Applied Soft Computing Journal 194 (2026) 114964

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A noise-robust feature selection using KNN and weighted fuzzy rough sets for imbalanced multi-scale data

Xiaoyan Zhang*, Jingwen Wang

College of Artificial Intelligence, Southwest University, Chongqing, 400715, PR China

HIGHLIGHTS

- This paper proposes a weighted KNN fuzzy neighborhood rough set (WKNN-FNRS) model.
- The strategy evaluates both noise sensitivity and class imbalance within a single fuzzy rough set-based framework.
- This model integrates feature selection process with scale-based tree reduction and DFS procedure.
- Our method can effectively remove redundant features under varying conditions.

ARTICLE INFO

Keywords:
Weighted KNN fuzzy neighborhood rough set
Generalized multi-scale decision table
Scale-based tree
Dynamic feature selection

ABSTRACT

Feature selection methods that are robust to noise and capable of handling imbalanced data are essential for improving classification accuracy. Although various noise-resistant methods have been developed, few address the combined challenges of noise and class imbalance within a unified framework. The KNN fuzzy neighborhood rough set (KNN-FNRS) model integrates fuzzy set theory with rough set approximations and is effective for feature selection in uncertain environments. However, it remains vulnerable to noisy samples and often performs poorly on highly imbalanced class distributions. To address these limitations, we propose a weighted KNN fuzzy neighborhood rough set (WKNN-FNRS) model. Sample quality evaluation metrics are introduced to quantify instance reliability and assign adaptive neighbor weights that down-weight noisy or low-quality examples. This enhances noise tolerance and improves performance on imbalanced datasets. The WKNN-FNRS model is further embedded in a multi-scale decision framework, and a scale-based tree reduction algorithm is developed. This algorithm constructs and prunes hierarchical scale trees to transform multi-scale data into a single-scale representation. This process preserves both coarse and fine-grained information, improving feature evaluation accuracy. A dynamic feature selection algorithm is then designed to incrementally refine the feature subset based on weighted rough set approximations, avoiding costly retraining when samples or attributes change. For evaluation, we inject class-specific Gaussian perturbations into up to 50% of samples and test on twelve benchmark datasets that span a wide range of imbalance ratios. All experiments follow a stratified nested cross-validation protocol (outer 10-fold, inner 5-fold); feature selection and hyperparameter tuning are performed strictly inside the outer training folds to avoid information leakage. We compare against multiple baselines using KNN, SVM and NB classifiers, report mean ± std over outer folds, and validate significance with appropriate statistical tests. Ablation studies (No Weight, No Fuzzy, Single Scale) and targeted parameter-sensitivity analyses confirm the contribution of each module and the stability of the method. Results show that WKNN-FNRS with scale reduction and DFS consistently outperforms competing methods in classification accuracy, noise resilience and computational efficiency.

1. Introduction

Feature selection methods that are robust to noise [1] and can handle class imbalance are crucial for maintaining high classification accuracy across diverse applications [2]. Such robust feature selection methods have broad applicability, with potential uses ranging from medical diagnosis [3] to fault detection [4]. Because real-world datasets often contain mislabeled or corrupted instances and exhibit class imbalance, it is imperative to develop feature-selection methods that can both filter unreliable observations and preserve rare but informative patterns.

* Corresponding author.
Email addresses: zxy19790915@163.com (X. Zhang), 18154181125@163.com (J. Wang).

https://doi.org/10.1016/j.asoc.2026.114964
Received 7 September 2025; Received in revised form 10 February 2026; Accepted 5 March 2026
Available online 6 March 2026

Researchers have proposed many strategies to reduce dimensionality, remove redundant or irrelevant features [5], and improve generalization [6,7]. Within this research landscape, rough set theory and its fuzzy extensions [8] have attracted considerable interest due to their ability to manage uncertainty [9] without requiring prior knowledge of underlying data distributions. For instance, Yin et al. [10] proposed a parameterized fuzzy-neighborhood similarity relation and developed a noise-robust multi-label FNRS (NT-MLFNRS) along with two multi-label feature selection methods (NRFSFN and ENFSFN) to boost noise resistance in multi-label learning. Sang et al. [11] introduced a fuzzy dominance neighborhood rough set (FDNRS) for dynamic ordinal data. Wang et al. [1] presented an FNRS-based subset selection approach that reduces the number of attributes while improving classification accuracy. Xu and Li [12] proposed a multi-label rough set theory based on KNN to perform multi-label feature selection on imbalanced data. Xu and Tian [13] proposed a technique for feature selection and information fusion based on preference ranking organization method in interval-valued multi-source decision-making information systems. Feng and Zhang [14] devised an algorithm to address supervised incremental feature selection using a regularization vector for dynamic multi-scale interval valued datasets. More recently, Zhang and Zhao [15] devised a generalized multi-granulation FNRS using composite entropy to better handle heterogeneous datasets and accelerate the learning process. Zhang and Shen [16] proposed a graph-driven feature selection method for interval-valued datasets via granular rectangular neighborhood rough sets. Taken together, these studies have advanced fuzzy-rough neighborhood approaches in various directions, including noise resistance, multi-scale handling, and incremental updates. However, many of these methods still rely on fixed neighborhood structures, lack explicit mechanisms for handling severe class imbalance, and face difficulties in efficiently integrating multi-scale information.

Among the various extensions of fuzzy neighborhood rough set theory, KNN-based formulations have attracted particular attention due to their simplicity, interpretability, and strong empirical performance. As research continues to deepen the KNN fuzzy neighborhood rough set (KNN-FNRS) framework [17] has emerged as an effective tool for approximating lower and upper bounds of decision classes by leveraging fuzzy membership functions over a sample's K-nearest neighbors [11]. This approach facilitates the evaluation of feature significance even when class boundaries overlap or when attribute values are imprecise. Despite its success, the standard KNN-FNRS model has several key limitations. First, it treats all neighbor contributions equally [18]. Second, the model is sensitive to approximation operators [19,20] and to mislabeled samples [21,22] both of which can distort neighborhood structures and degrade feature selection quality. In imbalanced scenarios where majority-class samples greatly outnumber minority instances, the model's global weighting scheme fails to emphasize rare patterns, resulting in poor detection of minority-class instances [23,24]. Moreover, emerging methods increasingly involve multi-scale data [25,26] representations such as hierarchical signal transforms, multi-resolution imaging, and multi-granularity features [27], in which selecting the most informative scales introduces additional complexity [28]. Generalized multi-scale information systems (GMIS) [29] provide a flexible framework to encode data at multiple scales, yet systematic scale reduction within GMIS remains underexplored in the context of rough set based feature selection [30].

To address the aforementioned shortcomings, as summarized in Table 1, we develop a unified framework that incorporates adaptive neighbor weighting, efficient multi-scale reduction, and incremental feature selection. First, we propose a weighted KNN fuzzy neighborhood rough set (WKNN-FNRS) model that adaptively adjusts neighbor contributions based on intra-class standard deviation and sample-quality evaluation value. The former measures how well an instance aligns with its same-class neighbors, while the latter quantifies its separability from other class samples. By integrating these metrics into the fuzzy membership function, WKNN-FNRS down-weights noisy or borderline points and emphasizes high-quality observations, thereby bolstering robustness to outliers and mitigating class-imbalance effects.

Table 1
The summary of existing methods.

Methods	Categories	Disadvantages
Fuzzy neighborhood rough set theory	For the original fuzzy neighborhood rough set [10,15,19,31]	Not suitable for noisy datasets.
	For the KNN fuzzy neighborhood rough set neighborhood [12,32,33]	Sensitive to approximation operators and mislabeled samples.
Weighted neighborhood rough set theory	Using a weighted method [9,23]	Reducing robustness to mislabeled and low-quality neighbors.
	Using a sample-quality method [6,18]	Class imbalance is often not explicitly handled.
Multi-scale feature selection methods	Using a scale and feature score method [14,34]	Naive multi-scale aggregation can dilute discriminative scale-specific signals.
	Static feature selection frameworks for changing data [35,36]	The computational cost is high.

Furthermore, we embed the WKNN-FNRS model within a generalized multi-scale information system (GMIS) framework and develop a scale-based tree reduction algorithm to identify the most discriminative scales. Specifically, we build a hierarchical scale tree and perform a depth-first search guided by information-gain and conflict-detection criteria; this process efficiently prunes redundant or uninformative scales. This scale-based tree methodology not only streamlines the feature space across scales but also preserves multi-scale information that may prove critical for complex pattern recognition tasks.

Ultimately, recognizing the dynamic nature of modern data streams [37,38], we design a dynamic feature selection (DFS) algorithm that incrementally updates the selected subset as samples or attributes change. Unlike traditional static methods that mandate complete retraining, our DFS algorithm exploits weighted rough set approximations to reassess feature relevance, dramatically reducing computational overhead and enabling real-time adaptation in evolving environments. The main contributions of this paper are summarized as follows.

1) By jointly addressing noise sensitivity and class imbalance within a single fuzzy rough set-based framework, WKNN-FNRS fills a critical gap in existing literature.
2) The scale-based tree reduction algorithm and DFS procedure collectively enable robust feature selection across both static multi-scale and dynamic streaming scenarios, thereby extending applicability to a broad spectrum of real-world use cases.
3) Dynamic feature selection algorithm that, by continuously adapting to changes in sample composition or feature characteristics, can effectively remove redundant features under varying conditions.

The remainder of this paper is organized as follows. Section 2 reviews the theoretical foundations of fuzzy neighborhood rough sets and generalized multi-scale information systems. Section 3 details the proposed WKNN-FNRS model and the method of weight calculation. Section 4 describes the scale-based tree reduction algorithm, as well as both static and dynamic feature-selection procedures. Section 5 presents the experimental setup, evaluation metrics, and comparative results. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Preliminaries

In this section, we review the theoretical foundations of fuzzy neighborhood rough sets (FNRS), which form the basis of the proposed

feature selection framework. Additionally, we introduce the generalized multi-scale decision system (GMSDS), a key component for modeling multi-scale data representations used in our method.

2.1. Fuzzy neighborhood rough set

Definition 1. Let $\mathcal{F} = \langle U, A, D, f \rangle$ be a fuzzy information system. Here U is the universe of objects, A is the set of conditional attributes, D is the set of decision attributes, and $f : U \times (A \cup D) \to [0,1]$ assigns a membership value to each object-attribute pair. To measure the similarity between two objects $x, y \in U$, a threshold parameter $\lambda \in [0,1]$ is introduced. For each attribute $a \in A$, the similarity relation is defined as:

$$R_a(x,y) = \begin{cases} 0 & |f_a(x) - f_a(y)| > \lambda \\ 1 - \left| f_a(x) - f_a(y) \right| & |f_a(x) - f_a(y)| \le \lambda. \end{cases} \tag{1}$$

For $\forall B \subseteq A$, the combined fuzzy relation is defined as

$$R_B^{\text{comb}}(x,y) = \min_{a \in B} R_a(x,y), \tag{2}$$

and the corresponding neighborhood granule of x, determined by the similarity threshold $(1-\lambda)$, is defined as

$$\widetilde{\alpha}_B(x) = \{ y \in U \mid R_B^{\text{comb}}(x,y) \ge 1 - \lambda \}. \tag{3}$$

Based on this granule, the lower and upper approximations of a decision class D_i are respectively given by

$$\underline{R}_B^{\text{comb}}(D_i) = \{ x \in U \mid \widetilde{\alpha}_B(x) \subseteq D_i \}, \tag{4}$$

$$\overline{R}_B^{\text{comb}}(D_i) = \{ x \in U \mid \widetilde{\alpha}_B(x) \cap D_i \neq \emptyset \}. \tag{5}$$

From these, the positive region of D with respect to B is defined as

$$POS_B(D) = \bigcup_{i=1}^{r} \underline{R}_B^{\text{comb}}(D_i), \tag{6}$$

and the dependency degree of D on B is calculated by

$$\gamma_B(D) = \frac{\left| POS_B(D) \right|}{|U|}. \tag{7}$$

In Eq. (1), $R_a(x,y)$ measures similarity on a single attribute, while Eq. (2) aggregates these minima to $R_B^{\text{comb}}(x,y)$. Eq. (3) uses the threshold $1-\lambda$ to define the fuzzy neighborhood $\widetilde{\alpha}_B(x)$. If every member of this neighborhood belongs to class D_i, Eq. (4) places x in the lower approximation $\underline{R}_B^{\text{comb}}(D_i)$; if the neighborhood only intersects D_i, Eq. (5) assigns x to the upper approximation $\overline{R}_B^{\text{comb}}(D_i)$. Eqs. (6) and (7) define the positive region $POS_B(D)$ and the dependency degree $\gamma_B(D)$, respectively, which together quantify how many samples are classified reliably by B and the overall contribution to classification.

2.2. Generalized multi-scale decision table

The generalized multi-scale decision system (GMSDS), a decision-oriented instance of a generalized multi-scale information system (GMIS) is designed to handle datasets in which both conditional and decision attributes are represented in multiple scales.

Definition 2. A GMSDS is formally defined as a triple $S = \langle U, C \cup D \rangle$. Each attribute a_i is associated with a set of scale levels, forming a multi-scale structure given by: $C = \left\{ a_i^k \mid k = 1,2,\dots,M_i,\ i = 1,2,\dots,m \right\}$, where k denotes the scale level and i indexes the corresponding attribute. Likewise, the decision attribute set D also supports multiple scales, expressed as: $D = \left\{ d^t \mid t = 1,2,\dots,n \right\}$, in which t indicates the scale level for the decision attribute.

Table 2
Generalized Multi-scale decision table.

Samples	A_1			A_2			A_3			d	
	A_1^1	A_1^2	A_1^3	A_2^1	A_2^2	A_2^3	A_3^1	A_3^2	A_3^3	d^1	d^2
x_1	2	2	2	3	3	3	0	1	2	0	1
x_2	3	3	3	3	3	3	0	1	2	0	1
x_3	2	2	2	1	1	2	0	1	2	1	1
x_4	2	2	2	2	2	2	1	1	2	1	1
x_5	3	3	3	1	1	2	0	1	2	1	1
x_6	2	2	2	1	1	2	0	1	2	1	1
x_7	1	1	2	2	2	2	2	2	2	2	2
x_8	1	1	2	1	1	2	0	1	2	2	2
x_9	2	2	2	2	2	2	0	1	2	2	2
x_{10}	0	1	2	0	1	2	3	3	3	2	2

Table 2 presents an illustrative example of a GMSDS, which is used to demonstrate how conditional and decision attributes are represented at multiple scales. As shown, the dataset contains ten samples. The conditional attributes A_1, A_2, A_3 are each represented at three distinct scales, while the decision attribute d is represented at two scales d^1 and d^2.

3. Weighted KNN fuzzy neighborhood rough set

Fuzzy neighborhood rough sets often rely on a fixed radius or global threshold, which makes them sensitive to noise and class imbalance. To improve local adaptation, we propose a weighted KNN-based variant that replaces the global radius with a sample-specific set of K nearest neighbors(KNN).

3.1. KNN fuzzy neighborhood rough set model

Definition 3. Let $S = \langle U, A, D \rangle$ denote a decision information system. For a subset of attributes $B \subseteq A$, the distance between two samples x and y is defined as

$$d_B(x,y) = \sqrt{\sum_{a \in B} (x^a - y^a)^2}, \tag{8}$$

where x^a and y^a denote the values of samples x and y for attribute a.

Based on this, we define a fuzzy similarity relation induced by B as

$$R_B^{\text{sim}}(x,y) = \frac{1}{1 + d_B(x,y)}. \tag{9}$$

This relation satisfies reflexivity and symmetry, and can be interpreted as a similarity degree between two samples. As the distance between two samples decreases, $R_B^{\text{sim}}(x,y)$ approaches 1. For any $x \in U$, we define $k_B(x)$ as the set of its K most similar samples according to R_B^{sim}, thereby forming the fuzzy KNN neighborhood.

Definition 4. Given a partition of U by decision classes $U/D = \{D_1, D_2, \dots, D_r\}$, the upper and lower approximations of a subset $X \subseteq U$ are defined as:

$$\overline{R}_B^{\text{sim}}(X) = \{ x \in U \mid k_B(x) \cap X \neq \emptyset \}, \tag{10}$$

$$\underline{R}_B^{\text{sim}}(X) = \{ x \in U \mid k_B(x) \subseteq X \}. \tag{11}$$

When applied to decision classes, the upper and lower approximations with respect to the decision attribute are given by:

$$\overline{R}_B^{\text{sim}}(D) = \bigcup_{i=1}^{r} \overline{R}_B^{\text{sim}}(D_i), \tag{12}$$

$$\underline{R}_B^{\text{sim}}(D) = \bigcup_{i=1}^{r} \underline{R}_B^{\text{sim}}(D_i). \tag{13}$$

Table 3
A fuzzy decision system.

U	a_1	a_2	a_3	a_4	a_5	a_6	d
x_1	0.58	0.13	1.00	0.78	0.16	0.51	1
x_2	0.45	0.40	0.73	0.57	0.00	0.82	1
x_3	0.40	0.33	0.80	0.63	0.10	0.77	1
x_4	0.36	0.48	0.45	0.59	0.24	0.20	2
x_5	0.85	0.39	0.51	0.29	0.02	0.15	2
x_6	0.55	0.63	0.42	0.75	0.88	0.07	2

Table 4
Neighbor weights.

Sample	Neighbor weights		
x_1	$\omega(x_2) = 0.75$	$\omega(x_3) = 0.75$	$\omega(x_4) = 0.64$
x_2	$\omega(x_1) = 0.81$	$\omega(x_3) = 0.75$	$\omega(x_4) = 0.64$
x_3	$\omega(x_1) = 0.81$	$\omega(x_2) = 0.75$	$\omega(x_4) = 0.64$
x_4	$\omega(x_3) = 0.75$	$\omega(x_5) = 0.48$	$\omega(x_6) = 0.86$
x_5	$\omega(x_1) = 0.81$	$\omega(x_2) = 0.75$	$\omega(x_4) = 0.64$
x_6	$\omega(x_1) = 0.81$	$\omega(x_4) = 0.64$	$\omega(x_5) = 0.87$

Definition 5. The positive region, which is the set of samples that can be definitively classified under B, is equivalent to the lower approximation:

$$POS_B^{KNN}(D) = \underline{R}_B^{\text{sim}}(D). \tag{14}$$

The dependency degree $\gamma_B^{KNN}(D)$ quantifies the extent to which the decision attribute is determined by B. It is defined as

$$\gamma_B^{KNN}(D) = \frac{|POS_B^{KNN}(D)|}{|U|}. \tag{15}$$

Example 1. To illustrate the model, consider a fuzzy decision system with six samples and six conditional attributes as shown in Table 3. The decision attribute d contains two distinct class values.

For $K = 3$, pairwise distances are computed using Eq. (8), and the corresponding fuzzy similarity matrix S is obtained via Eq. (9).

$$S = \begin{bmatrix} 1 & 0.64 & 0.69 & 0.56 & 0.53 & 0.47 \\ 0.64 & 1 & 0.86 & 0.58 & 0.54 & 0.45 \\ 0.69 & 0.86 & 1 & 0.59 & 0.53 & 0.46 \\ 0.56 & 0.58 & 0.59 & 1 & 0.61 & 0.58 \\ 0.53 & 0.54 & 0.53 & 0.61 & 1 & 0.49 \\ 0.47 & 0.45 & 0.46 & 0.58 & 0.49 & 1 \end{bmatrix}$$

The neighborhood of each sample is constructed by selecting its three most similar instances:

$$k(x_1) = \{x_2, x_3, x_4\}, \quad k(x_2) = \{x_1, x_3, x_4\},$$
$$k(x_3) = \{x_1, x_2, x_4\}, \quad k(x_4) = \{x_3, x_5, x_6\},$$
$$k(x_5) = \{x_1, x_2, x_4\}, \quad k(x_6) = \{x_1, x_4, x_5\}.$$

Fig. 1(a) visually demonstrates the fuzzy KNN neighborhood for a sample. The solid red circle represents the target sample, and the large black circle represents the K-nearest neighbor samples of the target sample. This illustration intuitively shows how introducing neighbor weights can reduce the influence of noisy or peripheral samples, allowing neighborhoods to focus on higher-quality samples.

3.2. Weight calculation method

Definition 6. Let $S = \langle U, A, D \rangle$ be a decision system partitioned into classes $U/D = \{D_1, \ldots, D_r\}$. For each decision class $D_i \in U/D$, we define $C_i = \{ y \in U \mid y \in D_i \}$, and compute the distance between two samples in the same class $x, y \in C_i$, using Eq. (8).

Definition 7. The intra-class standard deviation of x is denoted $\sigma_A(x)$, capturing the dispersion of samples within its decision class. We then assign each neighbor y a weight:

$$\omega(y) = \frac{1}{1 + \sigma_A(y)}, \tag{16}$$

so that tightly clustered samples ($\sigma_A \to 0$) approach weight 1, while dispersed or outlying samples receive weights closer to 0.

Definition 8. Combining this weight with the original similarity $R_A^{\text{sim}}(x, y)$ yields the weighted similarity:

$$\mu_A^{\omega}(x, y) = \omega(y) \cdot R_A^{\text{sim}}(x, y). \tag{17}$$

Through this weighting, noisy or peripheral neighbors contribute less, and the resulting neighborhood concentrates on high-quality samples (Fig. 1b). Table 4 lists $\sigma_A(x)$ and $\omega(x)$ for the example in Example 1.

3.3. Weighted KNN fuzzy neighborhood rough set model

We extend the fuzzy neighborhood rough set with a neighbor-weighting mechanism that evaluates each neighbor's reliability.

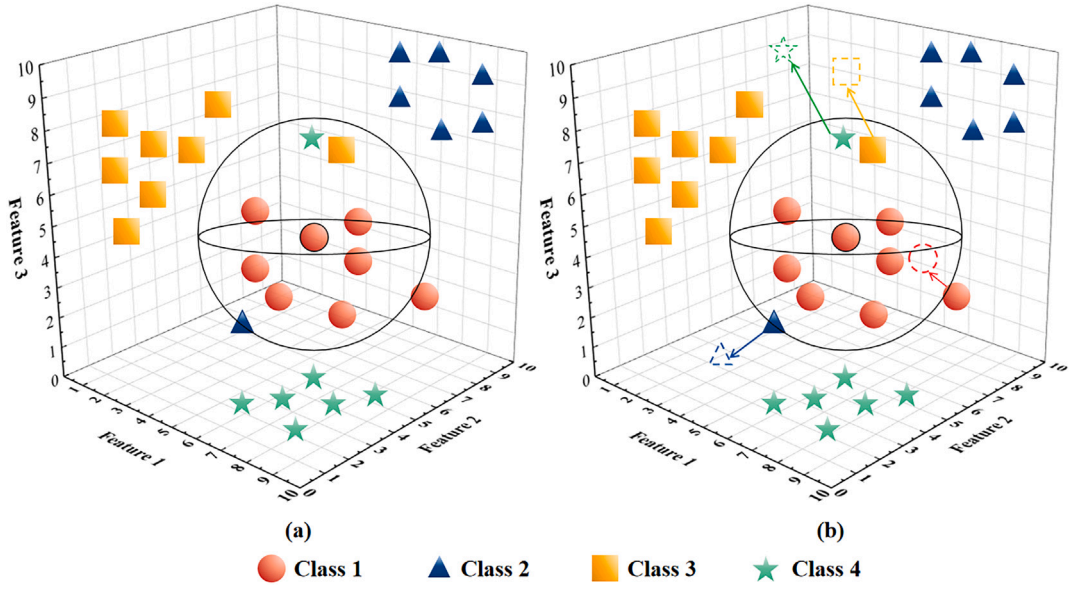

Fig. 1. KNN fuzzy neighborhood and weighted KNN fuzzy neighborhood.

Definition 9. Let $S = \langle U, A, D\rangle$ be a fuzzy decision system with decision-class partition $U/D = \{D_1, \dots, D_r\}$, and let $B \subseteq A$ be a subset of conditional attributes. For $\forall x \in D_i$, define its sample quality evaluation under the K most similar sample set $k(x)$ as:

$$SQE^{WKNN}(x) = \sum_{\substack{y\in k(x)\\ y\in D_i}} \mu_A^{\omega}(x,y) - \sum_{\substack{y\in k(x)\\ y\notin D_i}} \mu_A^{\omega}(x,y). \tag{18}$$

This quantity reflects how strongly x is supported by same-class neighbors compared with neighbors from other classes.

Definition 10. For any subset $X \subseteq U$ and a small threshold $\alpha > 0$, the lower and upper approximations under B are expressed as:

$$\underline{R}_B^{WKNN}(X) = \{ x \in U \mid SQE^{WKNN}(x) > \alpha\}, \tag{19}$$

$$\overline{R}_B^{WKNN}(X) = \{ x \in U \mid SQE^{WKNN}(x) \geq -\alpha\}, \tag{20}$$

where α is chosen close to 0 to allow a modest tolerance.

Definition 11. For the decision attribute D, we aggregate these set-level approximations across all classes:

$$\underline{K}_B(D) = \bigcup_{i=1}^{r} \underline{R}_B^{WKNN}(D_i), \tag{21}$$

$$\overline{K}_B(D) = \bigcup_{i=1}^{r} \overline{R}_B^{WKNN}(D_i). \tag{22}$$

The corresponding boundary region and dependency degree are defined by

$$KBND_B(D_i) = \overline{K}_B(D_i) - \underline{K}_B(D_i), \tag{23}$$

$$K\gamma_B(D) = \frac{\left|\underline{K}_B(D)\right|}{|U|}. \tag{24}$$

In this weighted model, samples whose neighborhoods show strong same-class support are assigned to the lower approximation, whereas those with ambiguous support fall into the boundary region. By incorporating neighbor weights into the similarity measure, the method adapts to noise and class imbalance thereby enabling more reliable feature evaluation and selection.

3.4. Property and proof

Proposition 1. (Monotonicity) *If $B_1 \subseteq B_2 \subseteq C$, $\forall x \in U$ and $\forall D_i$:*

$$\underline{K}_{B_1}(D_i) \subseteq \underline{K}_{B_2}(D_i), \qquad \overline{K}_{B_1}(D_i) \subseteq \overline{K}_{B_2}(D_i).$$

Proof. Adding attributes leads to stricter distance or similarity thresholds:

$$\mu_{B_2}(x,y) \leq \mu_{B_1}(x,y) \Rightarrow \mu_{B_2}^{\omega}(x,y) \leq \mu_{B_1}^{\omega}(x,y).$$

For the same sample, its lower or upper approximation under B_2 will not be less than that under B_1. □

Proposition 2. (Boundary Region Containment) *The boundary region is defined in Eq. (22). If $B_1 \subseteq B_2 \subseteq C$, then:*

$$KBND_{B_2}(D_i) \subseteq KBND_{B_1}(D_i).$$

Proof. By monotonicity:

$$\underline{K}_{B_1}(D_i) \subseteq \underline{K}_{B_2}(D_i) \subseteq \overline{K}_{B_2}(D_i) \subseteq \overline{K}_{B_1}(D_i),$$

the containment property immediately follows from set difference operations. □

4. Feature selection in Weighted KNN fuzzy neighborhood rough set

This section presents our main contributions: a scale-reduction procedure for generalized multi-scale tables, a static feature selection algorithm based on the weighted KNN fuzzy neighborhood rough set framework (SWKNNFS), and an efficient dynamic feature selection mechanism that supports incremental updates.

4.1. Scale reduction of a generalized multi-scale table

In a generalized multi-scale decision table, both conditional and decision attributes are observed at multiple scales. We select the most informative scale through a two-phase procedure: a rough selection followed by a precise selection.

During the rough selection phase, we compute the information gain of each conditional-scale attribute A_i^j with respect to each decision scale d^k: $IG(A_i^j \mid d^k) = H(d^k) - H(d^k \mid A_i^j)$, where $H(\cdot)$ denotes entropy. We then select the decision attribute scale d^{ℓ} that maximizes the aggregated information gain across all conditional scales.

In the precise selection phase, we organize each attribute's scales into a top-down scale-based tree (SBT): the root corresponds to the coarsest scale and the leaves to the finest scales. We traverse the SBT in depth-first order: if all samples covered by a node share the same label under the chosen decision scale, we prune that subtree and accept the current scale for the corresponding attribute. Otherwise, we descend further until label consistency is achieved or the leaf scales are reached. Fig. 2 gives a schematic illustration. For clarity, the figure is schematic only: conditional attributes a_t and decision attributes d_k may have different numbers of scales denoted by F_i and D_j. The scales for decision attributes are selected based on the information gain values shown in Table 5. After fixing the decision scale, the consistency checking results are shown in Table 6. As a result, the original multi-scale table is transformed into an equivalent single-scale decision table that is ready for feature selection.

4.2. Static feature selection algorithm

This section explains how to perform efficient feature selection when samples and features are static.

Definition 12. Let $S = \langle U, A, D\rangle$ be a decision table and $B \subseteq A$ a candidate subset. The importance of attribute a with respect to the current subset B and the decision attribute D is defined as:

$$IMP(a, B, D) = K\gamma_{B\cup\{a\}}(D) - K\gamma_B(D). \tag{25}$$

Features are chosen according to their dependency degree, computed by sample quality evaluation. Algorithm 1 summarizes the pseudocode for static weighted KNN fuzzy neighborhood rough set feature selection. The method first reduces the multi-scale decision table to a single-scale representation, then performs greedy forward selection. Starting from an empty set, it iteratively adds the attribute that maximally increases the fuzzy KNN dependency degree. The process stops when no attribute yields a positive gain, and the selected subset is returned. Fig. 3 shows the overall framework diagram of the proposed method.

4.3. Dynamic feature selection algorithm

This section introduces a dynamic feature selection (DFS) algorithm tailored for environments where samples or attributes are incrementally added or removed, though not both at the same time. To enable fast updates, we precompute and cache three items for any feature subset B: the sample similarity matrix, the positive-region index set, and dependency scores of candidate attributes relative to B and the decision set D.

When a new sample is added (Algorithm 2), we compute its similarities to existing samples, insert it into the appropriate positive region sets,

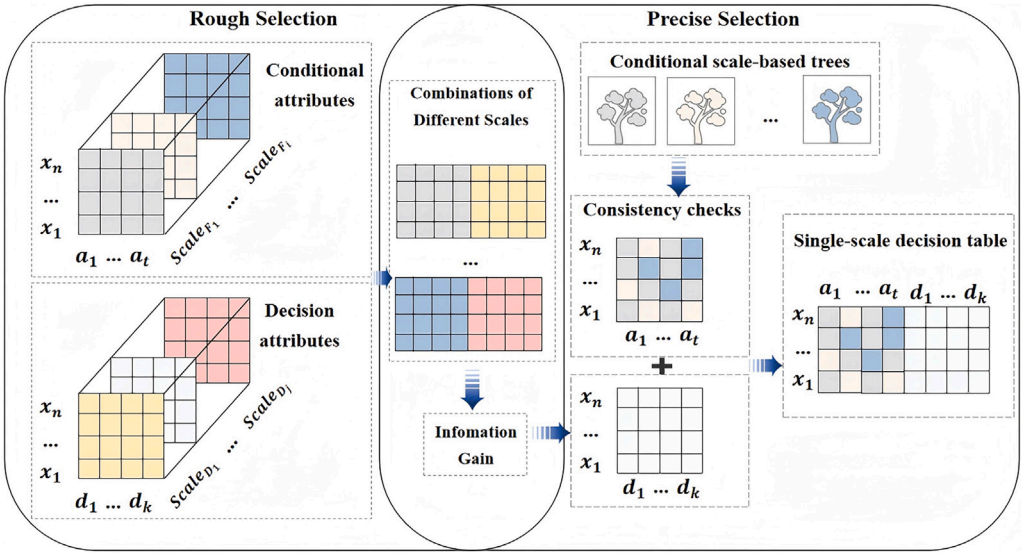

Fig. 2. Schematic diagram of scale-based tree reduction process.

Table 5
Information gain under different scales.

Attribute	$IG(A_i^j \mid d^1)$	$IG(A_i^j \mid d^2)$
A_1^1	0.6364	0.6100
A_1^2	0.3078	0.2813
A_1^3	0.1974	0.1710
A_2^1	0.9219	0.3710
A_2^2	0.7219	0.1710
A_2^3	0.5568	0.2813
A_3^1	0.4323	0.3668
A_3^2	0.3219	0.3219
A_3^3	0.1445	0.1445

Table 6
Consistency Check after Decision Scale Fixation.

Sample	a_1	a_2	a_3	d
x_1	2	3	0	0
x_2	3	3	0	0
x_3	2	1	0	1
x_4	2	2	1	1
x_5	3	1	0	1
x_6	2	1	0	1
x_7	1	2	2	2
x_8	1	1	0	2
x_9	2	2	0	2
x_{10}	1	0	3	2

and update each attribute's importance by its marginal effect on positive region size. Conversely, when a sample is removed (Algorithm 3), we delete its entries from the similarity cache and positive region indices and subtract its contribution from the attribute-importance metrics. Consequently, time and computational cost scale linearly with the number of modified samples and remain effectively independent of the full dataset size provided changes are sparse.

When a new feature is added (Algorithm 4), we treat it as an isolated candidate and evaluate its importance using the existing samples and cached positive regions. If the importance exceeds the threshold, the feature is added to B, the similarity matrix and positive region indices are updated, and other candidates' importance measures are quickly adjusted. Otherwise, the feature is discarded with no structural change. Conversely, in Algorithm 5, a non-B candidate is simply removed, while deleting a feature from B triggers a single recomputation of importance

Algorithm 1: SWKNNFS Algorithm.

Input : A GMDT $S = (U, A, D)$
Output: Feature subset B

$B \leftarrow \emptyset$;
while $A - B \neq \emptyset$ **do**
  forall $a \in A - B$ **do**
    calculate $SQE^{WKNN}(x)$;
    calculate $KPOS_B(X)$;
    calculate $K\gamma_{B\cup\{a\}}(D)$ and $K\gamma_B(D)$;
    calculate $IMP(a, B, D)$;
  end
  $a_1 \leftarrow \arg\max_{a\in A-B} IMP(a, B, D)$;
  if $IMP(a_1, B, D) > 0$ **then**
    $B \leftarrow B \cup \{a_1\}$;
  end
  else
    break;
  end
end
return B;

for remaining candidates and a localized replacement selection, thus avoiding a full re-evaluation of all features.

4.4. Time complexity analysis

For clarity, we use the following notation: $n = |U|$ is the number of samples, $p = |A|$ is the number of candidate features, $s = |B|$ is the current number of selected features, and k is the neighborhood size. Computing the combined similarity between a pair of samples over a feature subset B requires $O(|B|)$ time, so

$$T_{\text{pair}}(B) = O(|B|).$$

Computing the k-nearest neighbors for all n samples by brute-force similarity evaluation therefore requires examining $O(n^2)$ pairs, giving the leading cost

$$T_{\text{all-KNN}}(B) = O\big(n^2 \cdot |B|\big).$$

A naive greedy forward selection that, for each candidate attribute, recomputes positive regions from scratch would incur the following

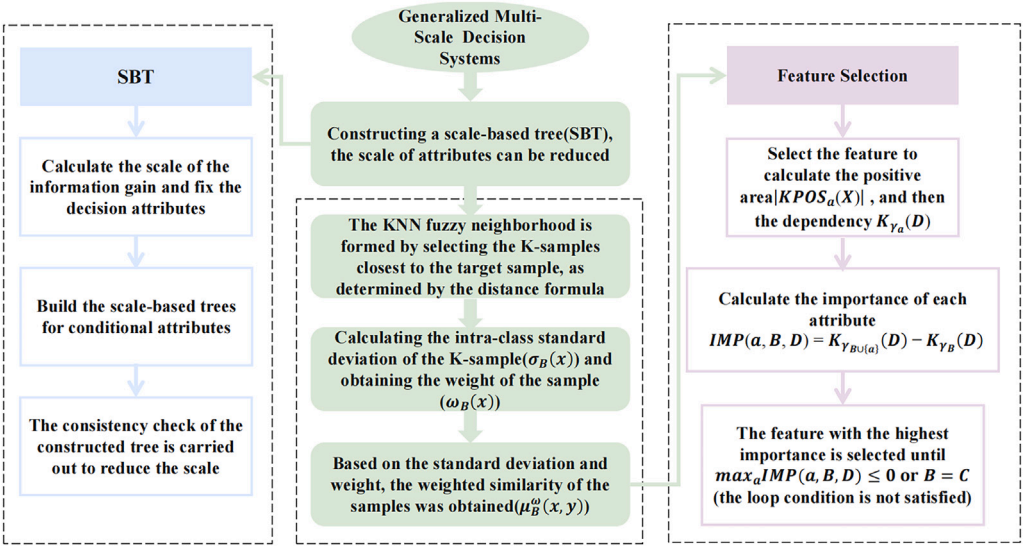

Fig. 3. Framework of our algorithm.

Algorithm 2: DFS for sample addition.

Input: Sample set U, feature set A, feature subset B, new sample x_{new}, parameter α
Output: Updated feature subset B'

$U \leftarrow U \cup \{x_{\text{new}}\}$;
$n \leftarrow |U|$;
$SQE_{\text{vals}} \leftarrow$ calculate $SQE^{WKNN}(B)$;
$K_{\gamma_{\text{prev}}} \leftarrow \text{count}(SQE_{\text{vals}} > \alpha)/n$;
for $a \in A - B$ **do**
  $B_{\text{temp}} \leftarrow B \cup \{a\}$;
  $SQE_{\text{vals\_a}} \leftarrow$ calculate $SQE^{WKNN}(B_{\text{temp}})$;
  $K_{\gamma_{\text{new}}} \leftarrow \text{count}(SQE_{\text{vals\_a}} > \alpha)/n$;
  if $K_{\gamma_{new}} - K_{\gamma_{prev}} > 0$ **then**
    $B \leftarrow B_{\text{temp}}$;
    $K_{\gamma_{\text{prev}}} \leftarrow K_{\gamma_{\text{new}}}$;
  end
end
return B';

Algorithm 3: DFS for sample removal.

Input: Sample set U, feature set A, feature subset B, removed sample x_{remove}, parameter α
Output: Updated feature subset B'

$U \leftarrow U - \{x_{\text{remove}}\}$;
$n \leftarrow |U|$;
$SQE_{\text{vals}} \leftarrow$ calculate $SQE^{WKNN}(B)$;
$K_{\gamma_{\text{prev}}} \leftarrow \text{count}(SQE_{\text{vals}} > \alpha)/n$;
for $a \in A - B$ **do**
  $B_{\text{temp}} \leftarrow B \cup \{a\}$;
  $SQE_{\text{vals\_a}} \leftarrow$ calculate $SQE^{WKNN}(B_{\text{temp}})$;
  $K_{\gamma_{\text{new}}} \leftarrow \text{count}(SQE_{\text{vals\_a}} > \alpha)/n$;
  if $K_{\gamma_{new}} - K_{\gamma_{prev}} > 0$ **then**
    $B \leftarrow B_{\text{temp}}$;
    $K_{\gamma_{\text{prev}}} \leftarrow K_{\gamma_{\text{new}}}$;
  end
end
return B';

worst-case cost:

$$T_{\text{static}} = O\left(\sum_{i=0}^{s_{\max}-1} (p-i)\cdot n^2\cdot(i+1)\right) = O\left(p\cdot n^2\cdot s_{\max}^2\right),$$

where $s_{\max}$ is the maximum number of features selected. This expression highlights that the $O(n^2)$ term quickly dominates as n grows.

To reduce this cost, the DFS implementation relies on three caches: (i) current pairwise similarities, (ii) positive-region index sets, and (iii) cached dependency scores for candidate attributes. With these caches, a single incremental operation primarily costs:

- computing similarities between the new sample and the existing n samples over B: $O(n\cdot|B|)$,
- updating affected KNN lists: $O(n\log k)$ in the worst case,
- recomputing per-sample quality metrics SQE and updating positive-region indices for affected samples: $O(n\cdot k)$ when similarity values are cached.

Hence, the amortized cost for a sparse incremental change is approximately

$$T_{\text{incremental}} = O\left(n\cdot|B| + n\log k + n\cdot k\right),$$

Algorithm 4: DFS for feature addition.

Input: Sample set U, feature set A, feature subset B, new feature a_{new}, parameter α
Output: Updated feature subset B'

$A \leftarrow A \cup \{a_{\text{new}}\}$;
$SQE_{\text{vals}} \leftarrow$ calculate $SQE^{WKNN}(B)$;
$K\gamma_{\text{prev}} \leftarrow \text{count}(SQE_{\text{vals}} > \alpha)/|U|$;
$SQE^{\text{new}}_{\text{vals}} \leftarrow$ calculate $SQE^{WKNN}(B \cup \{a_{\text{new}}\})$;
$K\gamma_{\text{new}} \leftarrow \text{count}(SQE^{\text{new}}_{\text{vals}} > \alpha)/|U|$;
if $K\gamma_{new} - K\gamma_{prev} > 0$ **then**
  $B \leftarrow B \cup \{a_{\text{new}}\}$;
  $K\gamma_{\text{prev}} \leftarrow K\gamma_{\text{new}}$;
end
return B';

which is effectively near-linear in n when the number of affected samples is small. Similarly, evaluating the marginal impact of a single candidate attribute using cached KNNs and per-sample metrics

Algorithm 5: DFS for feature removal.

Input: Sample set U, feature set A, feature subset B, removed feature a_{old}, parameter α
Output: Updated feature subset B'

$A \leftarrow A - \{a_{\text{old}}\}$;
$SQE_{\text{vals}} \leftarrow$ calculate $SQE^{WKNN}(B)$;
$K\gamma_{\text{prev}} \leftarrow \text{count}(SQE_{\text{vals}} > \alpha)/|U|$;
if $a_{old} \in B$ **then**
  $B \leftarrow B - \{a_{\text{old}}\}$;
  $best\_IMP \leftarrow 0$;
  $best\_feat \leftarrow$ null;
  for $a \in A - B$ **do**
    $B_{\text{temp}} \leftarrow B \cup \{a\}$;
    $SQE_{\text{vals\_a}} \leftarrow$ calculate $SQE^{WKNN}(B_{\text{temp}})$;
    $K\gamma_{\text{new}} \leftarrow \text{count}(SQE_{\text{vals\_a}} > \alpha)/|U|$;
    $IMP \leftarrow K\gamma_{\text{new}} - K\gamma_{\text{prev}}$;
    if $IMP > best\_IMP$ **then**
      $best\_IMP \leftarrow IMP$;
      $best\_feat \leftarrow a$;
    end
  end
  if $best\_feat \neq null$ **then**
    $B \leftarrow B \cup \{best\_feat\}$;
    $K\gamma_{\text{prev}} \leftarrow K\gamma_{\text{prev}} + best\_IMP$;
  end
end
return B';

typically takes $O(n \cdot k)$ rather than the full $O(n^2 \cdot |B|)$ recomputation. Therefore, with careful caching and localized updates, DFS avoids repeated global recomputation and scales substantially better in practice than the naive static approach. This theoretical advantage is consistent with the empirical running-time shown in Figs. 6 and 7.

5. Experimental analysis

In this section, a series of experiments are conducted to validate robustness to noise, effectiveness under class imbalance, benefits of multi-scale reduction, and efficiency in dynamic settings.

5.1. Data preprocessing and experimental setup

1) Data Preprocessing

Table 7 lists the 12 UCI datasets used in our experiments. We converted these datasets into generalized multi-scale decision tables as described in [29]. For the 12 datasets used in our experiments, we quantify class imbalance by the imbalance ratio IR = $n_{\text{major}}/n_{\text{minor}}$, where n_{major} and n_{minor} are the numbers of samples in the majority and minority classes, respectively. Following common practice in the literature [39], we categorize datasets into four groups by IR values: Balanced (IR $\leq$ 1.5), Slight imbalance (1.5 < IR $\leq$ 3), Moderate imbalance (3 < IR $\leq$ 10), and Severe imbalance (IR > 10). The imbalance ratios for all datasets are reported in Table 7. These statistics demonstrate that our benchmark collection spans the full range from balanced to severely imbalanced datasets.

2) Experimental Settings

The experiments were conducted in two parts: (1) module-level evaluations, and (2) comparisons with alternative feature-selection algorithms.

All experiments in this paper were conducted in `PyCharm 2023.1 (Professional)` on Windows 11, with an Intel® Core™ i5-1135G7 processor @ 2.40 GHz—2.42 GHz and 16.0 GB of RAM.

Part I: Module-by-module Effectiveness Evaluation

Table 7
Summary of datasets.

NO.	Datasets	Objects	Features	Classes	Imbalance ratios
1	PC	90	1177	2	2.333
2	Sonar	208	60	2	1.144
3	Derm	358	34	6	5.600
4	BCW	699	9	2	1.900
5	SGCD	1000	20	2	2.333
6	MPE	1080	80	8	1.429
7	WIL	2000	7	4	1.000
8	EOLBEHPC	2111	16	7	1.290
9	PSDAS	4424	36	3	2.782
10	SBD	6321	13	2	8.364
11	EES	14,980	14	2	1.228
12	AGMPD	31,991	8	2	55.026

(a) **Robustness test.** To evaluate noise resilience, we progressively corrupt each dataset using class-specific Gaussian perturbations. For a chosen noise level $\eta \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, we compute $m = \lfloor \eta \cdot n \rfloor$ and randomly select m samples (without replacement) to corrupt. For each selected sample i of class c and numeric feature j, we replace $x_{i,j}$ with $\tilde{x}_{i,j} = x_{i,j} + \epsilon_{i,j}$, where $\epsilon_{i,j} \sim \mathcal{N}(0, (\alpha_c \cdot \text{std}(X_{\cdot,j}))^2)$. Here $\text{std}(X_{\cdot,j})$ denotes the empirical standard deviation of feature j computed on the training data and $\alpha_c \geq 0$ is a class-specific relative noise scale. Unless stated otherwise, we use the same α_c for all classes. For each corrupted dataset, we run our feature-selection pipeline and evaluate classifier accuracy using stratified 10-fold cross-validation with KNN, SVM, Random Forest (RF), and Logistic Regression (LR).

(b) **SBT effectiveness test.** Each dataset is converted to a generalized multi-scale decision table. We evaluate classifier performance using the coarsest scale, the finest scale, and the SBT-reduced scale, and measure the average accuracy of KNN, SVM, Random Forest (RF), and Logistic Regression (LR) on each scale to assess the benefit of multi-scale reduction.

(c) **Time consumption under dynamic changes.** We measure runtime for dynamic changes in samples and features. For sample addition, we start from 50% of the dataset and increase the number of samples in 10% increments up to 100%. For sample removal, we start from 100% of the dataset and decrease the number of samples in 10% steps down to 50%. The same scheme is applied for feature dynamics. At each step we record the feature-selection time and compare the dynamic method to a static baseline.

(d) **Effect of feature-selection ratio.** To study how the fraction of retained features affects predictive performance, we conducted a feature-selection-ratio experiment. Let m denote the original number of features and r the retention ratio; for each r we set the selector to retain $k = \max(1, \lceil r \cdot m \rceil)$ features. We evaluated $r \in \{0.05, 0.10, 0.20, 0.30, 0.50, 1.00\}$. For unbiased estimation we used stratified 10-fold outer cross-validation, ensuring that feature selection was performed strictly inside each outer training fold.

Part II: Comparison with Other Algorithms

(a) **Comparative experiment selection.** We compare our method against eight established feature selection algorithms: Algorithm WKNFS [33] uses a forward greedy search based selector to select features. Algorithm AAnfsCF [11] is a noise-resistant method for noisy classification tasks. Algorithm FC [28] is a fast selection method based on multi-scale interval valued information. Algorithm LVPCM [40] uses variable precision neighborhood rough set to select features (we use the precision combination with highest accuracy). Algorithm IBMOA [41] integrates periodic pattern boundary handling (PMBH) and local search (LS) to balance exploration and exploitation. Algorithm FNGCE [15] is based on generalized multi-granular fuzzy rough set for feature

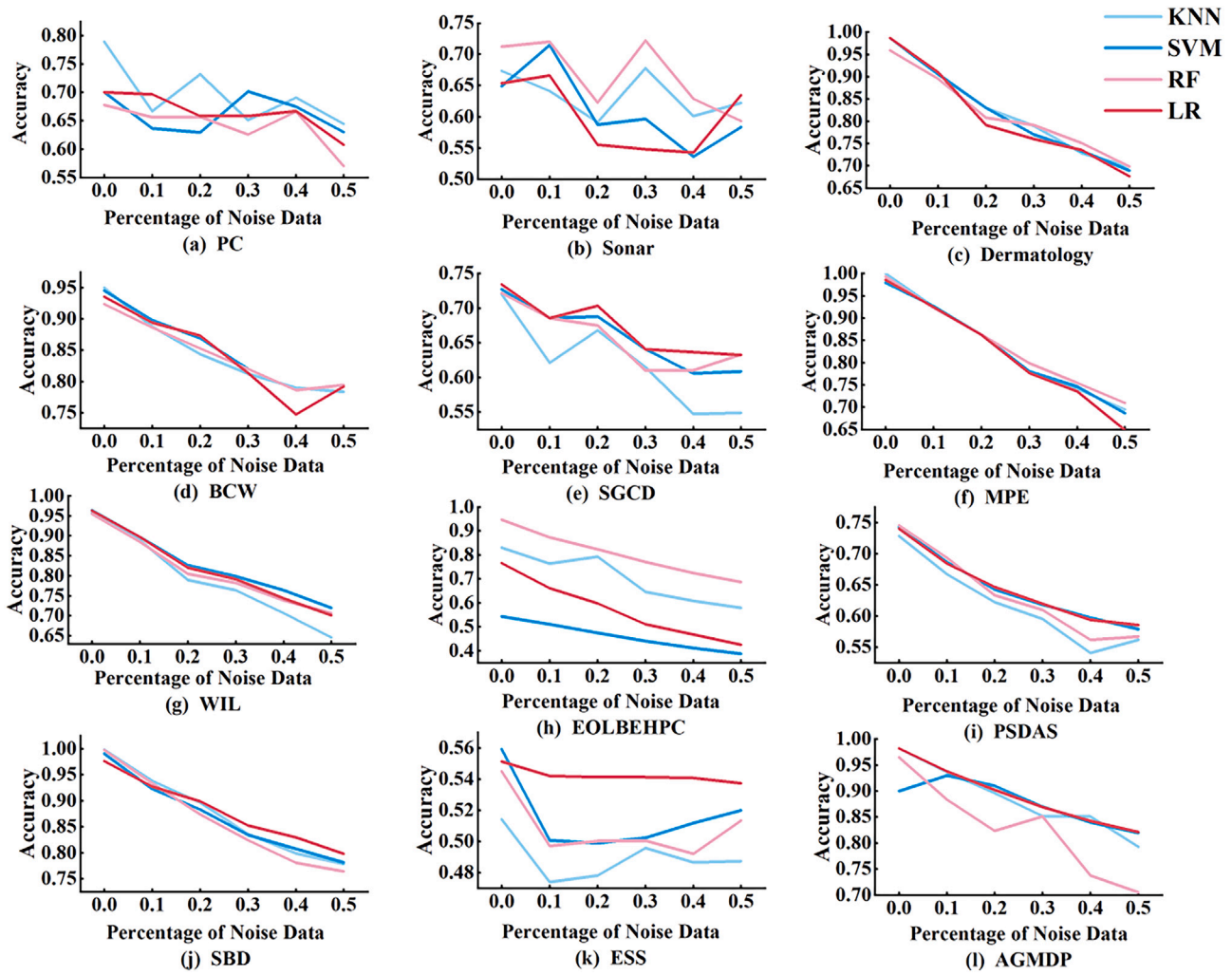

Fig. 4. Classification accuracy under increasing noise levels for four classifiers.

selection. Algorithm WGMIE [42] uses a generalized multi-granular advantage neighborhood rough set model with forward heuristic search. Algorithm MFS [43] employs information gain filtering in a coarse selection phase to reduce redundancy.

(b) **Cross-validation and hyperparameter tuning protocol.** To avoid information leakage and ensure fair comparisons, we adopt a nested cross-validation protocol for all experiments. The outer loop is stratified 10-fold cross-validation: in each outer iteration one fold is held out as the test set while the remaining folds form the outer training set. All feature selection procedures and any data-driven preprocessing are executed exclusively within the outer training set. Hyperparameters (including neighborhood size K and threshold α) are tuned by a stratified 5-fold inner cross-validation performed on the outer training set only. We apply identical hyperparameter ranges, inner-fold partitions, evaluation metrics, and random seeds across all compared methods. After selecting optimal hyperparameters in the inner loop, the full pipeline is retrained on the outer training set and evaluated on the outer test fold. Final performance is reported as the mean and standard deviation across the outer folds.

5.2. Experimental results and analysis

Fig. 4 presents the changes in accuracy as label noise increases from 0% to 50% across the twelve benchmarks and multiple classifiers. Overall, accuracy declines for all methods, but our approach shows a noticeably gentler fall and smaller variance in many cases (for example, PC improves from 71.11% to 80.00% under our selection, MPE reaches 99.98%, and Dermatology exhibits a variability of ±1.33%). These patterns indicate that down-weighting unreliable neighbors stabilizes neighborhood statistics and reduces the influence of mislabeled samples, yielding a more robust feature-selection outcome under label corruption.

The comparisons in Fig. 5 focus on multi-scale strategies. Using the SBT-reduced scales produces larger performance regions on the radar plots than using only the finest or coarsest scales or naively aggregating all scales; on average SBT yields gains of 8.9% versus the coarsest and 4.3% versus the finest scale, with peak accuracies up to 93.7% in some runs. By preferring scales with higher label consistency, SBT avoids diluting discriminative information with redundant granularities and provides a compact set of scales that preserves useful multi-scale signals without the cost of exhaustive search.

Runtime behavior is summarized in Figs. 6 and 7, which compare static full recomputation with our dynamic incremental updates under varying numbers of samples and features. Across datasets, the dynamic algorithm runs substantially faster and scales more gently as updates grow; static runtimes increase rapidly, especially when many attributes are present. The incremental scheme achieves these savings by reusing cached neighbor relations and updating only affected neighborhoods and feature scores, which reduces both CPU time and memory overhead and makes the method practical for evolving datasets.

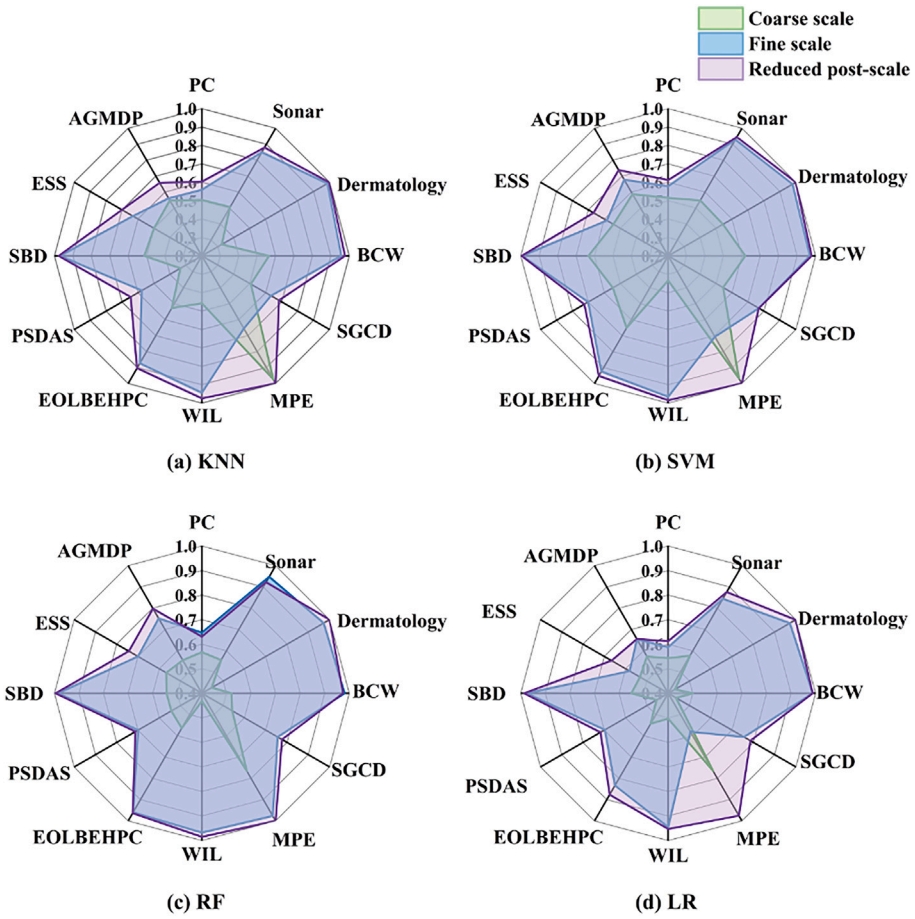

Fig. 5. Classification accuracy of reduced feature sets at different scales across four classifiers.

Tables 8–10 show classification results using KNN, SVM and Naive Bayes on the selected feature subsets. Our method attains consistently strong performance—above 96% on BCW and SBD and 98.22% on AGMPE—and often yields lower variance than competing selectors, indicating stable generalization under different classifiers and noise conditions. The imbalance-aware neighborhood aggregation contributes to these outcomes by normalizing neighbor contributions and amplifying neighborhoods that are consistent with minority labels, thereby preserving features that matter for under-represented classes and mitigating majority-class domination.

Fig. 8 examines how performance depends on the fraction of retained features. Most datasets reach near-peak accuracy once roughly 20–30% of features are kept, with rapid gains observed between 5% and 30% retention; beyond 40% some datasets even experience slight performance drops due to added redundancy or noise. This behavior confirms that the proposed selector efficiently identifies a compact subset of informative attributes, achieving a favorable trade-off between parsimony and predictive power.

5.3. Ablation study

To quantify the contribution of each component in the proposed WKNN-FNRS framework, we performed a systematic ablation study comparing four variants: the complete method (**Full**), a variant without neighborhood weighting (**No Weight**), a variant that replaces fuzzy membership with a neighbor-agreement ratio (**No Fuzzy**), and a single-scale variant that uses only one scale of features (**Single Scale**). All ablation experiments follow the same evaluation protocol as the main experiments: stratified 10-fold outer cross-validation is employed for unbiased estimation, and feature selection is performed strictly inside each outer training fold to avoid information leakage. The ablation results are summarized in Tables 11–13. Overall, the Full method achieves the best average performance across datasets and classifiers. Removing the fuzzy membership (No Fuzzy) typically yields the largest drop in predictive performance, indicating that fuzzy membership contributes substantially to modeling local similarity and achieving robustness to noisy neighborhoods. Disabling neighborhood weighting (No Weight) produces a smaller but consistent performance drop, showing that the class-dependent neighborhood weighting helps emphasize informative local structures. The single-scale variant performs competitively on some datasets but degrades performance on tasks that require multi-granular information, demonstrating the utility of multi-scale representations for capturing discriminative patterns. In summary, the ablation analysis validates the necessity of the proposed fuzzy membership and neighborhood weighting modules and provides practical guidance on the expected performance–cost trade-offs when these components are omitted.

5.4. Statistical tests

We apply the Iman-Davenport [44] and Wilcoxon tests to compare nine algorithms across twelve datasets. For the Iman-Davenport test ($\alpha = 0.05$, critical $= 2.045$), the F-statistics for all three classifiers exceed 2.045 and the p-values in Table 14 are below 0.05. Hence we reject the null hypothesis of equal performance and conclude that significant differences exist among the nine algorithms across the datasets.

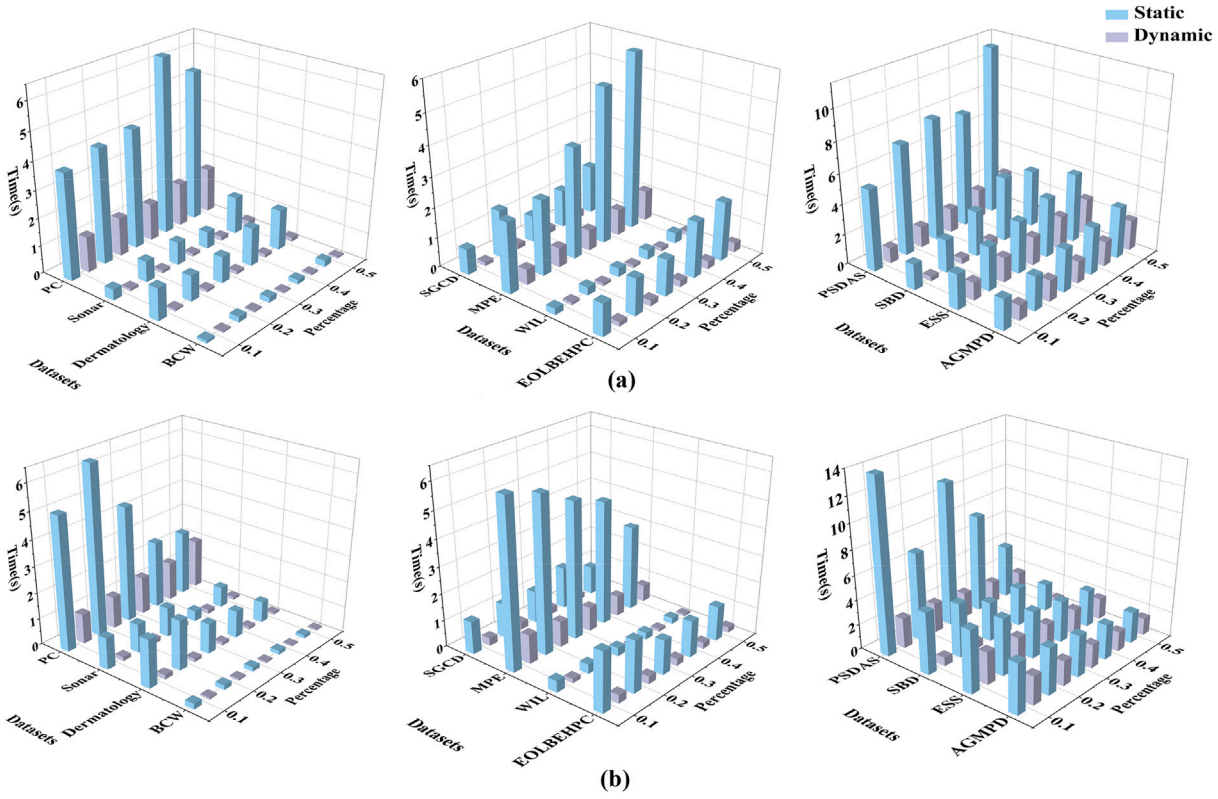

Fig. 6. Runtime when the number of samples increases (a) and decreases (b).

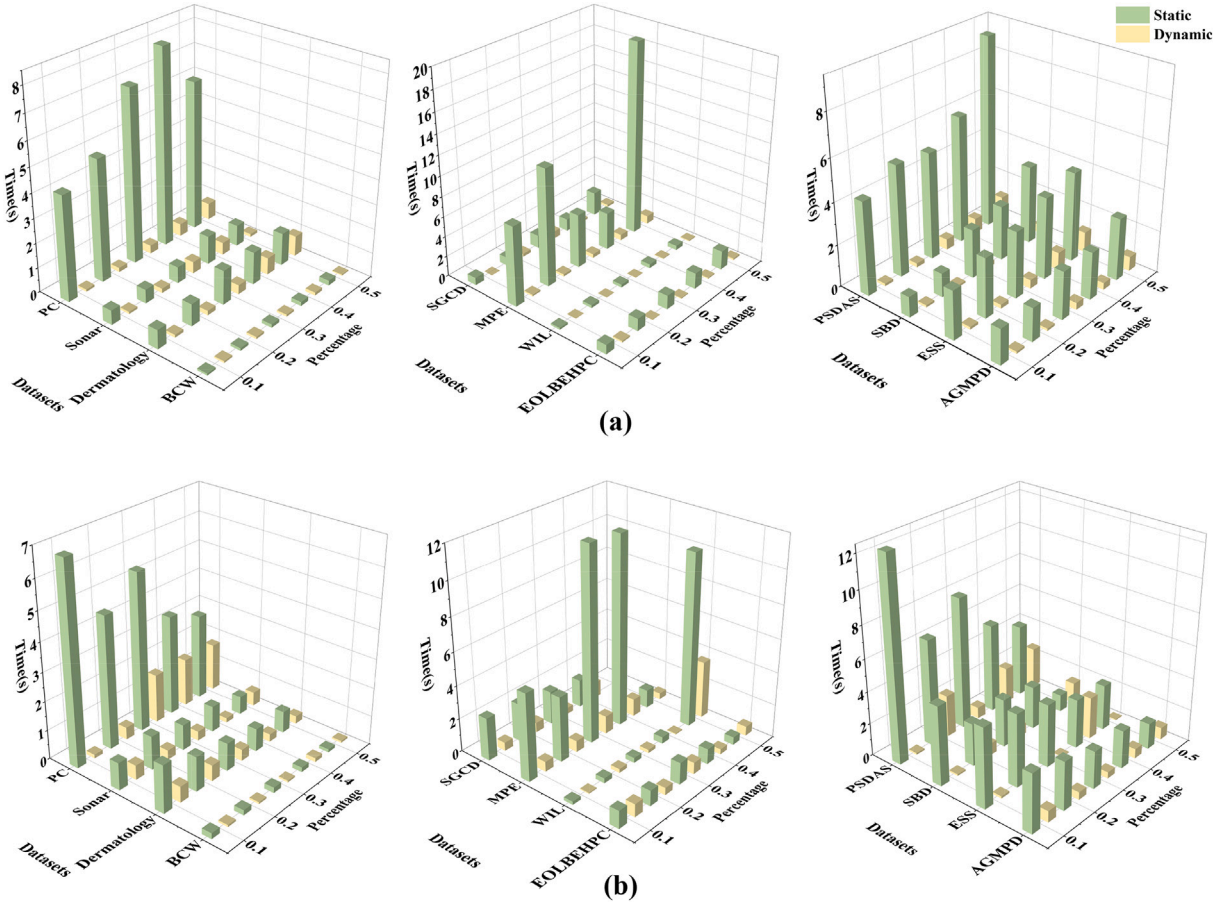

Fig. 7. Runtime when the number of attributes increases (a) and decreases (b).

Table 8
The classification accuracy of selected features on KNN(%).

Datasets	original data	WKNFS	AAnfsCF	FC	LVPCM	IBMOA	FNGCE	WGMIE	MFS	Ours
PC	71.11 ± 16.63	77.15 ± 3.43	68.33 ± 1.24	70.00 ± 19.91	77.50 ± 10.23	67.78 ± 10.48	70.00 ± 8.68	66.67 ± 8.61	67.65 ± 1.34	**80.00 ± 12.96**
Sonar	79.86 ± 9.25	80.69 ± 2.31	79.86 ± 9.25	49.93 ± 11.85	80.69 ± 8.77	80.36 ± 7.99	82.67 ± 8.65	82.71 ± 9.39	49.50 ± 10.15	**85.12 ± 8.12**
Derm	87.69 ± 5.77	98.37 ± 7.88	87.69 ± 5.77	74.89 ± 5.62	98.37 ± 4.97	84.98 ± 6.35	91.53 ± 6.30	96.99 ± 2.26	25.65 ± 6.89	**98.91 ± 1.33**
BCW	93.84 ± 2.32	96.28 ± 3.24	94.33 ± 2.76	95.14 ± 1.59	96.28 ± 1.89	94.71 ± 2.48	96.28 ± 1.59	95.75 ± 2.65	90.42 ± 2.47	**96.86 ± 1.54**
SGCD	69.20 ± 4.28	**72.07 ± 1.98**	69.20 ± 4.28	66.80 ± 4.12	72.07 ± 3.32	66.30 ± 5.22	71.30 ± 3.90	70.80 ± 3.97	68.30 ± 3.61	71.10 ± 2.98
MPE	95.72 ± 1.59	97.82 ± 6.77	81.21 ± 1.31	99.91 ± 0.28	93.33 ± 4.85	97.33 ± 6.74	99.81 ± 0.37	96.87 ± 3.42	26.20 ± 3.04	**99.98 ± 0.21**
WIL	96.75 ± 1.50	98.12 ± 4.20	98.30 ± 0.60	95.80 ± 0.90	98.10 ± 0.76	96.85 ± 1.18	79.10 ± 3.13	98.40 ± 0.70	86.50 ± 1.86	**98.50 ± 0.92**
EOLBEHPC	84.04 ± 2.49	86.97 ± 1.98	83.04 ± 1.99	**87.12 ± 2.35**	86.97 ± 1.29	65.89 ± 2.66	82.57 ± 1.47	78.92 ± 1.77	23.92 ± 1.26	84.04 ± 2.49
PSDAS	61.12 ± 1.92	69.19 ± 4.21	61.12 ± 1.92	59.68 ± 2.25	69.19 ± 1.43	70.93 ± 1.38	68.45 ± 1.27	68.11 ± 1.40	48.10 ± 5.68	**72.81 ± 1.26**
SBD	99.76 ± 0.19	98.89 ± 9.33	98.96 ± 0.37	89.81 ± 1.08	99.86 ± 0.11	99.84 ± 0.10	97.66 ± 0.53	99.02 ± 0.42	88.96 ± 0.90	**99.89 ± 0.87**
EES	56.28 ± 1.69	53.19 ± 6.47	97.06 ± 0.38	96.31 ± 2.16	55.38 ± 1.23	53.77 ± 1.21	60.77 ± 2.36	78.88 ± 3.14	86.17 ± 1.49	**97.06 ± 0.38**
AGMPE	98.16 ± 0.16	**98.34 ± 3.54**	96.36 ± 0.64	94.72 ± 2.19	98.34 ± 1.78	97.54 ± 1.22	94.55 ± 1.56	91.66 ± 2.67	84.31 ± 1.23	98.20 ± 0.03

Table 9
The classification accuracy of selected features on SVM (%).

Datasets	original data	WKNFS	AAnfsCF	FC	LVPCM	IBMOA	FNGCE	WGMIE	MFS	Ours
PC	70.00 ± 5.09	69.72 ± 2.43	71.23 ± 4.17	77.34 ± 5.09	69.72 ± 4.39	75.87 ± 3.25	72.85 ± 3.34	74.44 ± 11.17	71.84 ± 2.67	**80.74 ± 8.65**
Sonar	80.74 ± 8.65	79.71 ± 9.09	73.75 ± 2.91	50.48 ± 7.18	79.71 ± 8.34	81.29 ± 7.47	79.76 ± 8.32	**84.10 ± 9.71**	59.10 ± 9.52	81.24 ± 7.95
Derm	73.75 ± 2.91	97.54 ± 7.33	96.99 ± 0.78	67.73 ± 3.21	97.53 ± 2.90	70.47 ± 3.62	93.16 ± 5.39	98.09 ± 1.74	35.25 ± 2.55	**98.63 ± 1.37**
BCW	94.71 ± 1.57	95.84 ± 2.43	71.00 ± 2.28	94.85 ± 1.83	96.84 ± 2.17	94.99 ± 1.83	96.43 ± 1.46	94.87 ± 1.28	92.70 ± 2.67	**96.99 ± 0.78**
SGCD	71.00 ± 2.28	72.77 ± 2.65	67.00 ± 3.12	70.60 ± 2.11	72.77 ± 3.41	68.50 ± 2.01	71.00 ± 2.14	**75.70 ± 1.49**	68.30 ± 3.58	73.50 ± 2.33
MPE	94.00 ± 3.21	98.99 ± 1.23	98.05 ± 1.01	96.84 ± 2.19	98.37 ± 0.87	97.63 ± 1.02	98.12 ± 0.54	97.34 ± 1.87	30.93 ± 2.46	**99.09 ± 0.12**
WIL	96.65 ± 1.47	98.25 ± 3.76	60.59 ± 3.72	96.25 ± 0.98	98.25 ± 2.19	97.35 ± 1.48	80.60 ± 3.10	98.20 ± 0.90	87.70 ± 1.93	**98.30 ± 0.68**
EOLBEHPC	54.52 ± 3.74	58.59 ± 1.20	**89.48 ± 1.50**	54.95 ± 3.60	58.39 ± 1.29	57.65 ± 3.15	59.78 ± 3.44	82.47 ± 1.41	29.28 ± 0.19	87.59 ± 3.72
PSDAS	49.93 ± 0.10	73.71 ± 1.32	76.36 ± 1.60	69.26 ± 1.00	73.71 ± 1.33	49.93 ± 0.10	49.93 ± 0.10	76.09 ± 1.59	50.75 ± 1.25	**78.58 ± 1.86**
SBD	93.76 ± 3.20	99.07 ± 0.88	98.83 ± 0.38	90.14 ± 0.24	99.07 ± 0.28	99.15 ± 0.21	98.20 ± 0.47	98.20 ± 0.51	89.32 ± 0.08	**99.26 ± 0.20**
EES	55.12 ± 0.03	55.38 ± 0.76	55.12 ± 0.03	48.25 ± 1.35	55.38 ± 1.38	57.87 ± 2.67	47.77 ± 1.23	59.66 ± 2.31	**88.83 ± 6.34**	80.13 ± 2.11
AGMPE	94.22 ± 2.01	95.67 ± 0.54	61.76 ± 1.76	96.25 ± 1.11	98.22 ± 2.35	67.88 ± 2.45	68.53 ± 3.56	66.64 ± 4.876	82.49 ± 1.87	**98.22 ± 0.01**

Table 10
The classification accuracy of selected features on NB (%).

Datasets	original data	WKNFS	AAnfsCF	FC	LVPCM	IBMOA	FNGCE	WGMIE	MFS	Ours
PC	43.33 ± 14.44	34.86 ± 13.83	70.07 ± 12.28	32.22 ± 12.62	34.86 ± 9.41	55.56 ± 14.05	56.67 ± 12.62	42.22 ± 10.89	70.00 ± 5.09	**75.56 ± 12.96**
Sonar	61.50 ± 11.19	67.50 ± 12.97	63.49 ± 7.88	53.88 ± 6.61	67.50 ± 7.65	61.71 ± 8.98	65.33 ± 7.24	62.43 ± 8.97	56.71 ± 9.08	**68.26 ± 12.28**
Derm	86.61 ± 5.91	87.12 ± 5.67	86.57 ± 5.28	83.09 ± 3.69	87.12 ± 2.34	87.45 ± 2.94	83.90 ± 3.04	85.33 ± 3.51	35.79 ± 2.75	**87.46 ± 4.17**
BCW	94.56 ± 1.79	95.34 ± 2.08	95.31 ± 3.12	94.85 ± 2.23	95.70 ± 2.40	95.13 ± 2.51	95.56 ± 1.86	95.56 ± 1.97	92.70 ± 2.67	**95.99 ± 1.84**
SGCD	72.30 ± 3.63	69.67 ± 2.45	70.77 ± 1.87	67.00 ± 3.66	69.67 ± 1.28	68.60 ± 4.94	69.20 ± 5.21	71.61 ± 3.87	69.60 ± 4.98	**72.70 ± 2.76**
MPE	94.26 ± 0.91	69.67 ± 1.33	97.33 ± 1.21	96.98 ± 2.16	99.81 ± 1.96	95.88 ± 1.47	97.31 ± 1.22	98.34 ± 1.01	29.35 ± 3.18	**99.00 ± 0.12**
WIL	96.90 ± 1.45	**99.81 ± 1.45**	98.30 ± 0.81	95.40 ± 1.62	98.20 ± 1.54	97.05 ± 1.25	79.90 ± 3.78	98.30 ± 0.81	86.75 ± 2.80	98.30 ± 0.81
EOLBEHPC	56.23 ± 3.68	**98.20 ± 3.56**	60.02 ± 2.10	58.83 ± 2.02	61.14 ± 2.58	53.20 ± 3.58	59.69 ± 3.08	59.17 ± 3.25	29.28 ± 0.19	97.02 ± 2.10
PSDAS	68.35 ± 1.41	61.14 ± 1.87	68.35 ± 1.41	69.87 ± 1.56	70.29 ± 1.48	67.41 ± 1.48	69.15 ± 2.07	67.86 ± 1.42	50.90 ± 1.42	**89.12 ± 1.31**
SBD	96.98 ± 0.78	70.29 ± 0.35	96.98 ± 0.78	88.15 ± 1.26	97.37 ± 0.77	97.41 ± 0.75	70.07 ± 1.43	96.98 ± 0.78	89.32 ± 0.08	**98.04 ± 0.51**
EES	45.17 ± 0.93	**97.37 ± 8.67**	45.42 ± 1.29	47.72 ± 3.29	54.02 ± 3.29	58.97 ± 3.31	56.26 ± 1.28	66.49 ± 3.87	88.86 ± 3.94	94.42 ± 1.29
AGMPE	91.31 ± 2.65	98.78 ± 2.34	94.33 ± 2.31	92.53 ± 3.46	98.31 ± 5.43	96.56 ± 2.43	93.77 ± 1.65	87.35 ± 2.75	92.56 ± 2.89	**98.22 ± 0.01**

Because the overall test was significant, we next applied the Wilcoxon test to assess pairwise differences between our algorithm and the other methods. The significance level was set at 0.05, and for each comparison the null hypothesis was "there is no significant difference between our algorithm and the compared method." Rejecting the null hypothesis indicates a significant difference.

Table 15 shows that, under the KNN classifier, our method does not differ significantly from WKNFS and LVPCM. With SVM, the difference from WGMIE is not significant, and with NB, the difference from our WKNN-FNRS feature selection is also insignificant. In contrast, significant differences appear when comparing with the other methods. Furthermore, our algorithm ranks first on all three classifiers and clearly outperforms the runner-up, confirming its superior classification performance.

5.5. Parameter analysis

This section studies how the neighborhood size K and threshold α affect classification performance. We first used Gradient Boosting as an exploratory proxy for parameter sensitivity because tree-based ensembles are sensitive to feature perturbations and can reveal complex non-linear interactions. To ensure the relevance of these findings for our main classifiers (KNN, SVM, NB), we also conducted targeted sensitivity experiments for each classifier on representative datasets, using the same nested CV protocol and parameter ranges. The observed trends were consistent: increasing K initially reduces sensitivity to local noise but may obscure fine-grained distinctions, while varying α trades off tolerance against discrimination. Detailed results for the parameter grid are shown in Fig. 9.

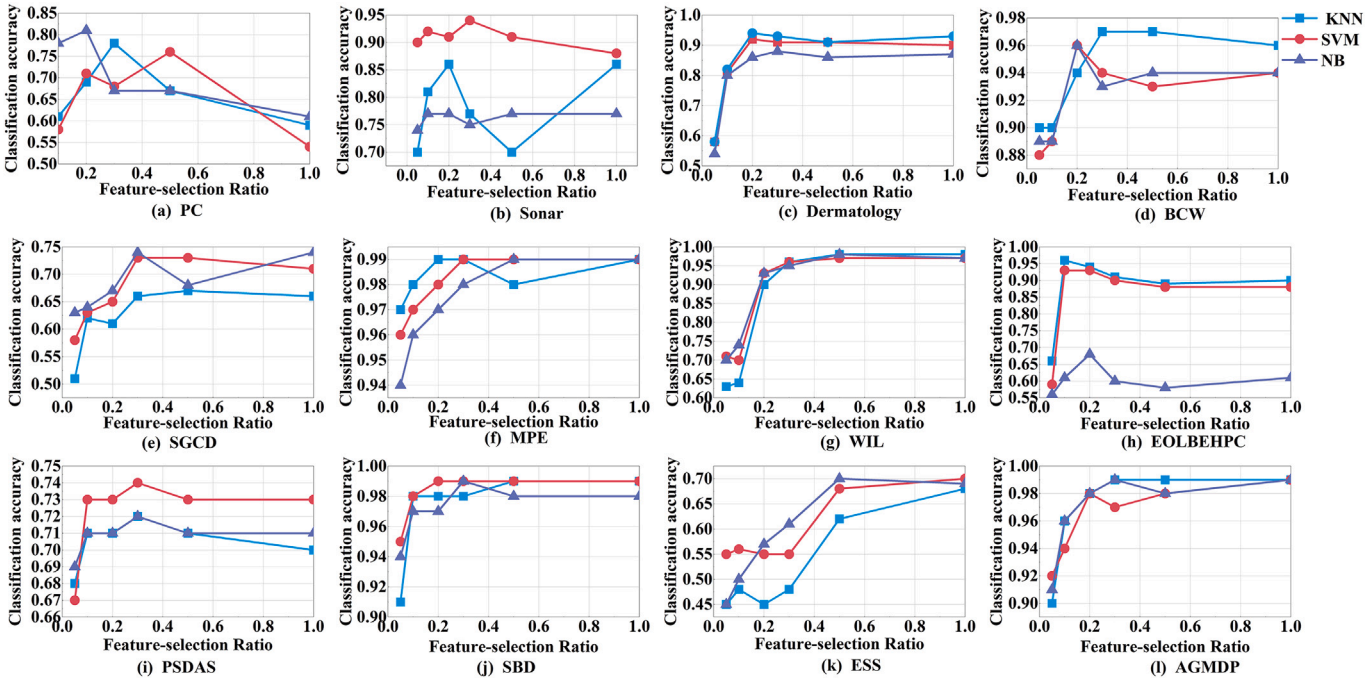

Fig. 8. Classification accuracy of KNN, SVM, and NB classifiers under different feature selection ratio.

Table 11
Ablation results (Accuracy) on the 12 datasets using **KNN**.

Dataset	Full	No weight	No fuzzy	Single scale
PC	0.889	0.552	0.778	0.679
Sonar	0.810	0.571	0.386	0.707
Derm	0.973	0.917	0.622	0.917
BCW	0.929	0.906	0.913	0.894
SGCD	0.728	0.696	0.519	0.574
MPE	0.998	0.950	0.936	0.948
WIL	0.975	0.935	0.931	0.910
EOLBEHPC	0.948	0.853	0.848	0.925
PSDAS	0.771	0.736	0.493	0.571
SBD	0.998	0.992	0.983	0.979
EES	0.832	0.643	0.578	0.459
AGMPD	0.982	0.963	0.970	0.964
Average	**0.903**	0.810	0.746	0.794

Table 12
Ablation results (Accuracy) on the 12 datasets using **SVM**.

Dataset	Full	No weight	No Fuzzy	Single Scale
PC	0.893	0.667	0.778	0.751
Sonar	0.910	0.762	0.478	0.721
Derm	0.976	0.945	0.806	0.861
BCW	0.985	0.942	0.926	0.943
SGCD	0.739	0.671	0.632	0.701
MPE	0.999	0.973	0.970	0.921
WIL	0.975	0.964	0.925	0.924
EOLBEHPC	0.929	0.876	0.783	0.895
PSDAS	0.736	0.667	0.493	0.622
SBD	0.994	0.973	0.962	0.955
EES	0.751	0.634	0.562	0.535
AGMPD	0.982	0.594	0.496	0.449
Average	**0.906**	0.803	0.735	0.773

Table 13
Ablation results (Accuracy) on the 12 datasets using **NB**.

Dataset	Full	No weight	No Fuzzy	Single Scale
PC	0.779	0.502	0.667	0.643
Sonar	0.759	0.720	0.515	0.651
Derm	0.861	0.735	0.694	0.689
BCW	0.985	0.942	0.863	0.936
SGCD	0.749	0.559	0.579	0.681
MPE	0.985	0.909	0.738	0.768
WIL	0.988	0.975	0.929	0.909
EOLBEHPC	0.670	0.559	0.592	0.611
PSDAS	0.726	0.706	0.493	0.619
SBD	0.998	0.982	0.910	0.971
EES	0.832	0.456	0.551	0.454
AGMPD	0.982	0.945	0.921	0.945
Average	**0.859**	0.749	0.704	0.740

Table 14
Result of the Iman-Davenport test.

Algorithm	Friedman value	Iman–Davenport F	P value
KNN	40.967	8.188	1.4×10^{-7}
SVM	38.511	7.368	1.8×10^{-7}
NB	42.906	8.889	7.6×10^{-9}

Table 15
P value of the Wilcoxon test.

Algorithm	WKNFS	AAnfsCF	FC	LVPCM	IBMOA	FNGCE	WGMIE	MFS
KNN	0.064	< 0.05	< 0.05	0.77	< 0.05	< 0.05	< 0.05	< 0.05
SVM	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.052	< 0.05
NB	0.11	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05

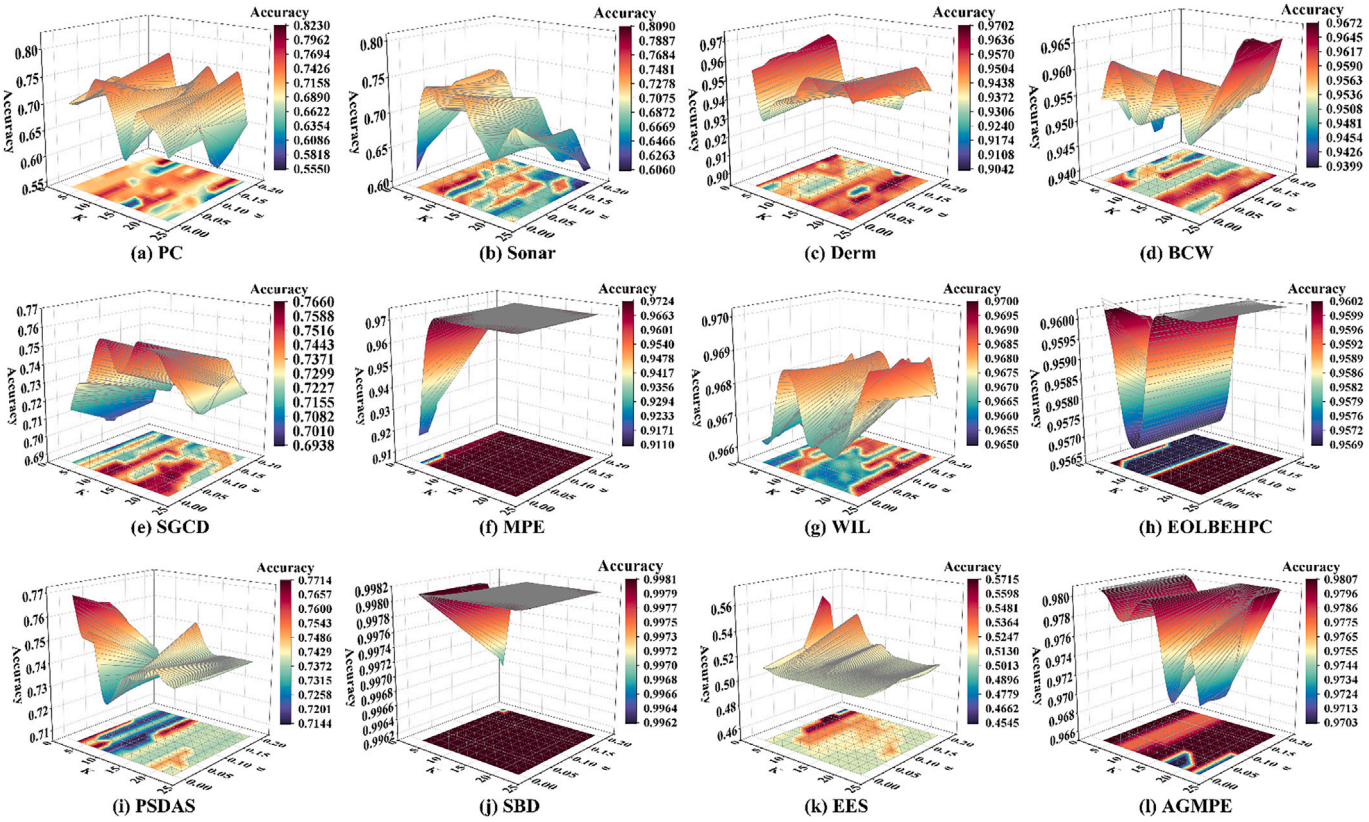

Fig. 9. Classification accuracy under different parameter combinations (α, K).

6. Conclusion

We present WKNN-FNRS feature, a weighted KNN fuzzy feature selection framework for generalized multi-scale decision systems that unifies three innovations to improve robustness, scalability, and efficiency: a weighted fuzzy neighborhood rough set that down-weights outliers based on intra-class standard deviation; a scale-based tree reduction that converts multi-scale tables into an information-preserving single-scale representation; and a dynamic feature selection (DFS) mechanism that incrementally updates only affected similarity and dependency measures to avoid full recomputation.

On twelve benchmark datasets, our method outperforms coarse and fine-scale baselines, improves average accuracy by 8.9% and 4.3%, remains stable with up to 50% injected noise, ranks first across KNN, SVM, and NB in statistical tests, and achieves substantial runtime savings via DFS. Overall, WKNN-FNRS offers a practical balance of discriminative power, noise resilience, and computational efficiency for dynamic multi-scale environments. Future work will integrate multi-scale and multi-label information and explore deep learning-based enhancements.

CRediT authorship contribution statement

Xiaoyan Zhang: Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Jingwen Wang:** Writing – review & editing, Writing – original draft, Visualization, Software, Investigation, Formal analysis, Data curation.

Declaration of competing interest

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

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 12371465) and the Chongqing Natural Science Foundation of China (Grant No. CSTB2023NSCQ-MSX1063).

Data availability

No data was used for the research described in the article.

References

[1] C. Wang, M. Shao, Q. He, Y. Qian, Y. Qi, Feature subset selection based on fuzzy neighborhood rough sets, Knowl.-Based Syst. 111 (2016) 173–179.
[2] L. Qiu, X. Wang, Y. Qu, K. Zhang, F. Gao, B. Yi, K. Li, A robust pseudo fuzzy rough feature selection using linear reconstruction measure, IEEE Trans. Fuzzy Syst. 32 (10) (2024) 5687–5701.
[3] W. Xu, X. Zhao, Game-theoretic dynamic ensemble for oncogene diagnosis: integrating neighborhood and precision rough sets, Appl. Intell. 55 (11) (2025) 21.
[4] Q. Cui, K. El-Arroudi, Y. Weng, A feature selection method for high impedance fault detection, IEEE Trans. Power Deliv. 34 (3) (2019) 1203–1215.
[5] M. Li, X. Zhang, J. Shang, Y. Ma, General quasi overlap functions and fuzzy neighborhood systems-based fuzzy rough sets with their applications, IEEE Trans. Knowl. Data Eng. 36 (12) (2024) 8349–8361.
[6] P. Jain, A.K. Tiwari, T. Som, A fitting model based intuitionistic fuzzy rough feature selection, Eng. Appl. Artif. Intell. 89 (2020) 103421.
[7] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[8] R. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Trans. Fuzzy Syst. 17 (4) (2009) 824–838.
[9] C. Wang, C. Wang, Y. Qian, Q. Leng, Feature selection based on weighted fuzzy rough sets, IEEE Trans. Fuzzy Syst. 32 (7) (2024) 4027–4037.
[10] T. Yin, H. Chen, Z. Yuan, T. Li, K. Liu, Noise-resistant multilabel fuzzy neighborhood rough sets for feature subset selection, Information Sciences 621 (2023) 200–226.
[11] B. Sang, W. Xu, H. Chen, T. Li, Active antinoise fuzzy dominance rough feature selection using adaptive k-nearest neighbors, IEEE Trans. Fuzzy Syst. 31 (11) (2023) 3944–3958.
[12] W. Xu, Y. Li, Multi-label feature selection for imbalanced data via knn-based multi-label rough set theory, Information Sciences 715 (2025) 122220.

[13] W. Xu, Z. Tian, Feature selection and information fusion based on preference ranking organization method in interval-valued multi-source decision-making information systems, Information Sciences 700 (2025) 121860.
[14] Z. Feng, X. Zhang, Supervised incremental feature selection using regularization vector for dynamic multi-scale interval valued datasets, Pattern Recognition 170 (2026) 111985.
[15] X. Zhang, W. Zhao, Uncertainty measures and feature selection based on composite entropy for generalized multigranulation fuzzy neighborhood rough set, Fuzzy Sets and Systems 486 (2024) 108971.
[16] X. Zhang, X. Shen, Graph-driven feature selection via granular-rectangular neighborhood rough sets for interval-valued data sets, Appl. Soft Comput. 170 (2025) 112716.
[17] M. Sarkar, Fuzzy-rough nearest neighbor algorithms in classification, Fuzzy Sets Syst. 158 (19) (2007) 2134–2152.
[18] X. Yang, H. Chen, T. Li, C. Luo, A noise-aware fuzzy rough set approach for feature selection, Knowl.-Based Syst. 250 (2022) 109092.
[19] B. Sang, L. Yang, H. Chen, W. Xu, X. Zhang, Fuzzy rough feature selection using a robust non-linear vague quantifier for ordinal classification, Expert Syst. Appl. 230 (2023) 120480.
[20] A. Theerens, C. Cornelis, Fuzzy rough sets based on fuzzy quantification, Fuzzy Sets and Systems 473 (2023) 108704.
[21] A. Theerens, O.U. Lenz, C. Cornelis, Choquet-based fuzzy rough sets, Int. J. Approx. Reason. 146 (2022) 62–78.
[22] L. Sun, T. Yin, W. Ding, Y. Qian, J. Xu, Feature selection with missing labels using multilabel fuzzy neighborhood rough sets and maximum relevance minimum redundancy, IEEE Trans. Fuzzy Syst. 30 (5) (2022) 1197–1211.
[23] J. Wan, H. Chen, T. Li, X. Yang, B. Sang, Dynamic interaction feature selection based on fuzzy rough set, Information Sciences 581 (2021) 891–911.
[24] X. Zhang, C. Mei, J. Li, Y. Yang, T. Qian, Instance and feature selection using fuzzy rough sets: a bi-selection approach for data reduction, IEEE Trans. Fuzzy Syst. 31 (6) (2023) 1981–1994.
[25] W. Wu, Y. Leung, Optimal scale selection for multi-scale decision tables, Int. J. Approx. Reason. 54 (8) (2013) 1107–1129.
[26] H. Zhao, P. Wang, Q. Hu, P. Zhu, Fuzzy rough set based feature selection for large-scale hierarchical classification, IEEE Trans. Fuzzy Syst. 27 (10) (2019) 1891–1903.
[27] W. Qian, F. Xu, J. Huang, J. Qian, A novel granular ball computing-based fuzzy rough set for feature selection in label distribution learning, Knowl.-Based Syst. 278 (2023) 110898.
[28] X. Zhang, Z. Feng, Feature selection based on contradictory state sequence for multi-scale interval valued decision table, Inf. Sci. 677 (2024) 120926.
[29] Z. Huang, J. Li, W. Dai, R. Lin, Generalized multi-scale decision tables with multi-scale decision attributes, Int. J. Approx. Reason. 115 (2019) 194–208.
[30] X. Zhang, Y. Huang, Optimal scale selection and knowledge discovery in generalized multi-scale decision tables, Int. J. Approx. Reason. 161 (2023) 108983.
[31] C. Wang, E. Chen, M. Ren, X. Yu, Y. Lin, S. Li, Multi-label feature selection based on fuzzy neighborhood mutual discrimination index, in: International Conference on Information Technology in Medicine and Education (ITME), 2022, pp. 608–612.
[32] S. An, M. Zhang, C. Wang, W. Ding, Robust fuzzy rough approximations with KNN granules for semi-supervised feature selection, Fuzzy Sets Syst. 461 (2023) 108476.
[33] N. Wang, E. Zhao, A new method for feature selection based on weighted k-nearest neighborhood rough set, Expert Syst. Appl. 238 (2024) 122324.
[34] L. Zhang, G. Lin, L. Wei, Y. Kou, Feature subset selection for multi-scale neighborhood decision information system via mutual information, Artif. Intell. Rev. 57 (1) (2024).
[35] W. Xu, W. Ye, Incremental feature selection: parallel approach with local neighborhood rough sets and composite entropy, Pattern Recognition 159 (2025) 111141.
[36] K. Yuan, D. Miao, H. Zhang, W. Pedrycz, An efficient and robust feature selection approach based on zentropy measure and neighborhood-aware model, IEEE Trans. Neural Netw. Learn. Syst. 36 (9) (2025) 16351–16365.
[37] W. Huang, Y. She, X. He, W. Ding, Fuzzy rough sets-based incremental feature selection for hierarchical classification, IEEE Trans. Fuzzy Syst. 31 (10) (2023) 3721–3733.
[38] L. Dong, R. Wang, D. Chen, Incremental feature selection with fuzzy rough sets for dynamic data sets, Fuzzy Sets Syst. 467 (2023) 108503.
[39] A. Rangel-Díaz-de-la Vega, Y. Villuendas-Rey, C. Yanez-Marquez, O. Camacho-Nieto, I. Lopez-Yanez, Impact of imbalanced datasets preprocessing in the performance of associative classifiers, Appl. Sci. 10 (8) (2020) 2779.
[40] K. Yuan, W. Xu, D. Miao, A local rough set method for feature selection by variable precision composite measure, Appl. Soft Comput. 155 (2024) 111450.
[41] R.M. Hussien, A.A. Abohany, A.A. Abd El-Mageed, K.M. Hosny, Improved binary meerkat optimization algorithm for efficient feature selection of supervised learning classification, Knowl.-Based Syst. 292 (2024) 111616.
[42] W. Xu, Q. Bu, Feature selection using generalized multi-granulation dominance neighborhood rough set based on weight partition, IEEE Trans. Emerg. Top. Comput. Intell. 9 (1) (2025) 213–227.
[43] X. Zhang, J. Lin, Scalable data fusion via a scale-based hierarchical framework: adapting to multi-source and multi-scale scenarios, Inf. Fusion. 114 (2025) 102694.
[44] R.L. Iman, J.M. Davenport, Approximations of the critical region of the fbietkan statistic, Commun. Stat.-Theory Methods 9 (6) (1980) 571–595.