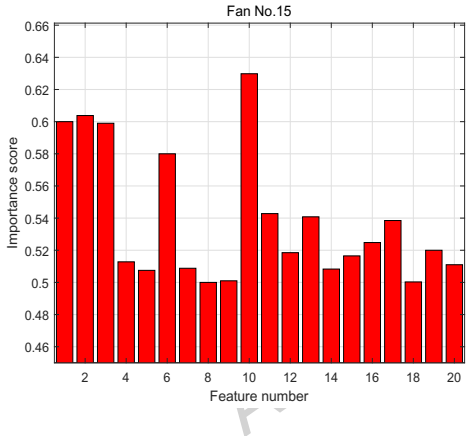

Figure 3: Importance of the attributes of fan No.15.

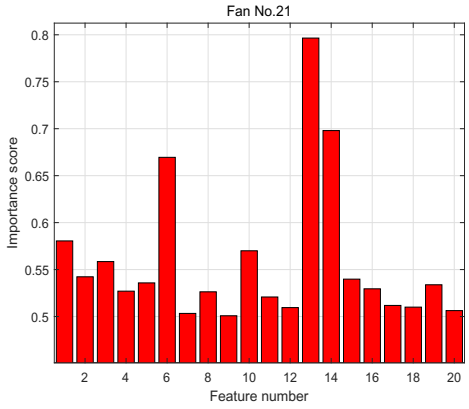

Figure 4: Importance of the attributes of fan No.21.

According to the results displayed in Figures 3 and 4, for Fan No. 15, the top eight attributes are 1, 2, 3, 6, 10, 11, 13 and 17. For Fan No. 21, the top eight attributes are 1, 2, 3, 6, 10, 13, 14 and 15.

4.1.3. Diagnostic experiments

The preprocessed data is subsequently inputted into the classification models for experimentation purposes. In this section, five conventional models—BP[47], ELM[48], SVM[49], DT[50], and KNN—alongside two outstanding deep learning methods TSMixer[51] and PatchTST[52] are employed to validate the proposed model's efficacy. The specific parameters for the experimental models are detailed in the Table 10 below:

Table 10: Values set for control parameters of compared models in diagnostic experiment

Model	Parameter	Value
BP	epochs	50
	Lr	5e-2
ELM	*LL*	20
	C	50
SVM	Kernel	rbf
	C	0.1
DT	Maxdepth	2
KNN	*k*	50
TSMixer	epochs	50
	lr	1e-5
	activation	GELU
PatchTST	epochs	50
	lr	1e-5
	P	4
KNKNN	δ	0.1
	m	2
	σ	5

The experimental results for wind turbines No. 15 and No. 21 are presented in Table 11, with diagnostic accuracy and evaluation metrics shown in Table 12.

Table 11: Results of the compared models for Fan No.15 and Fan No.21 in diagnostic experiment

Model	Fan No.15				Fan No.21			
	TP	FP	FN	TN	TP	FN	FP	TN
BP	775	646	163	5220	405	232	67	1825
ELM	829	592	221	5162	408	229	84	1808
SVM	705	716	162	5221	171	466	22	1870
DT	492	929	89	5294	534	103	430	1462
KNN	737	684	106	5277	287	350	53	1839
TSMixer	852	569	264	5119	281	356	132	1760
PatchTST	801	620	225	5158	145	492	26	1866
KNKNN	1184	237	93	5290	627	10	12	1880

Table 12: Evaluation of the compared models for Fan No.15 and Fan No.21 in diagnostic experiment

Model	Fan No.15				Fan No.21			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
BP	0.8811	0.8580	0.7576	0.7926	0.8818	0.8726	0.8002	0.8273
ELM	0.8805	0.8433	0.7712	0.7990	0.8762	0.8584	0.7981	0.8215
SVM	0.8710	0.8463	0.7330	0.7693	0.8070	0.8433	0.6284	0.6483
DT	0.8504	0.8488	0.6649	0.7019	0.7892	0.7441	0.8055	0.7565
KNN	0.8839	0.8798	0.7495	0.7907	0.8406	0.8421	0.7113	0.7444
TSMixer	0.8776	0.8317	0.7753	0.7982	0.8070	0.7561	0.6857	0.7067
PatchTST	0.8758	0.8367	0.7609	0.7895	0.7952	0.8197	0.6069	0.6185
KNKNN	0.9515	0.9421	0.9080	0.9237	0.9913	0.9880	0.9890	0.9885

Across the Wind Turbine 15 and Wind Turbine 21 diagnostic experiments in Table 11 and Table 12, KNKNN consistently outperformed all baselines on the four evaluation metrics. In Wind Turbine 15, KNKNN achieved accuracy 95.15%, precision 94.21%, recall 90.80%, and F1-score 92.37%, surpassing the strongest conventional baseline (KNN) by 6.76, 6.23, 15.85, and 13.30 percentage points, respectively. In Wind Turbine 21, KNKNN attained 99.13% accuracy, 98.80% precision, 98.90% recall, and 98.85% F1-score, exceeding the top-performing BP model by 10.95%, 11.54%, 18.88%, and 16.12%, respectively. Collectively, these results demonstrate the superior performance.

We conducted 10 repetitions of the experiments and performed paired t-tests comparing KNKNN with seven baselines shown in the Table 13. Table 13 presents the paired t-test p-values comparing KNKNN against seven benchmark methods (BP, ELM, SVM, DT, KNN, TSMixer, and PatchTST) across four evaluation metrics on two wind turbine datasets. All reported p-values are substantially below the conventional significance threshold of 0.05, with the vast majority falling below 1.0e-6 and many even below 1.0e-12. These results provide exceptionally strong statistical evidence that KNKNN consistently and significantly outperforms every competing method on both Fan No.15 and Fan No.21 datasets, across all metrics of accuracy, precision, recall, and F1-score.

The significance levels remain remarkably low regardless of the comparison model—whether traditional classifiers such as SVM, KNN, and ELM, or more recent deep learning architectures like TSMixer and PatchTST. Notably, the p-values are similarly minuscule on both datasets, demonstrating that the superiority of KNKNN is not dataset-specific but generalizes well. The extreme p-values (e.g., 1.31e-15 for SVM recall on Fan 15 and 3.31e-16 for SVM recall on Fan 21) further reinforce the overwhelming advantage of the proposed kernel neutrosophic k-nearest neighbor approach. In summary, the statistical analysis unequivocally confirms that KNKNN achieves significantly better classification performance than all compared state-of-the-art methods in this fault diagnosis task.

Table 13: Performance Comparison of KNKNN and Other Methods on the fan datasets p-value

KNKNN vs	Fan No.15				Fan No.21			
Comparative Methods	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
BP	6.84e-07	1.83e-06	1.94e-06	1.93e-08	2.34e-08	7.32e-08	1.95e-07	6.09e-08
ELM	1.02e-12	2.42e-13	9.63e-12	3.88e-12	2.13e-11	1.17e-10	1.33e-10	4.89e-11
SVM	6.09e-15	2.38e-13	1.31e-15	1.72e-15	3.62e-16	1.47e-11	3.31e-16	5.59e-15
DT	2.31e-14	7.82e-12	9.49e-15	2.77e-14	1.20e-12	7.15e-13	2.94e-10	2.64e-12
KNN	1.58e-14	1.42e-12	2.59e-15	4.21e-15	1.87e-14	1.86e-11	1.70e-14	7.24e-14
TSMixer	1.87e-13	4.48e-13	4.74e-13	1.18e-13	5.37e-12	5.91e-09	1.36e-10	5.43e-10
PatchTST	7.64e-13	1.90e-11	9.67e-11	1.48e-11	3.78e-13	3.34e-09	3.10e-12	3.62e-11

The superior performance of KNKNN over deep learning models (e.g., TSMixer, PatchTST) can be attributed to several theoretical factors. First, the Gaussian kernel in KNKNN implicitly maps the input features into a high-dimensional Reproducing Kernel Hilbert Space (RKHS), where complex nonlinear relationships become more linearly separable. This is particularly advantageous when the number of training samples is limited, as deep models with millions of

parameters tend to overfit in such scenarios. Second, the neutrosophic triplet (T, I, F) explicitly models the uncertainty of each sample. For noisy or borderline instances, the indeterminacy component I reduces their influence on the final decision, effectively suppressing misclassifications caused by outliers—a mechanism absent in standard neural classifiers. Third, KNKNN requires minimal parameter tuning (only k, σ, δ, m) and no iterative training, making it inherently robust to noise and distribution shifts. In contrast, deep learning models rely heavily on the quality of learned features; if the feature extractor fails to separate classes cleanly due to imbalance or noise, the subsequent classifier will also underperform. KNKNN, by directly operating on the feature space with a robust distance metric and uncertainty weighting, mitigates this limitation. These theoretical advantages collectively explain why KNKNN consistently outperforms more complex deep architectures in the fault diagnosis tasks considered.

In this experiment, the effectiveness of incorporating kernel functions into the NKNN model is validated through both horizontal and vertical comparative analyses. Experimental results demonstrate that projecting original data into a high-dimensional space with a Gaussian kernel function significantly enhances the proposed KNKNN's performance compared to the traditional NKNN model. The KNKNN model not only achieves superior classification accuracy but also shows greater resilience against data noise and outliers. The results clearly illustrate that kernel method greatly enhances the model's performance in handling complicated non-linear connections. Consequently, this broadens the application scenarios of the NKNN model in practical classification tasks and markedly elevates its effectiveness.

4.2. Case 2

4.2.1. Dataset description

The data utilised in this experiment originates from the bearing dataset developed by Case Western Reserve University[53], which has seen extensive application within the field of bearing fault diagnosis. The modality of this dataset is time-domain vibration signals. The noise in this dataset primarily consists of electrical noise, mechanical noise, and background noise. The experiment employed drive-end vibration data with a sampling frequency of 12 kHz, encompassing four operational conditions: normal operation, inner race fault, outer race fault, and ball element fault.

For data preprocessing, a sliding window approach is implemented with a window size of 8 data points and a step size of 32 samples, generating a total of 19,026 samples across all conditions. The dataset exhibits a balanced distribution with 7,616 normal samples and 11,410 fault samples, ensuring robust model training and evaluation. The dataset exhibits an imbalance rate of 1:1.5.

4.2.2. Feature extraction

The experimental hardware conditions, comparison models, and model parameters are the same as those in Section 4.1. Four distinct feature extraction methodologies are investigated in this comparative study: the proposed Neutrosophic Transformer, a conventional Transformer architecture, Random Forest-based feature selection, and Principal Component Analysis for dimensionality reduction. All extracted features are normalized to the [0,1] range using Min-Max scaling prior to classification.

For the CWRU bearing dataset (Case 2), we apply a sliding window of length 8 to generate samples that contain consecutive time steps. Hence, deep learning models such as Conv1D, LSTM-Attention, and Transformer can exploit local temporal patterns within each window. In contrast, traditional methods (RF, PCA) treat the flattened 8-dimensional feature vector as an independent sample, thus ignoring the temporal structure. The results are documented in Tables 14 and 15.

Table 14: Results of the compared models for CWRU in feature extraction experiment

Feature extraction methods	TP	FP	FN	TN
RF	3808	231	4	5485
PCA	3806	113	6	5603
Conv1D	3787	74	25	5642
LSTM-Attention	3812	212	0	5504
Transformer	3812	47	0	5669
Neutrosophic Transformer	3812	22	0	5694

Table 15: Results of the compared models for CWRU in feature extraction experiment

Feature extraction methods	Accuracy	Precision	Recall	F1-score
RF	0.9753	0.9710	0.9793	0.9745
PCA	0.9875	0.9850	0.9893	0.9870
Conv1D	0.9896	0.9882	0.9902	0.9892
LSTM-Attention	0.9777	0.9737	0.9815	0.9770
Transformer	0.9951	0.9939	0.9959	0.9949
Neutrosophic Transformer	0.9977	0.9971	0.9981	0.9976

According to the results in Table 14 and the evaluation metrics in Table 15, the Neutrosophic Transformer outperforms the standard Transformer by 0.26% in accuracy, 0.32% in precision, 0.22% in recall, and 0.27% in F1 score, surpassing it across all metrics. These results underscore the value of integrating neutrosophic principles into the Transformer architecture. Moreover, the Neutrosophic Transformer significantly outperforms comparative methods across all metrics. Its four evaluation metrics exceed the best-performing traditional Conv1D method by 0.81%, 0.89%, 0.79%, and 0.84% respectively.

Through 10 repeated experiments, the Neutrosophic Transformer demonstrated statistically significant advantages over other methods on the CWRU bearing fault dataset. As shown in the Table 16, the Neutrosophic Transformer achieved highly significant (p ¡ 0.001) improvements across these comparisons: accuracy, precision, recall, and F1 score. When compared with traditional methods (Random Forest, PCA), p-values are consistently below 1e-7, with the lowest p-value reaching 2.15e-10 (PCA for precision). When contrasted with deep learning benchmarks models (Original Transformer, Conv1D, LSTM-Attention), the Neutrosophic Transformer also shows significant advantages, with p-values ranging from 1.08e-03 (Original Transformer) to 2.39e-07 (LSTM-Attention). These results confirm that the Neutrosophic Transformer's performance improvements in feature extraction and classification on the CWRU dataset are statistically reliable and not due to random variations.

Table 16: Performance Comparison of Neutrosophic Transformer and Other Methods on the Fan Dataset p-value

Neutrosophic Transformer	Fan No.15			
vs Comparative Methods	Accuracy	Precision	Recall	F1-score
RF	3.21e-08	1.47e-08	6.69e-08	3.03e-08
PCA	3.84e-10	2.15e-10	4.98e-10	3.60e-10
Conv1D	6.49e-06	7.06e-06	5.48e-06	6.36e-06
LSTM-Attention	2.38e-07	1.41e-07	2.39e-07	2.20e-07
Transformer	1.08e-03	1.01e-03	1.08e-03	1.07e-03

In the CWRU bearing fault diagnosis task, although the degree of data imbalance is relatively smaller compared to the wind turbine dataset, the recognition capabilities of different feature extraction methods for minority classes (fault samples) still exhibit significant differences. The Recall metric shows that all methods maintain a high fault recognition rate, reflecting the relatively balanced nature of this dataset. However, the performance variations among different methods still reveal the impact of feature extraction mechanisms on classification performance.

Deep learning-based feature extraction methods demonstrate clear advantages on the CWRU dataset, particularly with Neutrosophic Transformer achieving the highest Recall (0.9982) and F1-score (0.9977). Compared with the traditional Random Forest, which yields Recall 0.9793 and F1-score 0.9745, the Neutrosophic Transformer provides improvements that are statistically and practically meaningful, even though the absolute gains are modest given the high baseline performance. Conv1D and Transformer also achieve Recall scores of 0.9902 and 0.9959, respectively, further validating the effectiveness of deep feature extraction.

From the comparison between deep learning methods and traditional methods, neural network-based approaches such as Conv1D, LSTM-Attention, and Transformer outperform RF and PCA across all metrics. This demonstrates that deep learning models, through multi-level nonlinear transformations, can learn feature representations more suit-

able for classification tasks. Specifically, Neutrosophic Transformer achieves further improvement upon the existing Transformer architecture by incorporating neutrosophic uncertainty modeling. This validates the importance of explicitly modeling uncertainty in feature extraction when dealing with noisy and uncertain industrial data.

Overall, the Neutrosophic Transformer demonstrates distinct advantages as a feature extraction method and can be considered an effective feature extraction approach. Following extraction via the Neutrosophic Transformer, 7,623 normal samples and 11,433 frozen samples are generated.

4.2.3. Diagnosis evaluation

The preprocessed data are subsequently input into the classification models for experimentation. The specific parameters of the models compared in this section correspond to those in Table 9 of Section 6.1. A total of 10 experiments are conducted. The experimental results and evaluation metrics are presented in Table 17 and Table 18.

Table 17: Results of the compared models for CWRU in diagnostic experiment

Model	TP	FP	FN	TN
BP	3549	263	405	5311
ELM	2128	1684	1010	4706
SVM	3576	236	536	5180
DT	3022	790	935	4781
KNN	3683	129	528	5188
TSMixer	3645	167	498	5218
PatchTST	970	2842	297	5419
KNKNN	3742	70	220	5496

Table 18: Evaluation of the compared models for CWRU in diagnostic experiment

Model	Accuracy	Precision	Recall	F1-score
BP	0.9299	0.9252	0.9301	0.9274
ELM	0.7173	0.7073	0.6908	0.6949
SVM	0.9190	0.9130	0.9222	0.9166
DT	0.8190	0.8110	0.8146	0.8126
KNN	0.9310	0.9252	0.9369	0.9293
TSMixer	0.9302	0.9244	0.9345	0.9283
PatchTST	0.6705	0.7108	0.6013	0.5787
KNKNN	0.9696	0.9659	0.9716	0.9685

Table 17 and 18 show that, across the CWRU bearing fault dataset, KNKNN outperformed the five traditional models on all evaluated metrics. Specifically, KNKNN achieved 96.96% accuracy, 96.59% precision, 97.16% recall, and 96.85% F1-score respectively. Compared to KNN—the best among the compared methods—these represent relative improvements of 3.86%, 4.07%, 3.47%, and 3.92%. Despite occasional underperformance in certain metrics when compared to other traditional models, KNKNN generally exhibits strong capabilities and demonstrates high applicability.

Table 19 reports the paired t-test p-values comparing KNKNN against seven baseline models on the CWRU bearing dataset. The statistical tests yield p-values far below 0.05 for all comparisons, underscoring the reliability of the observed improvements. Particularly, the p-values against BP range from 0.0137 to 0.0315, indicating a significant but relatively moderate advantage over the multilayer perceptron. In contrast, the p-values for all other competing methods (ELM, SVM, DT, KNN, TSMixer, PatchTST) are extremely small, with most falling below 1.0e-7 and many even below 1.0e-12. This demonstrates that KNKNN overwhelmingly outperforms these models with exceptionally high statistical confidence.

The results on the CWRU dataset are highly consistent with those observed on the two fan datasets, further validating the robustness and generalization capability of the proposed KNKNN approach. Despite the different characteristics of the CWRU bearing fault data, KNKNN maintains its decisive superiority over a diverse set of traditional machine learning algorithms and modern deep learning architectures. The consistently minimal p-values provide compelling statistical evidence that KNKNN is a highly effective and reliable classifier for fault diagnosis tasks across different industrial systems.

Table 19: Performance Comparison of KNKNN and Other Methods on the CWRU dataset p-value

KNKNN vs	CWRU			
Comparative Methods	Accuracy	Precision	Recall	F1-score
BP	1.37e-02	3.15e-02	1.99e-02	2.69e-02
ELM	1.12e-08	1.03e-08	3.43e-08	5.03e-08
SVM	2.18e-13	2.81e-13	6.33e-14	1.86e-13
DT	2.73e-11	6.11e-16	6.68e-17	5.31e-12
KNN	1.56e-12	1.49e-12	1.23e-12	1.53e-12
TSMixer	2.46e-07	1.83e-07	1.25e-07	2.13e-07
PatchTST	4.79e-09	3.66e-10	4.22e-08	4.65e-07

4.3. Ablation Study

To identify which module in the Neutrosophic Transformer contributes the most to performance gains and to evaluate the effectiveness of incorporating kernel functions, the ablation experiments are conducted on both the Neutrosophic Transformer and KNKNN. The following models are used as feature extractors including the baseline Transformer, the Neutrosophic Transformer with only the dynamic threshold mechanism (DT), the Neutrosophic Transformer with only the attention residual refinement mechanism (ARR), and the complete Neutrosophic Transformer. Standard classifiers including MLP, NKNN, and KNKNN are employed as classifiers. This configuration yields 12 experimental scenarios. We employ an MLP with a single hidden layer of 10 neurons and ReLU activation as a representative neural network classifier. The MLP is set to 20 iterations with a learning rate of 1e-2. All other parameter settings followed Sections 4.1 and 4.2. Experiments are conducted on the CWRU dataset, with results reported in Table 20.

Table 20: Evaluation of the compared models for CWRU in ablation experiment

Feature extractor	Classifier	Accuracy	Precision	Recall	F1-score
Transformer	MLP	0.9287	0.9243	0.9406	0.9275
	NKNN	0.9484	0.9428	0.9570	0.9472
	KNKNN	0.9488	0.9433	0.9573	0.9476
Neutrosophic Transformer with DT	MLP	0.8909	0.8928	0.9091	0.8900
	NKNN	0.9543	0.9487	0.9617	0.9531
	KNKNN	0.9551	0.9495	0.9624	0.9540
Neutrosophic Transformer with ARR	MLP	0.5157	0.7262	0.5965	0.4732
	NKNN	0.9501	0.9445	0.9582	0.9489
	KNKNN	0.9522	0.9466	0.9600	0.9510
Neutrosophic Transformer	MLP	0.9513	0.9457	0.9591	0.9502
	NKNN	0.9551	0.9495	0.9624	0.9540
	KNKNN	0.9555	0.9500	0.9628	0.9544

The standard Transformer is employed as the baseline feature extractor, producing F1-scores of approximately 0.947 when paired with NKNN and KNKNN, and about 0.928 with the MLP. When the DT is applied in isolation,

the MLP's F1-score fell to 0.890, while NKNN and KNKNN rose to 0.953 and 0.954, respectively, with recall exceeding 0.96. This pattern suggests that DT enhances sensitivity to minority class samples by shifting the decision boundary, yet the resulting feature space can pose greater learning challenges for a shallow neural network. By contrast, distance-based methods such as NKNN and KNKNN remain robust to these distortions. With the ARR applied alone, the MLP's F1-score drops sharply to 0.473, accompanied by a recall of only 0.596, nearly a failure, whereas NKNN and KNKNN sustain F1-scores around 0.95. It implies that ARR strengthens fine-grained nonlinear feature representations, but the modified feature space, although advantageous for distance-based classifiers, is less suitable for the MLP because of its limited capacity and potential for overfitting. KNN-type methods, which rely on local distances, are comparatively less affected. When both DT and ARR are integrated (the full Neutrosophic Transformer), the best or near-best performance is observed across classifiers. Specifically, the MLP's F1-score recovers to 0.950, while NKNN and KNKNN reaches 0.954 and 0.9544, respectively, with recall remaining above 0.96. Therefore, DT and ARR are complementary, collectively preserving a well-structured feature space that accommodates different classifier types.

A focused comparison between NKNN and KNKNN indicates that KNKNN consistently outperforms NKNN across all configurations (e.g., on Neutrosophic Transformer features, F1-score increases from 0.9540 to 0.9544), which confirms that the kernel function further exploits nonlinear relationships and provides a marginal performance gain. In summary, the DT is key to improving minority-class recall, while the ARR enhances fine-grained feature representation. Their synergy enables the full Neutrosophic Transformer to achieve optimal results across classifiers, with the kernel offering additional improvement over NKNN.

Table 21: Evaluation of the compared models for CWRU in ablation experiment under noise level 0.3

Feature extractor	Classifier	Accuracy	Precision	Recall	F1-score
Transformer	MLP	0.5329	0.7156	0.6091	0.4995
	NKNN	0.7390	0.7897	0.7781	0.7386
	KNKNN	0.7470	0.7907	0.7836	0.7468
Neutrosophic Transformer with DT	MLP	0.7407	0.7887	0.7788	0.7403
	NKNN	0.7440	0.7873	0.7804	0.7439
	KNKNN	0.7595	0.7589	0.7697	0.7570
Neutrosophic Transformer with ARR	MLP	0.7193	0.7559	0.7524	0.7192
	NKNN	0.7415	0.7887	0.7794	0.7412
	KNKNN	0.7491	0.7914	0.7851	0.7489
Neutrosophic Transformer	MLP	0.7365	0.7855	0.7750	0.7361
	NKNN	0.7482	0.7884	0.7834	0.7481
	KNKNN	0.7612	0.7760	0.7839	0.7608

An ablation study is conducted on the CWRU dataset with a noise level of 0.3 to evaluate the contribution of each module under noisy conditions. As shown in Table 21, the features extracted by the baseline Transformer suffer severe performance degradation in noisy environments, with the MLP classifier achieving an accuracy of only 0.5329 and an F1-score as low as 0.4995, which implies that the original Transformer is sensitive to noise. The introduction of the DT significantly improves the feature quality of the MLP, raising accuracy to 0.7407 and F1 to 0.7403. It demonstrates that DT effectively mitigates the impact of noise by adaptively adjusting the decision boundary. When the ARR is applied alone, the MLP F1-score rose to 0.7192, still below the DT-only result, suggesting that ARR enhances robustness by preserving fine-grained features, while DT more directly addresses boundary uncertainty. The full Neutrosophic Transformer, which combines DT and ARR, achieves an MLP F1-score of 0.7361, comparable to DT alone.It indicates that the synergy between the two may involve a slight trade-off in noisy conditions, yet still significantly outperforms the baseline.

NKNN and KNKNN consistently outperform MLP across all feature extractors, highlighting the stability of distance-based methods on noisy data. Especially, KNKNN outperforms NKNN in all configurations, validating the ability of the kernel function to model nonlinear noise patterns. On features extracted by the full Neutrosophic

Transformer, KNKNN achieves the highest F1-score of 0.7608, with precision slightly lower than NKNN but recall improved, resulting in a better overall balance. These results confirm that the combination of neutrosophic representation and kernel methods offers the greatest robustness under strong noise. Compared to the noise-free ablation experiment, the relative contributions of DT and ARR are more pronounced under noisy conditions: DT plays a crucial role in protecting linear classifiers such as MLP, while ARR better preserves feature structures beneficial for KNN-based methods. Each module is proved to be essential for ensuring the noise robustness of the proposed framework.

4.4. Complexity Analysis

4.4.1. Time Complexity

Table 22: Time used in feature extraction and diagnostic experiments

Method	Fan No.15			Fan No.21			CWRU		
	Training	Inference	Total	Training	Inference	Total	Training	Inference	Total
RF	0.690	0.004	0.694	0.230	0.004	0.234	1.000	0.006	1.006
PCA	0.001	0.001	0.002	0.001	0.001	0.002	0.002	0.001	0.003
Conv1D	26.264	0.056	26.319	9.435	0.018	9.453	31.193	0.033	31.226
LSTM-Attention	34.943	0.028	34.971	13.147	0.010	13.156	43.565	0.032	43.598
Transformer	59.944	0.015	59.958	22.751	0.006	22.758	49.257	0.011	49.268
Neutrosophic Transformer	99.004	0.036	99.040	37.163	0.014	37.177	138.448	0.0543	138.503
BP	0.421	0.008	0.429	0.178	0.000	0.178	0.442	0.003	0.444
ELM	0.011	0.002	0.013	0.002	0.000	0.002	0.007	0.005	0.012
SVM	0.492	1.034	1.526	0.075	0.146	0.222	0.059	0.094	0.153
DT	0.011	0.000	0.011	0.004	0.000	0.004	0.123	0.001	0.124
KNN	0.010	0.167	0.177	0.004	0.035	0.039	0.001	3.646	3.647
TSMixer	53.216	0.015	53.231	19.200	0.009	19.209	70.890	0.013	70.903
PatchTST	72.397	0.044	72.441	26.678	0.014	26.692	232.459	0.497	232.956
KNKNN	175.444	0.000	175.444	22.448	0.000	22.448	317.623	0.000	317.623

Based on the running time data given in Table 22 and the performance metrics in Section 4.1 and 4.2, an empirical analysis of the computational complexity of the proposed method is conducted. The additional modules are introduced in Neutrosophic Transformer for neutrosophic feature extraction, dynamic thresholding, and residual correction on top of the standard Transformer (with a theoretical complexity of $O(Ld^2_{\text{model}})$), leading to a larger constant factor. Consequently, the overall training time is approximately 1.6 to 2.8 times that of the original Transformer (i.e. 99.0s vs. 59.9s on Fan No.15, and 138.4s vs. 49.3s on CWRU). However, its inference time remains below 0.05 seconds, meeting the real-time requirements of industrial applications. Simultaneously, the F1-score in feature extraction tasks is improved by 2.3 to 22.2 percentage points compared to the next best method, indicating a substantial enhancement in feature quality.

In the diagnostic phase, the KNKNN model requires kernel-distance computations for all training samples, which increases the complexity to $O(N^2d)$. As a result, the training time far exceeds that of traditional KNN (e.g., 175.4s vs. 0.01s on Fan No.15), while inference time is effectively zero since all computations are carried out during training. In terms of performance, the F1-score of KNKNN surpasses that of KNN by 16.8% on Fan No.15 and 3.9% on CWRU. It also substantially outperforms other deep learning models such as TSMixer and PatchTST (i.e. PatchTST achieves an F1-score of only 0.58 on CWRU). Although the proposed method incurs a significant increase in training overhead, the advantages in minority-class recognition and noise robustness greatly exceed those of conventional methods, and its inference latency remains negligible. This favorable trade-off supports the applications in industrial where timely and robust fault detection is essential.

4.4.2. Space Complexity

Table 23: Theoretical space complexity of compared models

Model	**Theoretical Space Complexity**	**Estimated Parameters**
RF	$O(n_{\text{estimators}} \cdot \text{max\_depth} \cdot d)$	Tens of KB
PCA	$O(n_{\text{components}} \cdot d)$	Hundreds of bytes
Conv1D	$O(C^2 \cdot K + C \cdot d)$	~10K
LSTM-Attention	$O(H^2 + H \cdot d)$	~20K–30K
Transformer (original)	$O(Ld^2_{\text{model}})$	~10K–50K
Neutrosophic Transformer	$O(Ld^2_{\text{model}} + \text{extra heads})$	~200K
BP	$O(\text{width}^2)$	~200
ELM	$O(n_{\text{hidden}} \cdot d + n_{\text{hidden}})$	~400
SVM	$O(n_{\text{SV}} \cdot d)$	Data-dependent
DT	$O(\text{nodes})$	Small
KNN	$O(N \cdot d)$	Stores training set
TSMixer	$O(\text{layers} \cdot H^2)$	~34K
PatchTST	$O(\text{layers} \cdot H^2 \cdot n_{\text{patches}})$	~100K
KNKNN	$O(N \cdot d) + O(N)$	Same as KNN + $3N$

An examination of space complexity in Table 23 shows that, the proposed Neutrosophic Transformer inherits the basic structure of the standard Transformer, whose parameter count scales as $O(Ld^2_{\text{model}})$. With the chosen configuration ($d_{\text{model}} = 64$, $L = 3$), the total parameter count is about 200K, which is still far below the millions of parameters typical of modern deep learning models and easily accommodated by standard GPU memory. The additional modules (neutrosophic head, residual correction) introduce only a few thousand extra parameters, incurring negligible memory overhead. For KNKNN, the space requirement is essentially the same as that of conventional KNN. It primarily stores the entire training set ($O(Nd)$), plus an extra three floating-point values per training sample for the neutrosophic triplets. Hence, the additional memory cost is marginal.

Compared with other deep models, such as TSMixer (approximately 34K parameters) and PatchTST (roughly 100K parameters), the Neutrosophic Transformer exhibits moderately higher parameter counts, yet it remains within a completely acceptable range for industrial deployment. Moreover, the inference stage for all proposed models consumes negligible additional memory since large intermediate structures are not retained. Consequently, the space overhead introduced by the novel components is not a practical constraint. The substantial gains in classification accuracy and robustness are achieved primarily at the cost of increased training time, not memory. These benefits are achieved at the cost of longer training times, while maintaining real-time inference capability.

4.5. Noise Injection Experiment

Table 24: Performance of the comparative models under different noise conditions

Method	Accuracy					F1-score				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
RF	0.8734	0.7988	0.7294	0.6834	0.6429	0.8699	0.7929	0.7203	0.6725	0.6273
PCA	0.9530	0.8899	0.8150	0.7588	0.7056	0.9515	0.8868	0.8092	0.7518	0.6955
Conv1D	0.9614	0.8877	0.8139	0.7415	0.7100	0.9599	0.8836	0.8067	0.7304	0.6965
LSTM-Attention	0.9508	0.8849	0.8129	0.7419	0.6999	0.9495	0.8825	0.8087	0.7357	0.6916
Transformer	0.9436	0.8744	0.8035	0.7175	0.6924	0.9416	0.8703	0.7969	0.7092	0.6807
Neutrosophic Transformer	0.9680	0.9009	0.8263	0.7562	0.7071	0.9668	0.8976	0.8208	0.7484	0.6981
BP	0.9169	0.8878	0.8847	0.8717	0.8577	0.9137	0.8820	0.8794	0.8692	0.8540
ELM	0.7120	0.7174	0.6598	0.6416	0.6368	0.6932	0.6909	0.6206	0.5791	0.5741
SVM	0.9165	0.9145	0.9116	0.9097	0.9032	0.9141	0.9119	0.9090	0.9070	0.9002
DT	0.7666	0.7207	0.7001	0.6436	0.6237	0.7666	0.7201	0.6996	0.6383	0.6185
KNN	0.9299	0.9272	0.9200	0.9136	0.9049	0.9281	0.9254	0.9181	0.9117	0.9030
TSMixer	0.9310	0.9004	0.9148	0.8918	0.8808	0.9289	0.8986	0.9127	0.8899	0.8781
PatchTST	0.7067	0.7024	0.6855	0.6747	0.6241	0.6391	0.6358	0.6314	0.5964	0.4632
KNKNN	0.9569	0.9538	0.9444	0.9289	0.9141	0.9555	0.9524	0.9427	0.9269	0.9119

To assess robustness to noise, we conduct noise injection experiments on the CWRU bearing dataset. Specifically, Gaussian white noise is added to the original data, with the noise intensity defined as a multiple of the standard deviation of each feature, taking values of 0.1, 0.2, 0.3, 0.4, and 0.5. All models are trained and tested under the same noise levels, using a 1:1 random split that preserved the original class proportions. The aim is to simulate real-world scenarios where sensor data in engineering environments is contaminated by noise, and to examine the performance degradation of different feature extraction methods and classifiers under noise interference.

Across the overall trend in Table 24, as the noise level increases, the accuracy and F1-score of all models are degraded at different rates. Traditional machine learning models such as SVM, KNN, and BP demonstrate excellent noise robustness, maintaining accuracy above 0.90 and F1-scores close to 0.90 even at a high noise level of 0.5. This phenomenon indicates that these models are insensitive to the scale of features and can still effectively capture discriminative information in the data under high-noise environments. In contrast, deep learning models such as Conv1D, LSTM-Attention, and Transformer perform excellently at low noise (0.1) with accuracies around 0.95, but their performance deteriorates rapidly as noise increases, falling below 0.71 at 0.5, revealing their vulnerability to noise. This may be because the complex parameter structures of deep models are prone to overfitting noise, while the relatively simpler decision boundaries of traditional models offer better generalization ability.

Among the feature extraction methods, the Neutrosophic Transformer consistently outperforms other feature extractors in the noise range of 0.1 to 0.3, with its accuracy gradually decreasing from 0.968 to 0.826, significantly higher than the ordinary Transformer and Conv1D. The neutrosophic representation (T, I, F) and the dynamic threshold mechanism help suppress noise impact while preserving discriminative information. However, when the noise level exceeds 0.4, the advantage of Neutrosophic Transformer gradually narrows, becoming comparable to models including Conv1D, indicating that any feature extractor struggles to completely separate signal from noise under extremely strong noise, and performance is limited by the signal-to-noise ratio of the data itself.

Particularly, as a variant of KNN that combines kernel methods with neutrosophic theory, KNKNN demonstrates the strongest robustness across all noise levels. At a noise level of 0.5, its accuracy remains 0.9141 with an F1-score of 0.9119, significantly outperforming other deep classifiers (e.g., TSMixer, PatchTST) and traditional models like SVM and KNN. This result indicates that the kernelized distance metric, combined with neutrosophic modeling of uncertainty, mitigates the distortion of neighbor relations caused by noise, sustaining stable discriminative performance in highly contaminated data. By contrast, PatchTST performs the worst under noise, achieving only 0.6241 accuracy and 0.4632 F1 at 0.5, illustrating its reliance on local temporal structure which noise disrupts.

The noise injection experiments systematically validate the behavioral differences of each model under noise interference. The proposed Neutrosophic Transformer demonstrates strong noise resistance during the feature extraction stage, while the KNKNN classifier further elevates this robustness to a leading level, maintaining an accuracy above 0.91 even under high noise of 0.5, which supports the applicability and superiority of the proposed framework in real-world industrial noisy environments.

5. Conclusion

This paper addresses key challenges in data-driven monitoring and classification, such as severe class imbalance and noise interference, by proposing a novel hybrid model that integrates the Neutrosophic Transformer with the KNKNN algorithm. The Neutrosophic Transformer incorporates a dynamic threshold mechanism and an attention residual refinement mechanism. Through a ternary representation of true, false, and uncertain values, it effectively mitigates the impact of noise during feature extraction while maintaining stable performance under imbalanced data distributions. By integrating kernel functions, the KNKNN significantly enhances the model's capacity to characterize complex nonlinear patterns, thereby achieving high-precision classification across diverse operational scenarios.

From a computational perspective, our model maintains practical effieciency, i.e. the Neutrosophic Transformer retains the $O(Ld^2_{\text{model}})$ complexity of standard Transformers, while the KNKNN classifier adds only linear overhead relative to sample size and feature dimension. This balance between performance and efficiency makes our approach suitable for real-world industrial systems where both accuracy and computational constraints must be considered. The experimental results on real-world engineering datasets, including those from wind turbine and rotational machinery monitoring, demonstrate that our model achieves best accuracy and lowest variance even under strong noise and severe class imbalance. The robustness and generalisation capability are validated in complex environments. The model exhibits particular advantages in noise suppression and uncertainty quantification, effectively distinguishing discriminative patterns from interference to enhance classification reliability.

This work provides a robust and adaptive framework for classification under realistic data conditions. By leveraging the uncertainty modelling capabilities of neutrosophic theory, it effectively addresses the limitations of conventional methods when dealing with noisy and imbalanced data, contributing to enhanced analytical reliability in data-driven system monitoring.

However, this work still has certain limitations, such as the lack of direct association with posterior confidence, a reliance on grid search methods for hyperparameter optimization (which wastes some computational resources), and its current application being confined to the field of rotating machinery. Future research will continue to explore how to integrate neutrosophic parameters with model uncertainty (e.g., confidence, entropy) through approaches such as learning uncertainty mappings or Bayesian methods, thereby establishing a more rigorous theoretical foundation. Additionally, we aim to equip the model with capabilities for adaptive parameter tuning, cross-domain validation, and dynamic environment adaptation.

References

[1] Cao, Heling, et al. "Multiple fault localization based on ant colony algorithm via genetic operation." *Journal of King Saud University-Computer and Information Sciences* 35.8 (2023): 101668.

[2] Bhandari, Guru Prasad. "Dependency-based fault diagnosis approach for SOA-based systems using Colored Petri Nets." *Journal of King Saud University-Computer and Information Sciences* 34.2 (2022): 480-491.

[3] Cortina, Gerard, et al. "Mean kinetic energy distribution in finite-size wind farms: A function of turbines' arrangement." *Renewable energy* 148 (2020): 585-599.

[4] Jeelani, Zubair, and Fasel Qadir. "Cellular automata-based approach for salt-and-pepper noise filtration." *Journal of King Saud University-Computer and Information Sciences* 34.2 (2022): 365-374.

[5] Benfradj, Awatef, et al. "Integration of artificial intelligence (AI) with sensor networks: Trends, challenges, and future directions." *Journal of King Saud University-Computer and Information Sciences* 36.1 (2024): 101892.

[6] Tao, Liangliang, et al. "NCLWO: Newton's cooling law-based weighted oversampling algorithm for imbalanced datasets with feature noise." Neurocomputing 610 (2024): 128538.

[7] Tao, Liangliang, et al. "Newton cooling theorem-based local overlapping regions cleaning and oversampling techniques for imbalanced datasets." Neurocomputing 616 (2025): 128959.

[8] Tao, Liangliang. "Local entropy-adversarial oversampling for imbalanced datasets." Neurocomputing (2025): 131959.

[9] Tao, Liangliang, Qingya Wang, and Faqiang Wang. "Semi-Supervised Local Entropy-Decayed Oversampling for Imbalanced Data." Knowledge-Based Systems (2025): 115009.

[10] Vaswani, Ashish, et al. ”Attention is all you need.” *Advances in neural information processing systems* 30 (2017).

[11] Ding, Yifei, et al. ”A novel time–frequency Transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings.” *Mechanical Systems and Signal Processing* 168 (2022): 108616.

[12] Lu, Zhiqiang, et al. ”Rotating machinery fault diagnosis under multiple working conditions via a time-series transformer enhanced by convolutional neural network.” *IEEE Transactions on Instrumentation and Measurement* 72 (2023): 1-11.

[13] Xu, Zhiqiang, et al. ”DTST: A dual-aspect time series transformer model for fault diagnosis of space power system.” *IEEE Transactions on Instrumentation and Measurement* 73 (2024): 1-10.

[14] Xue, Kai, et al. ”Joint dispatch and economic collaboration of multiple regional energy systems via Transformer-based load prediction and two-stage stochastic optimization.” *Energy* (2025): 137321.

[15] Kellil, N., et al. ”Hybrid Vision Transformer Model for Defect Classification in Photovoltaic Modules Using Thermographic Imaging: Leveraging Self-Attention Mechanisms for Enhanced Accuracy.” *Renewable Energy* (2025): 124138.

[16] Sert, Eser, and Soner Kiziloluk. ”A hybrid deep learning approach combining neutrosophic set theory and multi-axis vision transformer for nutrient deficiency classification in plants.” Engineering Applications of Artificial Intelligence 162 (2025): 112705.

[17] Özyurt, Fatih, et al. ”Brain tumor detection based on Convolutional Neural Network with neutrosophic expert maximum fuzzy sure entropy.” Measurement 147 (2019): 106830.

[18] Ramachandran, Akshat, et al. ”Ntrans-net: a multi-scale neutrosophic-uncertainty guided transformer network for indoor depth completion.” 2023 IEEE International Conference on Image Processing (ICIP). IEEE, 2023.

[19] Liu, Ruonan, et al. ”Artificial intelligence for fault diagnosis of rotating machinery: A review.” *Mechanical Systems and Signal Processing* 108 (2018): 33-47.

[20] Suyanto, Suyanto, et al. ”A multi-voter multi-commission nearest neighbor classifier.” *Journal of King Saud University-Computer and Information Sciences* 34.8 (2022): 6292-6302.

[21] Ali, Mohammad Zawad, et al. ”Machine learning-based fault diagnosis for single-and multi-faults in induction motors using measured stator currents and vibration signals.” *IEEE Transactions on Industry Applications* 55.3 (2019): 2378-2391.

[22] Cai, Yuang, et al. ”Rotating rectifier fault diagnosis of nuclear multiphase brushless excitation system based on DTW metric and kNN classifier.” *IEEE Transactions on Power Electronics* 38.8 (2023): 10329-10343.

[23] Drir, Nadia, et al. ”Hybrid CNN-EML model for fault diagnosis in Electroluminescence images of photovoltaic cells.” *Renewable Energy* (2025): 123343.

[24] Malik, Mehak Mushtaq, et al. ”A novel deep CNN model with entropy coded sine cosine for corn disease classification.” *Journal of King Saud University-Computer and Information Sciences* 36.7 (2024): 102126.

[25] Boongoen, Tossapon, Natthakan Iam-On, and James Mullaney. ”Providing contexts for classification of transients in a wide-area sky survey: An application of noise-induced cluster ensemble.” *Journal of King Saud University-Computer and Information Sciences* 34.8 (2022): 5007-5019.

[26] Bhagyalakshmi, Vishwanath, Ramchandra Vittal Pujeri, and Geetha Dundesh Devanagavi. ”GB-SVNN: Genetic BAT assisted support vector neural network for arrhythmia classification using ECG signals.” *Journal of King Saud University-Computer and Information Sciences* 33.1 (2021): 54-67.

[27] Keller, James M., Michael R. Gray, and James A. Givens. ”A fuzzy k-nearest neighbor algorithm.” *IEEE transactions on systems, man, and cybernetics* 4 (1985): 580-585.

[28] Zhu, Hong, Xizhao Wang, and Ran Wang. ”Fuzzy monotonic K-nearest neighbor versus monotonic fuzzy K-nearest neighbor.” *IEEE Transactions on Fuzzy Systems* 30.9 (2021): 3501-3513.

[29] Zhu, Hongwei, and Otman Basir. ”An adaptive fuzzy evidential nearest neighbor formulation for classifying remote sensing images.” *IEEE Transactions on Geoscience and Remote Sensing* 43.8 (2005): 1874-1889.

[30] Li, Wei, Yumin Chen, and Yuping Song. ”Boosted K-nearest neighbor classifiers based on fuzzy granules.” *Knowledge-Based Systems* 195 (2020): 105606.

[31] Banerjee, Imon, Sankha Subhra Mullick, and Swagatam Das. ”On Convergence of the Class Membership Estimator in Fuzzy *k*-Nearest Neighbor Classifier.” *IEEE Transactions on Fuzzy Systems* 27.6 (2018): 1226-1236.

[32] Akbulut, Yaman, et al. ”NS-k-NN: Neutrosophic set-based k-nearest neighbors classifier.” *Symmetry* 9.9 (2017): 179.

[33] Ahmed, Reem, Fuzhan Nasiri, and Tarek Zayed. ”A novel Neutrosophic-based machine learning approach for maintenance prioritization in healthcare facilities.” *Journal of building engineering* 42 (2021): 102480.

[34] Mellit, Adel, and Soteris Kalogirou. ”Recent advances in the application of infrared thermographic imaging and embedded artificial intelligence for fault diagnosis and predictive maintenance of photovoltaic plants: Challenges and future directions.” *Renewable and Sustainable Energy Reviews* 223 (2025): 116057.

[35] Liu, Keying, et al. ”Adaptive frequency attention-based interpretable Transformer network for few-shot fault diagnosis of rolling bearings.” *Reliability Engineering* & System Safety (2025): 111271.

[36] Dong, Yutong, et al. ”An interpretable integration fusion time-frequency prototype contrastive learning for machine fault diagnosis with limited labeled samples.” *Information Fusion* 124 (2025): 103340.

[37] Hofmann, Thomas, Bernhard Schölkopf, and Alexander J. Smola. *”Kernel methods in machine learning.”* (2008): 1171-1220.

[38] Zadeh, Lotfi A. ”Fuzzy sets.” *Information and control* 8.3 (1965): 338-353.

[39] Attanassov, Krassimir T. ”Intuitionistic fuzzy sets.” *Fuzzy sets and systems* 20.1 (1986): 87-96.

[40] Smarandache, Florentin. ”Neutrosophy: neutrosophic probability, set, and logic: analytic synthesis & *synthetic analysis*.” (1998).

[41] Dehghani, Mohammad, et al. ”Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems.” *Knowledge-based systems* 259 (2023): 110011.

[42] China Academy of Information and Communications Technology. Industrial Big Data Innovation Platform SCADA Dataset. V1.0, Zenodo, 2024.08.09, https://doi.org/10.5281/zenodo.13284787.

[43] Breiman, Leo. ”Random forests.” *Machine learning* 45.1 (2001): 5-32.

[44] Greenacre, Michael, et al. ”Principal component analysis.” *Nature Reviews Methods Primers* 2.1 (2022): 100.

[45] Kiranyaz, Serkan, et al. ”1D convolutional neural networks and applications: A survey.” Mechanical systems and signal processing 151 (2021): 107398.
[46] Liao, Yuchen, et al. ”Attention-based LSTM (AttLSTM) neural network for seismic response modeling of bridges.” Computers & Structures 275 (2023): 106915.
[47] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. ”Learning representations by back-propagating errors.” *nature* 323.6088 (1986): 533-536.
[48] Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. ”Extreme learning machine: theory and applications.” *Neurocomputing* 70.1-3 (2006): 489-501.
[49] Cortes, Corinna, and Vladimir Vapnik. ”Support-vector networks.” *Machine learning* 20.3 (1995): 273-297.
[50] Quinlan, J. Ross. ”Induction of decision trees.” *Machine learning* 1.1 (1986): 81-106.
[51] Zanchi, Marco, et al. ”Influence of microclimatic conditions on dairy production in an Automatic Milking System: Trends and Time-Series Mixer predictions.” Computers and Electronics in Agriculture 229 (2025): 109730.
[52] Nie, Y. ”A Time Series is Worth 64Words: Long-term Forecasting with Transformers.” arXiv preprint arXiv:2211.14730 (2022).
[53] Smith, Wade A., and Robert B. Randall. ”Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study.” *Mechanical systems and signal processing* 64 (2015): 100-131.