Fusing Monotonic Decision Tree Based on Related Family

Tian Yang, Fansong Yan, Fengcai Qiao, Jieting Wang, Yuhua Qian*,

Abstract—Monotonic classification is a special ordinal classification task that involves monotonicity constraints between features and the decision. Monotonic feature selection can reduce dimensionality while preserving the monotonicity constraints, ultimately improving the efficiency and performance of monotonic classifiers. However, existing feature selection algorithms cannot handle large-scale monotonic data sets due to their lack of consideration for monotonic constraints or their high computational complexities. To address these issues, building on our team's previous research, we define the monotonic related family method with lower time complexity to select informative features and obtain multi-reducts carrying complementary information from multi-view for raw feature space. Using bi-directional rank mutual information, we build two trees for each feature subset and fuse all trees using the corresponding decision support level (BFMDT). Compared with six representative algorithms for monotonic feature selection, BFMDT's average classification accuracy increased by 4.06% (FFREMT), 6.77% (FCMT), 5.61% (FPRS_up), 6.05% (FPRS_down), 5.86%(FPRS_global), 4.41% (Bagging), 7.65% (REMT) and 21.89% (FMKNN), the average execution time compared to tree-based algorithms decreased by 83.41% (FFREMT), 96.96% (FCMT), 75.64% (FPRS_up), 59.43% (FPRS_down), 84.65%(FPRS_global), 81.50% (Bagging) and 63.41% (REMT), while most of comparing algorithms were unable to complete computation on six high-dimensional datasets.

Index Terms—Rough set, Granular computing, Related family, Monotonic classification, Decision tree, Feature selection.

1 Introduction

Classification is a vital research topic in machine learning and data mining, aiming to train classifiers using labeled samples to predict the labels of unlabeled samples. Classification tasks can be divided into nominal classification and ordinal classification based on monotonicity constraints. Ordinal classification tasks exhibit ordinal relationships among different decision classes [1]–[3]. Monotonic classification tasks refer to a special case of ordinal classification, where monotonicity constraints exist between features and decisions, i.e., samples with higher feature values cannot be assigned to lower decision classes. Due to its universality in the real world, monotonic classification has garnered increasing interest in the fields of data mining [4], knowledge discovery [5], pattern recognition [6], intelligent decision-making [7], [8] and so on.

Classical classifiers in machine learning, such as Neural Networks, k-Nearest Neighbors (kNN), and Decision Trees, are not suitable for solving monotonic classification problems as they do not consider monotonicity constraints. To achieve monotonicity, Monotonic Neural Networks [9] impose constraints on the network parameters, Totally and Partially Monotone Neural Networks are explored in [10]. Monotonic Nearest Neighbor algorithms [11] aim to achieve monotonicity of the training data by relabeling the dataset and modifying the neighbor rules to ensure that the predicted results satisfy monotonicity constraints. Monotonic Decision Trees [6] introduce the notion of rank entropy as a criterion for selecting split points in decision tree construction, and modify the prediction mechanism of leaf nodes to satisfy monotonicity constraints. Compared to the first two models, the monotonic decision tree model is rule-based and interpretable rather than probabilistic or instance, making it better suited for handling structured data.

Ensemble learning, such as Bagging [12] and Boosting [7], [13], improving a classifier's robustness and generalization by combining the evaluation of prediction results from multiple classifiers, is also applied to monotonic classification [14]. Especially, due to providing multiple complementary feature subsets, some feature selection algorithms are widely utilized in ensemble learning [15], [16].

Rough Set Theory [17], introduced by Pawlak, has proven to be an effective mathematical tool for monotonic classification [4], [6], [14], [16], [18], [19] and ensemble learning [16], [20]. Dominance Rough Set Theory, which substitutes equivalence relationships with preference relationships, is widely employed for feature selection in ordinal decision tables, including normal ordinal data [21]–[25], interval-valued ordinal data [26], [27], and dynamic ordinal data [28], [29] et al.. The discernibility matrix [30] and the significance measure [3] are two primary effective methods used for feature selection based on rough sets. They have been combined with monotonic classifiers to improve monotonic classification problems [31], [32].

Our research team has long been dedicated to the ex-

- *Tian Yang and Fansong Yan are with Hunan Provincial Key laboratory of Intelligent Computing and Language Information Processing (NO. 2018TP1018) and Institute of Interdisciplinary Studies, Hunan Normal University, Changsha 410081, China. (e-mail: math_yangtian@126.com; fansong_yan@icloud.com).*
- *Fengcai Qiao is with College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410073, China. (e-mail: fcqiao@nudt.edu.cn).*
- *Jieting Wang and Yuhua Qian are with Institute of Big Data Science and Industry, Shanxi university, Taiyuan 030006, China. (e-mail: jietingwang@email.sxu.edu.cn; jinchengqyh@126.com).*

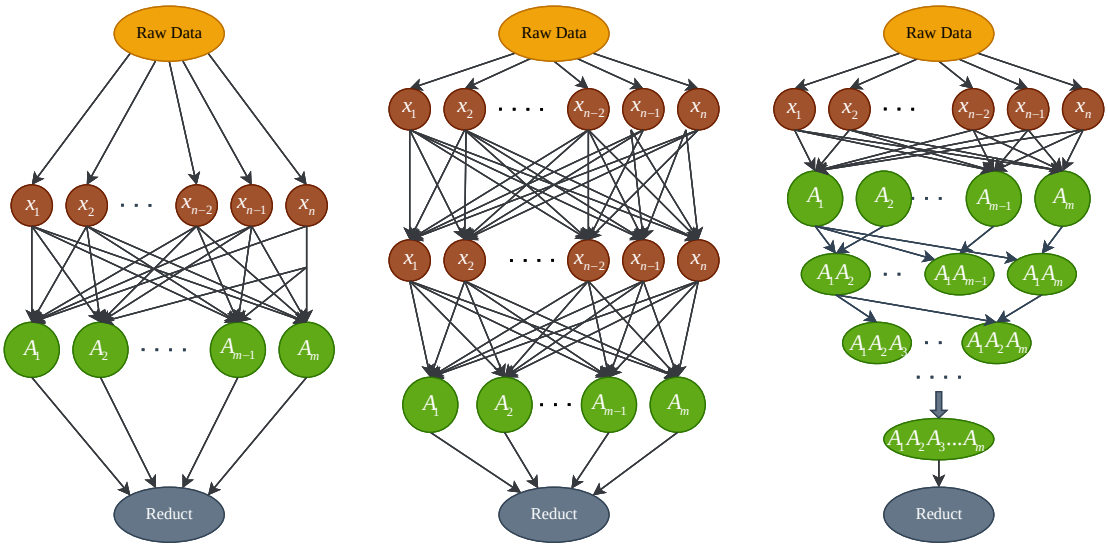


Fig. 1: The computation process of related family (left), discernibility matrix (center) and significance measure (right).

ploration of fusing monotonic decision tree, employing feature selection methods based on both the significance measure [14] and the discernibility matrix [16], [19]. Among these researches, Qian et al. developed an method called FREMT [14] for monotonic feature selection and decision tree fusion based on feature significance in dominance rough sets, aiming to further enhance the generalization ability of ordinal decision trees; Xu et al. [19] propose FCMT based on discernibility matrix in dominance rough set; Wang et al. [16] propose FFREMT, which employs a fuzzy discernibility matrix as a feature selection method and predicts through voting across all trees. Besides our works, Hu et al. [33] introduced three fuzzy significance measures using a forward heuristic algorithm FPRS, which can select monotonic features. Although feature selection algorithms based on feature significance and discernibility matrix are theoretically well-founded, their drawbacks are also evident. The complexities of significance measure-based methods [14], [33] and discernibility matrix-based methods [16], [19] are both quadratic about feature scale and sample scale, respectively. As a result, they struggle to handle high-dimensional data.

To achieve more efficient feature selection for large-scale data processing, the first author et al. [34]–[36] propose the related family method for efficient feature selection. Because it evaluate the significance of every single feature and avoid repeat granulation, the complexities for both time and space can be reduced to linear. The computation process of related family is compared with discernibility matrix and significance measure in Fig. 1. The algorithm based on the discernibility matrix performs pairwise comparisons of sample on each feature, whereas the feature significance-based method involves complex calculations at the feature level. In contrast, the related family only requires a single traversal of the sample set for each feature, resulting exponentially boosting computation efficiency and scalability of feature selection by even thousands of times [37]. However, the original related family approach is unable to exploit the ordinal information of the data.

Granulation is one of the fundamental processes in rough sets, where the problem space is partitioned into information granules forming granular layers, enabling concise and accurate knowledge representation under given granulation criteria to improve algorithm interpretability and knowledge inference efficiency, thereby satisfying the requirements of large-scale data processing. The granulation approaches used in the discernibility matrix [16], [19], significance measure [38]–[40] and related family [35], [37] and others [37], [41], [42] have achieved promising results in various aspects, but either exhibit high time or space complexity, or do not consider preference relations in monotonic datasets, thus failing to efficiently process ordinal data sets. In order to develop an accurate and efficient approach for ordinal data processing, we introduce innovations in monotonic classification from four perspectives:

1) Monotonic Consistent granulation: Monotonic Partition based on single feature is presented to ensure that all samples within a granule adhere to strict monotonicity constraints. As a result, the time complexity about sample scale is reduced as $O(nlogn)$.
2) Accurate granule evaluation: The fitting degree is proposed for assessing the quality of all monotonic partitions, which offers a more precise evaluation method compared to dependency degree.
3) Efficient feature evaluation mode: Monotonic Related Family, a highly efficient feature evaluation framework based on individual features, is constructed for reducing the time complexity with respect to the feature scale as $O(m)$.
4) Fusing monotonic decision tree: Monotonic Related Family computes multi-reducts for each ordinal decision table, which supply complementary information from various perspectives. Each reduct induces two monotonic decision trees (superiority and inferiority), then all trees generated by different reducts are fused to construct an accurate and robust classifier, called BFMDT.

Five representative algorithms of monotonic decision tree, including FFREMT [16], FCMT [19], FPRS (up, down,

global) [33], Bagging [12] and REMT [6], are compared with BFMDT on eighteen datasets. Experiments show that the average CA (classification accuracy) increased by: 4.06% (FFREMT), 6.77% (FCMT), 5.61% (FPRS_up), 6.05% (FPRS_down), 5.86%(FPRS_global), 4.41% (Bagging), 7.65% (REMT) and 21.89% (FMKNN); the average MAE (mean absolute error) decreased by: 6.35% (FFREMT), 6.89% (FCMT), 6.58% (FPRS_up), 6.96% (FPRS_down), 6.78%(FPRS_global), 16.14% (Bagging), 16.54% (REMT) and 56.19% (FMKNN); the average execution time of the algorithm reduced by: 83.41% (FFREMT), 96.96% (FCMT), 75.64% (FPRS_up), 59.43% (FPRS_down), 84.65%(FPRS_global), 81.50% (Bagging) and 63.41% (REMT). Additionally, algorithms with feature selection could not complete experiments on six high-dimensional datasets.

2 PRELIMINARIES

This section presents a review of the concepts associated with feature selection in ordinal classification, specifically within the monotonic classification problems. Additionally, the fundamental concepts of the related family are also discussed.

2.1 Monotonicity Constraints

Let $(U, A \cup \{D\})$ be an ordinal decision table, where $U = \{x_1, \ldots, x_n\}$ is a set of samples, $A = \{a_1, \ldots, a_m\}$ is a set of features to describe the samples, and D is the decision class with values $\{d_1, d_2, \ldots, d_k\}$. Let $v(x, a)$ be the value of x with respect to the feature $a \in A$. There are four relations between two samples in a decision table: $\succcurlyeq_a$, $\succcurlyeq_D$, $\preccurlyeq_a$ and $\preccurlyeq_D$, which signify the relation of superiority and inferiority with respect to a or D, respectively.

Definition 1. [30], [43] Given a set of samples U, $\forall x \in U$ and $B \subseteq A$, where $A = \{a_1, \ldots, a_m\}$. Let $v(x, a_k)$ be the feature value of sample x under a_k, $k = 1, 2, ..., m$. The ordinal relations between samples in terms of feature a_k or D is denoted by $\preccurlyeq$ and $\succcurlyeq$. Thus, the preference relations on U is defined

$$x_i \preccurlyeq_B x_j \Leftrightarrow v(x_i, a_k) \leqslant v(x_j, a_k)\ for\ \forall a_k \in B, \quad (1)$$

$$x_i \succcurlyeq_B x_j \Leftrightarrow v(x_i, a_k) \geqslant v(x_j, a_k)\ for\ \forall a_k \in B. \quad (2)$$

The dominance class $[x_i]$ and inferiority class $[x_i]$are defined as:

$$[x_i]_B^{\preccurlyeq} = \{x_j | x_i \preccurlyeq_B x_j\},\ [x_i]_D^{\preccurlyeq} = \{x_j | x_i \preccurlyeq_D x_j\}, \quad (3)$$

$$[x_i]_B^{\succcurlyeq} = \{x_j | x_i \succcurlyeq_B x_j\},\ [x_i]_D^{\succcurlyeq} = \{x_j | x_i \succcurlyeq_D x_j\}. \quad (4)$$

The monotonicity constraints are defined as:

$$x_i \preccurlyeq_B x_j \Rightarrow v(x_i, D) \leqslant v(x_j, D), \quad (5)$$

$$x_i \succcurlyeq_B x_j \Rightarrow v(x_i, D) \geqslant v(x_j, D). \quad (6)$$

The inferior class characterizes the "not superior" relationship among samples, while the dominant class represents the "not inferior" relationship. This also holds true for the decision set D. Based on these observations, monotonicity constraints are defined.

Based on the monotonicity constraints, the lower approximation and upper approximation of the set which dominates d_i are defined as follows:

Definition 2. [19] Let $d_i^{\succcurlyeq}$ be a sample set whose class is no worse than class d_i. The lower approximation and upper approximation are:

$$\underline{R_B^{\succcurlyeq}} d_i^{\succcurlyeq} = \{x \in U | [x]_B^{\succcurlyeq} \subseteq d_i^{\succcurlyeq}\}, \quad (7)$$

$$\overline{R_B^{\succcurlyeq}} d_i^{\succcurlyeq} = \{x \in U | [x]_B^{\preccurlyeq} \cap d_i^{\succcurlyeq} \neq \emptyset\}. \quad (8)$$

2.2 Rank Entropy and REMT

The C4.5 algorithm, a representative decision tree method, leverages Mutual Information as the metric for the selection of split points. Nonetheless, this approach overlooks the monotonicity of data, potentially leading to the omission of ordinal information throughout the computational process. To overcome this limitation, Rank Entropy and Rank Mutual Information are introduced to assess the monotone consistency of ordinal data.

Definition 3. [6], [18] Let $(U, A \cup \{D\})$ be an ordinal decision table, U is a set of samples described by feature set A, $B \subseteq A$. The ascending and descending rank entropy with respect to B are defined

$$RE_B^{\preccurlyeq}(U) = -\frac{1}{|U|} \sum_{i=1}^{n} log \frac{|[x_i]_B^{\preccurlyeq}|}{|U|}, \quad (9)$$

$$RE_B^{\succcurlyeq}(U) = -\frac{1}{|U|} \sum_{i=1}^{n} log \frac{|[x_i]_B^{\succcurlyeq}|}{|U|}. \quad (10)$$

The ascending and descending rank conditional information of the set U regarding B and C are defined as:

$$RE_{B|C}^{\preccurlyeq}(U) = -\frac{1}{|U|} \sum_{i=1}^{n} log \frac{|[x_i]_B^{\preccurlyeq} \cap [x_i]_C^{\preccurlyeq}|}{|[x_i]_C^{\preccurlyeq}|}, \quad (11)$$

$$RE_{B|C}^{\succcurlyeq}(U) = -\frac{1}{|U|} \sum_{i=1}^{n} log \frac{|[x_i]_B^{\succcurlyeq} \cap [x_i]_C^{\succcurlyeq}|}{|[x_i]_C^{\succcurlyeq}|}. \quad (12)$$

The ascending and descending rank mutual information (ARMI and DRMI) regarding B and C are defined as:

$$RMI^{\preccurlyeq}(B, C) = -\frac{1}{|U|} \sum_{i=1}^{n} log \frac{|[x_i]_B^{\preccurlyeq}| \times |[x_i]_C^{\preccurlyeq}|}{|U| \times |[x_i]_B^{\preccurlyeq} \cap [x_i]_C^{\preccurlyeq}|}, \quad (13)$$

$$RMI^{\succcurlyeq}(B, C) = -\frac{1}{|U|} \sum_{i=1}^{n} log \frac{|[x_i]_B^{\succcurlyeq}| \times |[x_i]_C^{\succcurlyeq}|}{|U| \times |[x_i]_B^{\succcurlyeq} \cap [x_i]_C^{\succcurlyeq}|}. \quad (14)$$

The Rank Mutual Information (RMI) serves as a measure to evaluate the monotone relationship between features B and C. Consequently, both rank entropy and RMI are frequently utilized for determining the split points in monotonic decision trees.

In the realm of monotonic classification, the Rank Entropy-based Monotonic Tree (REMT) algorithm [6] has

gained significant popularity due to its reliance on Ascending Rank Mutual Information (ARMI) for generating monotonic consistent decision rules.

Given an feature set $\{a_1, a_2, \ldots, a_m\}$ and the sample set $U_i = \{x_1, x_2, \ldots, x_n\}$ for the i-th sub dataset, let $ARMI^{\leqslant}(a_j, c, D)$ be the ARMI induced by a conditional feature a_j and the decision feature D with a split point c, the $ARMI^{\leqslant}(a_j, c, D)$ are computed as follow

For $x \in U_i$, let

$$v'(x, a_j) = \begin{cases} 1, & v(x, a_j) \leqslant c; \\ 2, & v(x, a_j) > c. \end{cases} \tag{15}$$

$$ARMI^{\leqslant}(a_j, c, D) = -\frac{1}{|U_i|} \sum_{x \in U_i} log \frac{|[x]'^{\leqslant}_{a_j}| \times |[x]^{\leqslant}_D|}{|U_i| \times |[x]'^{\leqslant}_{a_j} \cap [x]^{\leqslant}_D|}, \tag{16}$$

where $[x]'^{\leqslant}_{a_j}$ is formed by the new values $v'(x, a_j)$.

$$\begin{aligned} &ARMI^{\leqslant}(a^*, c^*, D) = \\ &\qquad \max\{ARMI^{\leqslant}(a_j, c, D) | c \in C_j, a_j \in A\}, \end{aligned} \tag{17}$$

where C_j is the set of all candidate values for feature a_j, c^* is a number to split the value domain of a^* such that the rank mutual information between a^* and D yields the largest value. Then select a feature that archives the maximum ARMI among all features as the split feature a^* and its optimal split point c^*. Since REMT is a binary tree, finding one split point is sufficient. The split point c^* of feature a^* divides the sample set U_i into two subsets: $U_{i1} = \{x \in U_i | v(a^*, x) \leqslant c^*\}$ and $U_{i2} = \{x \in U_i | v(a^*, x) > c^*\}$.

In order to mitigate the problem of overfitting, a threshold value of δ is applied in the algorithm. The rank mutual information (ARMI) computed on U_i for split point a^* and c^* signifies the degree of monotone consistency within the sample set. Once the monotone consistency is lower than a given threshold, the node ceases to split and is designated as a leaf node. In cases where the decision in the sample set U_i or the feature value under a_j is singular, the ARMI is equal to 0, and the sample set is likewise identified as a leaf node.

2.3 Related Family

Related family [34], dependency degree [3] and the discernibility matrix [19] can be all utilized for feature selection in decision tables. Due to much greater computational efficiency, related family is used to processing large scale data.

Definition 4. [34] Let (U, Δ, D) be a covering decision system, $\Delta = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$ be a family of coverings of $U = \{x_1, x_2, ..., x_n\}$, and for any $x_i \in U$, $r(x_i) = \{\mathcal{C} \in \Delta | \exists C_k \in \cup\Delta$ and $\exists X \in U/D$ s.t. $x_i \in C_k \in \mathcal{C}$ and $C_k \subseteq X\}$. Then $R(U, \Delta, D) = \{r(x_i) | x_i \in U\}$ is called the related family of (U, Δ, D).

Based on the related family, all reducts can be computed using boolean operations, providing multiple perspectives for classification tasks.

Definition 5. [34] Let (U, Δ, D) be a covering decision system, $\Delta = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$ be a family of coverings of $U = \{x_1, x_2, ..., x_n\}$, $R(U, \Delta, D) = \{r(x_i) | x_i \in U\}$. The related function $f(U, \Delta, D)$ is a boolean function of m boolean variables $\overline{\mathcal{C}_1}, \overline{\mathcal{C}_2}, \ldots, \overline{\mathcal{C}_m}$ corresponding to the coverings $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$, respectively, which is defined as

$$f(U, \Delta, D)(\overline{\mathcal{C}_1}, \overline{\mathcal{C}_2}, \ldots, \overline{\mathcal{C}_m}) = \wedge\{\vee(r(x_i) | x_i \in U)\} \tag{18}$$

Definition 6. [34] Let (U, Δ, D) be a consistent covering decision system, where Δ be a family of coverings on U, $R(U, \Delta, D)$ be the related family of (U, Δ, D), and $f(U, \Delta, D)$ be the related function. If $g(U, \Delta, D) = \vee_{k=1}(\wedge\Delta_k)(\Delta_k \in \Delta)$ is the reduced disjunctive form obtained from $f(U, \Delta, D)$ via the laws of multiplication and absorption. That is, for any $\Delta_k \subseteq \Delta, k = 1, 2, \ldots, l$, there is no repeated element in Δ_k. Then $RED(\Delta, D) = \{\Delta_1, \Delta_2, \ldots, \Delta_l\}$.

3 Feature Selection based on Monotonic Related Family

The discernibility matrix and the significance measure (such as dependency degree) are indeed effective feature selection methods. Nevertheless, their high computational complexity, concerning either sample size or feature scale, can result in time-consuming processes when handling large sample datasets and may even prove impossible to complete with limited resources. In this section, to reduce the computation complexity of feature selection, we propose a novel feature selection method, named monotonic related family, from the perspective of feature values under every single feature.

3.1 Monotonic Partition

Different feature selection methods may correspond to various granulation approaches. For monotonic related family, it is necessary to adopt a new granulation technique. We introduce the definition of Maximal Monotonic Interval (MMI), Monotonic Partition, and the corresponding granulation method first.

Definition 7. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $\mathcal{G}_t(a_j) \subseteq U$ and $a_j \in A$. If $\mathcal{G}_t(a_j)$ satisfies:

1) For $\forall x_i, x_k \in \mathcal{G}_t(a_j)$, if $x_i \leqslant_{a_j} x_k$ then $x_i \leqslant_D x_k$; if $x_i \geqslant_{a_j} x_k$ then $x_i \geqslant_D x_k$.
2) For $\forall x_0 \in U$, if $\inf_{x \in \mathcal{G}_t(a_j)} v(x, a_j) \leqslant v(x_0, a_j) \leqslant \sup_{x \in \mathcal{G}_t(a_j)} v(x, a_j)$, then $x_0 \in \mathcal{G}_t(a_j)$.

Then $\mathcal{G}_t(a_j)$ is defined as a Ascending Maximal Monotonic Interval, shorted for AMMI, where $\inf_{x \in M_t} v(x, a_j)$, $\sup_{x \in M_t} v(x, a_j)$ are the infimum and supremum respectively. By exchanging the $x_i \leqslant_D x_k$ with $x_i \geqslant_D x_k$ in 1), the $\mathcal{G}_t(a_j)$ is defined as a Descending Maximal Monotonic Interval, short for DMMI. AMMI and DMMI are together referred to as MMI.

Based on the range of MMI's feature values, we can define the dominance relationships of feature a_j among MMIs. This dominance relationship has a certain similarity to the dominance relationship between interval values. In the related literature [26], the degree of interval dominance has been explicitly defined. In this paper, due to non-overlap between every two MMIs, it is much easier to judge the dominance relationship. The dominance relationship between two MMIs is defined.

Definition 8. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $a_j \in A$ be a feature, $\mathcal{G}_t(a_j)$ and $\mathcal{G}_u(a_j)$ be two MMIs induced by a_j. If $min\{v(x, a_j)|x \in \mathcal{G}_t(a_j)\} \geqslant max\{v(x, a_j)|x \in \mathcal{G}_u(a_j)\}$, then $\mathcal{G}_t(a_j) \geqslant \mathcal{G}_u(a_j)$; if $max\{v(x, a_j)|x \in \mathcal{G}_t(a_j)\} \leqslant min\{v(x, a_j)|x \in \mathcal{G}_u(a_j)\}$, then $\mathcal{G}_t(a_j) \leqslant \mathcal{G}_u(a_j)$. In other conditions, we say $\mathcal{G}_t(a_j)$ is incomparable with $\mathcal{G}_u(a_j)$.

Then Monotonic Partition under feature a_k are defined based on the concept of MMI.

Definition 9. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $U = \{x_1, x_2, \ldots, x_n\}$, $A = \{a_1, a_2, \ldots, a_m\}$. For any $a_j \in A$, $\{\mathcal{G}_1(a_j), \mathcal{G}_2(a_j), \ldots, \mathcal{G}_h(a_j)\}$ are all AMMIs induced by feature a_j, $\bigcap_{t=1}^{h} \mathcal{G}_t(a_j) = \emptyset, \bigcup_{t=1}^{h} \mathcal{G}_t(a_j) = U$. If $\mathcal{G}_1(a_j) \leqslant \mathcal{G}_2(a_j) \leqslant \ldots \leqslant \mathcal{G}_h(a_j)$, then $MP^{\leqslant}(a_j) = \{\mathcal{G}_1(a_j), \mathcal{G}_2(a_j), \ldots, \mathcal{G}_h(a_j)\}$ is defined as Ascending Monotonic Partition induced by a_j, short for AMP. If $\mathcal{G}_1(a_j) \geqslant \mathcal{G}_2(a_j) \geqslant \ldots \geqslant \mathcal{G}_h(a_j)$, and all MMIs are DMMI, then $MP^{\geqslant}(a_j) = \{\mathcal{G}_1(a_j), \mathcal{G}_2(a_j), \ldots, \mathcal{G}_h(a_j)\}$ is defined as Descending Monotonic Partition induced by a_j, short for DMP.

AMP and DMP are collectively referred to as MP. Each feature can form an AMP and a DMP. Consequently, the Ascending and Descending Monotonic Granule Family can be defined based on this observation, denoted by $\Delta^{\leqslant}$ and $\Delta^{\geqslant}$.

Definition 10. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, the Ascending and Descending Monotonic Granule Family $\Delta^{\leqslant}$ and $\Delta^{\geqslant}$ are defined as

$$\Delta^{\leqslant} = \{MP^{\leqslant}(a_j)|a_j \in A\}, \tag{19}$$

$$\Delta^{\geqslant} = \{MP^{\geqslant}(a_j)|a_j \in A\}. \tag{20}$$

Algorithm 1 is presented to obtain $\Delta^{\leqslant}$ from an ordinal decision table $(U, A \cup \{D\})$, which calculates an MP for each feature, utilizing sorting and comparison operations. The time complexity of sorting operation is $O(nlogn)$ for each $a_j \in A$, while the time complexity of comparison operation is $O(mn)$ for all features. Consequently, the time complexity of Algorithm 1 is $O(mn + m * nlogn) = O(mn(1 + logn)) = O(mnlogn)$. To store the generated $\Delta^{\leqslant}$, a space of $O(mn)$ is required. By replacing the $\leqslant$ to $\geqslant$ in step 9, we can obtain the $\Delta^{\geqslant}$.

In Algorithm 1, the subscripts of x_i and x_k denote the indices of the samples after sorting. The samples to be incorporated into $MP^{\leqslant}(a_j)$ are those with original indices that correspond to these new subscripts following the sorting process.

Example 1.

An artificial dataset of twelve samples is used to demonstrate AMP generation, with nine for training and three for testing.

In the initial step, the ordinal decision table is sorted by each feature to create AMPs. Taking feature d for an example, the sample x_2 with the lowest value initiates the first AMMI, $\mathcal{G}_1(d)$. After six iterations of expansion via the algorithm, $\mathcal{G}_1(d)$ includes samples $\{x_2, x_1, x_3, x_4, x_5, x_9, x_7\}$.

Algorithm 1 Obtaining a monotonic granules family $\Delta^{\leqslant}$

Require: An ordinal decision table $(U, \{A \cup D\})$.
Ensure: $\Delta^{\leqslant}$ that generated from the decision table.
1: $|U|$ and $|A|$ are the number of samples and features of ODT.
2: initialize $\Delta^{\leqslant} = \emptyset$.
3: **for** each $a_j \in A$ **do**
4: Sort samples by ascending feature values of a_j, for those samples with the same feature value, sort them by ascending decision values.
5: initialize $i = 1, MP^{\leqslant}(a_j) = \emptyset$.
6: **while** $i \leqslant |U|$ **do**
7: $k = i$; % i and k are sample indexes ranked by a_j.
8: **while** $k + 1 \leqslant |U|$ and $x_k \leqslant_D x_{k+1}$ **do**
9: $k = k + 1$;
10: **end while**
11: $MP^{\leqslant}(a_j) \leftarrow \{x_i, \ldots, x_k\}$;
12: $i = k + 1$;
13: **end while**
14: $\Delta^{\leqslant} \leftarrow MP^{\leqslant}(a_j)$;
15: **end for**

TABLE 1: An artificial ordinal decision table

U/A	a	b	c	d	e	f	g	h	D
x_1	0.1	0.0	0.3	0.2	1.0	0.4	0.6	0.8	1
x_2	0.3	0.1	0.6	0.0	0.6	0.2	0.1	0.9	1
x_3	0.3	0.0	0.5	0.5	0.9	0.0	0.7	0.6	1
x_4	0.1	0.3	0.3	0.5	0.1	0.1	0.8	0.7	2
x_5	0.3	0.2	0.7	0.7	0.0	0.4	0.4	0.5	2
x_6	0.0	0.5	0.1	1.0	0.1	0.5	0.1	0.3	2
x_7	0.6	0.5	0.1	0.8	0.6	0.5	0.0	0.2	3
x_8	0.1	0.6	1.0	1.0	0.2	0.7	0.3	0.2	3
x_9	0.1	1.0	1.0	0.7	0.1	1.0	1.0	0.0	3

Sample x_6 with decision value 2 does not fit the monotonicity of $\mathcal{G}_1(d)$ and is thus excluded, initiating the next AMMI. Following the same method, the second AMMI, $\mathcal{G}_2(d)$, is formed with x_6 and x_8. Together, $\mathcal{G}_1(d)$ and $\mathcal{G}_2(d)$ cover the entire set U, creating the AMP $MP^{\leqslant}(d)$. This process highlights the dataset's strong monotonicity with respect to feature d and leads to the derivation of $\Delta^{\leqslant}$.

An MMI maintains local monotonicity by ensuring consistency between features and decisions, and we define its fitting degree accordingly.

Definition 11. Let $(U, A \cup \{D\})$ be an ordinal decision table, $MP^{\leqslant}(a_j) = \{\mathcal{G}_1(a_j), \mathcal{G}_2(a_j), \ldots, \mathcal{G}_h(a_j)\} \in \Delta^{\leqslant}$ represents the AMP formed by $a_j \in A$. If $x_i \in \mathcal{G}_t(a_j)(1 \leqslant t \leqslant h)$, the ascending fitting degree of sample x_i under feature a_j is defined as

$$\mathcal{F}^{\leqslant}(x_i, a_j) = \frac{|\mathcal{G}_t(a_j) \cap [x_i]_D|}{|[x_i]_D|}(1 - \frac{2 \cdot inver(a_j)}{N(N-1)}) \tag{21}$$

and the ascending significance of feature a_j is defined as

$$\mathcal{F}^{\leqslant}(a_j) = \sum_{1 \leqslant i \leqslant |U|} \mathcal{F}^{\leqslant}(x_i, a_j) \tag{22}$$

where $x_i \in \mathcal{G}_t(a_j)$, $|*|$ signifies the cardinality of $*$, and the function $inver(*)$ denotes the inversion number of decision values for all samples when sorted by feature a_j.

Equation (21) and (22) evaluate a feature a_j from two aspects: (1) local monotonicity: $\frac{|\mathcal{G}_t(a_j) \cap [x_i]_D|}{|[x_i]_D|}$ assesses the monotone consistency of the granule $\mathcal{G}_t(a_j)$; (2) global monotonicity: $(1 - \frac{2 \cdot inver(a_j)}{N(N-1)})$ evaluate the monotone consistency among all granules induced by feature a_j.

The ascending fitting degree $\mathcal{F}^{\leqslant}(x_i, a_j)$ of feature a_j and significance $\mathcal{F}^{\leqslant}(a_j)$ of a_j are defined based on $MP^{\leqslant}(a_j)$, analogous to their descending counterparts which use $MP^{\geqslant}(a_j)$. These measures evaluate feature a_j's correlation with the decision, accounting for both increasing and decreasing trends. The predominant trend is determined by comparing $\mathcal{F}^{\leqslant}(a_j)$ and $\mathcal{F}^{\geqslant}(a_j)$, with the higher value indicating the feature's monotonic relationship with the decision. For consistency in data processing, features that decrease monotonically are adjusted to reflect an increase. A fitting matrix is then constructed to systematically store these fitting degrees for all samples and features, aligning with the dimensions of the ordinal decision table.

Definition 12. A fitting degree matrix $M(U, A \cup D) = (c_{ij})_{n \times m}$ is defined as

$$c_{ij} = \mathcal{F}(x_i, a_j) \tag{23}$$

where $x_i \in U$, $a_j \in A$, n and m are the numbers of samples and features, respectively.

Algorithm 2 transforms the decreasing feature to increasing, and obtains the fitting degree matrix.

Algorithm 2 Obtaining the fitting degree matrix

Require: An ordinal decision table $(U, \{A \cup D\})$.
Ensure: The fitting degree matrix $M(U, A \cup D) = (c_{ij})_{n \times m}$
1: normalize the ordinal decision table;
2: $M(U, A \cup D) = (c_{ij})_{n \times m} = Zero(n, m)$;% n, m are numbers of samples and features, respectively;
3: form the $\Delta^{\leqslant}$ and $\Delta^{\geqslant}$ by Algorithm 1;
4: **for** each a_j in A **do**
5: compute the $\mathcal{F}^{\leqslant}(a_j)$ and $\mathcal{F}^{\geqslant}(a_j)$ by Definition 11;
6: **if** $\mathcal{F}^{\leqslant}(a_j) \geqslant \mathcal{F}^{\geqslant}(a_j)$ **then**
7: mark the feature as monotonic increase;
8: **else**
9: mark the feature as monotonic decrease;
10: replace $MP^{\leqslant}(a_j)$ in $\Delta^{\leqslant}$ with $MP^{\geqslant}(a_j)$;
11: **end if**
12: **end for**
13: **for** all features marked as monotonic decrease **do**
14: for all feature values v: $v = 1 - v$;
15: **end for**
16: set $c_{ij} = \mathcal{F}(x_i, a_j), 1 \leqslant i \leqslant n, a_j \in A$;
17: return $M(U, \{A \cup D\})$;

Example 2. (Follow up Example 1) Table 2 exhibit the fitting degree matrix of Table 1.

TABLE 2: Fitting degree matrix $M(U, A \cup D)$

U/A	a	b	c	d	e	f	g	h
x_1	0.23	1.0	0.24	0.94	0.35	0.63	0.41	0.19
x_2	0.46	1.0	0.48	0.94	0.18	0.63	0.2	0.19
x_3	0.46	1.0	0.48	0.94	0.35	0.31	0.41	0.09
x_4	0.23	1.0	0.24	0.63	0.53	0.31	0.2	0.09
x_5	0.23	1.0	0.24	0.63	0.53	0.63	0.2	0.19
x_6	0.23	1.0	0.24	0.31	0.53	0.63	0.2	0.19
x_7	0.23	1.0	0.24	0.63	0.18	0.94	0.2	0.28
x_8	0.46	1.0	0.48	0.31	0.35	0.94	0.2	0.28
x_9	0.46	1.0	0.48	0.63	0.35	0.94	0.2	0.28

3.2 Feature Selection Based on Monotonic Related Family

In most rough set-based feature selection algorithms, the neighborhood radius parameter is important for the classification performance. However, the value set of the parameter, interval $[0, 1]$, is an infinite set, implying the difficulty of optimal parameter searching. Unlike other feature selection algorithms [16], [19], [27], [44], [45], the value set of parameter σ in this paper, which is finite, can be derived from the fitting degree matrix. The MMIs generated from ODT are finite, so the values of fitting degree also constitute a finite set, with each value being applicable as a parameter σ. We primarily focus on samples with fitting degrees exceeding σ.

Due to the monotone consistency of all granules produced in Subsection 3.1, the lower approximation of any monotonic decision class ($d_i^{\leqslant}$ or $d_i^{\geqslant}$) is itself based on Definition 2, which can not be applied in this paper. Consequently, utilizing the fitting degree and parameter σ, we established both upper and lower approximations to formulate a novel rough set model and subsequently defined the σ-positive region.

Definition 13. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $B \subseteq A$, the monotonic lower and upper approximation of d_i are defined as

$$\underline{R_B^{\geqslant}} d_i^{\geqslant} = \{x | x \in d_i^{\geqslant} \ and \ \exists a_j \in B, \ s.t. \ \mathcal{F}^{\geqslant}(x, a_j) \geqslant \sigma\} \tag{24}$$

$$\underline{R_B^{\leqslant}} d_i^{\leqslant} = \{x | x \in d_i^{\leqslant} \ and \ \exists a_j \in B, \ s.t. \ \mathcal{F}^{\leqslant}(x, a_j) \geqslant \sigma\} \tag{25}$$

$$\overline{R_B^{\geqslant}} d_i^{\geqslant} = U - \underline{R_B^{\leqslant}} d_{i-1}^{\leqslant} \tag{26}$$

$$\overline{R_B^{\leqslant}} d_i^{\leqslant} = U - \underline{R_B^{\geqslant}} d_{i+1}^{\geqslant} \tag{27}$$

Definition 14. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $B \subseteq A$, $a_j \in B$, σ is the parameter of fitting degree. The σ-positive region of ODT is

$$POS_B^{\sigma \leqslant}(D) = \{x | \exists a_j \in B, \ s.t. \ \mathcal{F}^{\leqslant}(x, a_j) \geqslant \sigma\} \tag{28}$$

$$POS_B^{\sigma \geqslant}(D) = \{x | \exists a_j \in B, \ s.t. \ \mathcal{F}^{\geqslant}(x, a_j) \geqslant \sigma\} \tag{29}$$

when there is no confusion, we omit $\leqslant$ or $\geqslant$ from the superscript.

Drawing upon the concept of a σ-positive region, we proceed to define the notion of a σ-reduct.

Definition 15. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $P \subseteq A$, σ is the parameter of fitting

degree. For any $a \in A$, if $POS^{\sigma}_{A-\{a\}}(D) = POS^{\sigma}_{A}(D)$, we say a is dispensable in A. Otherwise, a is indispensable in A. If $POS^{\sigma}_{P}(D) = POS^{\sigma}_{A}(D)$ and any $p \in P$ is indispensable, then P is a σ-reduct of A. The collection of all indispensable elements in A is called core, denoted by $CORE^{\sigma}(A)$. The collection of all σ-reducts of A is denoted by $RED^{\sigma}(A)$.

In this paper, to facilitate data processing, we adjust monotonically decreasing features to increasing. Thus, if there is no extra information, $POS^{\sigma}_{A}(D)$, σ-reduct, $CORE^{\sigma}(A)$ and $RED^{\sigma}(A)$ are refer to the notions based on monotonically increasing.

Theorem 1. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table. For $P, Q \subseteq A$, if $P \subseteq Q$, then $POS^{\sigma}_{P}(D) \subseteq POS^{\sigma}_{Q}(D)$.

Proof: If $P = Q$, we have $POS^{\sigma}_{P}(D) = POS^{\sigma}_{Q}(D)$ by Definiton 15. If $P \subset Q$, let $K = Q - P$, then $POS^{\sigma}_{Q}(D) = POS^{\sigma}_{P\cup K}(D) = \{x | \exists a_j \in P \cup K,\ s.t.\ \mathcal{F}^{\leqslant}(x, a_j) \geqslant \sigma\} = POS^{\sigma}_{P}(D) \cup POS^{\sigma}_{K}(D)$. Thus $POS^{\sigma}_{P}(D) \subseteq POS^{\sigma}_{Q}(D)$. □

Theorem 1 highlights the non-decreasing nature of the positive region with growing feature subsets. The fitting degree matrix captures the monotonic consistency across samples, where an increase in a sample's fitting degree indicates a stable or enhanced monotonic relationship with the decision classes. As the fitting degree of a particular sample escalates, the MMI encompassing it either resides in an identical decision domain (sole decision class) or exhibits enhanced monotonicity (multiple decision classes), which can be discerned from the matrix itself. This framework lays a solid theoretical foundation for feature selection. Subsequently, we introduces the concepts of monotonic related sets and family to enable the process.

Definition 16. Let $(U, A \cup \{D\})$ be an ordinal decision table, $A = \{a_1, a_2, \ldots, a_m\}$ and $D = \{d_1, d_2, \ldots, d_k\}$, the monotonic related set is defined as

$$mr(x_i) = \{a_j \in A | \mathcal{F}^{\leqslant}(x_i, a_j) \geqslant \sigma\} \tag{30}$$

The monotonic related family $MR(U)$ of $(U, A \cup \{D\})$ is defined as

$$MR(U) = \{mr(x_i) | x_i \in U\} \tag{31}$$

Theorem 2 elucidates the relationship between monotonic related family, positive region, reducts and core.

Theorem 2. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $mr(x_i)$ is the monotonic related set of sample x_i, $P \subseteq A$, σ is the parameter of fitting degree, then

1) $POS^{\sigma}_{P}(D) = POS^{\sigma}_{A}(D)$ if and only if $P \cap mr(x_i) \neq \emptyset$ for any $mr(x_i) \neq \emptyset$.
2) P is a σ-reduct of A if P is a minimal subset of A such that $P \cap mr(x_i) \neq \emptyset$ for any $mr(x_i) \neq \emptyset$.
3) $CORE^{\sigma}(A) = \{a | \exists mr(x_i) \in MR(U)\ s.t.\ mr(x_i) = \{a\}\}$.

Proof:

1) Suppose for any $mr(x_i) \neq \emptyset$, $P \cap mr(x_i) = \emptyset$. For any $x_i \in POS^{\sigma}_{A}(D)$, it is obvious that $mr(x_i) \neq \emptyset$, since $P \cap mr(x_i) = \emptyset$, then exists $p \in P$ such that $p \in mr(x_i)$. Thus $\mathcal{F}^{\leqslant}(x_i, p) \geqslant \sigma$, which means $x_i \in POS^{\sigma}_{P}(D)$, then $POS^{\sigma}_{A}(D) \subseteq POS^{\sigma}_{P}(D)$. Considering $P \subseteq A$, $POS^{\sigma}_{A}(D) = POS^{\sigma}_{P}(D)$.
2) It is evident.
3) ($\Rightarrow$) Suppose $q \in CORE^{\sigma}(A)$, then q is indispensable in A, then $POS^{\sigma}_{A-\{q\}}(D) \neq POS^{\sigma}_{A}(D)$. Then $\exists x_i \in U\ s.t.\ x_i \in POS^{\sigma}_{A}(D)$ and $x_i \notin POS^{\sigma}_{A-\{q\}}(D)$. It is evident that p is the only feature in A such that $\mathcal{F}^{\leqslant}(x_i, p) \geqslant \sigma$, then $mr(x_i) = \{q\}$. it is evident that $CORE^{\sigma}(A) \subseteq \{a | \exists mr(x_i) \in MR(U)\ s.t.\ mr(x_i) = \{a\}\}$.
($\Leftarrow$) Suppose for any $a \in A$, if $\exists mr(x_i) \in MR(U)\ s.t.\ mr(x_i) = \{a\}\}$, then a is the only feature such that $\mathcal{F}^{\leqslant}(x_i, a) \geqslant \sigma$, then $x_i \notin POS^{\sigma}_{A-\{a\}}(D)$. It means $x_i \in POS^{\sigma}_{A}(D)$ and $x_i \notin POS^{\sigma}_{A-\{a\}}(D)$, then $POS^{\sigma}_{A}(D) = POS^{\sigma}_{A-\{a\}}(D)$, which means a is indispensable in A, thus $\{a | \exists mr(x_i) \in MR(U)\ s.t.\ mr(x_i) = \{a\}\} \subseteq CORE^{\sigma}(A)$

□

Then the matrix representation $M^{\sigma}(U, A \cup \{D\})$ of the monotonic related family are defined.

Definition 17. A σ-related matrix $M^{\sigma}(U, A \cup \{D\}) = (e_{ij})_{n\times m}$ is defined as

$$e_{ij} = \begin{cases} 1, \text{ if } a_j \in mr(x_i), \\ 0, \text{ otherwise,} \end{cases} \tag{32}$$

where $x_i \in U$, $a_j \in A$, n and m are the number of samples and features, respectively.

Theorem 3 posits that based on the fitting degree matrix and the σ-related matrix, the positive region of any feature subset can be generated; therefore, we can perform feature selection based on these two matrices or monotonic related family.

Theorem 3. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $M(U, A \cup \{D\}) = (c_{ij})_{n\times m}$ be the fitting degree matrix, and $M^{\sigma}(U, A \cup \{D\}) = (e_{ij})_{n\times m}$ be the σ-related matrix. Then

1) For any $P \subseteq A$, $POS^{\sigma}_{P}(D) = \{x_i | e_{ij} = 1\ and\ a_j \in P\}$.
2) $POS^{\sigma}_{P}(D) = POS^{\sigma}_{A}(D)$ if and only if for any $x \in POS^{\sigma}_{A}(D)$, there is $a_j \in P$ such that $e_{ij} = 1$.
3) P is a σ-reduct of A if P is a minimal subset of A such that for any $x \in POS^{\sigma}_{A}(D)$, there is $a_j \in P$ such that $e_{ij} = 1$.
4) $CORE^{\sigma}(A) = \{a_j \in A | e_{ij} = 1\ and\ \forall t \neq j, e_{it} = 0\}$.

The σ-related matrix $M^{\sigma}(U, A \cup \{D\})$ and the monotonic related family $MR(U)$ can be generated based on the fitting degree matrix $M(U, A \cup \{D\})$ as shown in the following theorem.

Theorem 4. Let $ODT = (U, A \cup \{D\})$ be an ordinal decision table, $M(U, A \cup \{D\}) = (c_{ij})_{n\times m}$ be the fitting degree matrix, σ be the parameter, $MR(U)$ be the monotonic related family of $(U, A \cup \{D\})$ and $M^{\sigma}(U, A \cup \{D\}) = (e_{ij})_{n\times m}$ be the σ-related matrix.

1) For $\forall x_i \in U$, $mr(x_i) = \{a_j | c_{ij} \geqslant \sigma\}$.
2) For $\forall x_i \in U$, $mr(x_i) = \{a_j | e_{ij} = 1\}$.

3) For $\forall x_i \in U$ and $\forall a_j \in A$, $e_{ij} = 1$ if and only if $c_{ij} \geqslant \sigma$.

Proof:

1) For x_i and a_j, if $c_{ij} \geqslant \sigma$, then $\mathcal{F}(x_i, a_j) \geqslant \sigma \Leftrightarrow a_j \in mr(x_i)$.
2) For x_i and a_j, $e_{ij} = 1 \Leftrightarrow a_j \in mr(x_i)$.
3) For x_i and a_j, $e_{ij} = 1 \Leftrightarrow a_j \in mr(x_i) \Leftrightarrow \mathcal{F}(x_i, a_j) \geqslant \sigma \Leftrightarrow c_{ij} \geqslant \sigma$

□

Example 3. (Follow up Example 2) The corresponding matrix representation with parameter $\sigma = 0.5$ can be found in Table 3. In each row, the columns with a value of 1 form the monotonic related set for the sample. For example, $mr(x_1) = \{b, d, f\}$.

TABLE 3: $M^{0.5}(U, A \cup \{D\})$ based on $MR(U)$

U/A	a	b	c	d	e	f	g	h
x_1	0	1	0	1	0	1	0	0
x_2	0	1	0	1	0	1	0	0
x_3	0	1	0	1	0	0	0	0
x_4	0	1	0	1	1	0	0	0
x_5	0	1	0	1	1	1	0	0
x_6	0	1	0	0	1	1	0	0
x_7	0	1	0	1	0	1	0	0
x_8	0	1	0	0	0	1	0	0
x_9	0	1	0	1	0	1	0	0

To showcase the superiority of the monotonic related family, we juxtapose it with the discernibility matrix. Because an $n \times m$ matrix is applied to finding reducts, the space complexity of the monotonic related family is $O(nm)$, in contrast to the discernibility matrix, which exhibits a space complexity of $O(n^2m)$. Consequently, the monotonic related family is capable of managing much larger scale data compared to the discernibility matrix.

Skowron and Rauszer introduced a feature selection approach that utilizes a discernibility function for the simultaneous identification of all reducts [46]. This method predominantly relies on the principles of absorption and distribution [47]. Similar to discernibility matrix, a boolean function $f(U, A \cup \{D\})$ is defined to acquire the all reducts based on the monotonic related family.

Definition 18. The monotonic related function with respect to $(U, A \cup \{D\})$ is defined as

$$f(U, A \cup \{D\}) = \wedge\{\vee(mr(x)) | \forall x \in U, mr(x) \neq \emptyset\} \quad (33)$$

where $\vee$ and $\wedge$ are the disjunction and conjunction operators, respectively.

By applying the absorption law and the distribution law, the disjunction norm form can be transformed into the conjunction, subsequently yielding multiple reducts. These reducts are represented as terms of the conjunction norm form. Theorem 5 elucidates how to derive the boolean monotonic related function $f(U, A \cup \{D\})$ from the fitting degree matrix and the σ-related matrix.

Theorem 5. Let A be the monotonic related family of $(U, A \cup \{D\})$, $M^\sigma(U, A \cup \{D\}) = (e_{ij})_{n\times m}$ be the σ-related matrix, $M(U, A \cup \{D\}) = (c_{ij})_{n\times m}$ be the fitting degree matrix, and $f(U, A \cup \{D\})$ be the monotonic related function. Then

1) $f(U, A \cup \{D\}) = \wedge\{\vee\{a_j | e_{ij} = 1\}) | \forall x_i \in U\}$
2) $f(U, A \cup \{D\}) = \wedge\{\vee\{a_j | c_{ij} \geqslant \sigma\}) | \forall x_i \in U\}$

Theorem 6 introduces the method of deriving all reducts via the monotonic related function $f(U, A \cup \{D\})$.

Theorem 6. Let A be the monotonic related family of $(U, A \cup \{D\})$, and $f(U, A \cup \{D\})$ be the monotonic related function. If $g(U, A \cup \{D\}) = (\wedge A_1) \vee (\wedge A_2) \ldots \vee (\wedge A_l)$ is the reduced disjunctive form transferred from $f(U, A \cup \{D\})$ via the laws of multiplication and absorption, which is, for any $A_k \subseteq A, k = 1, 2, ..., l$, there is no repeated element in A_k. Then $RED^\sigma(A) = \{A_1, A_2, \ldots, A_l\}$.

Proof: For every $k = 1, 2, \ldots, l$, $\wedge A_k \leqslant \vee mr(x_i)$ for any $mr(x_i) \in MR(U)$, so $A_k \cap mr(x_i) \neq \emptyset$. Let $A'_k = A_k - \{q\}$ for any $q \in A_k$, then $g(U, A \cup \{D\}) \leqslant \vee_{t=1}^{k-1}(\wedge A_t) \vee (\wedge A_k^{'}) \vee (\vee_{t=k+1}^{l}(\wedge A_t))$. If for every $mr(x_i) \in MR(U)$, we have $A_k^{'} \cap mr(x_i) \neq \emptyset$, then $\wedge A_k^{'} \leqslant \vee mr(x_i)$ for every $mr(x_i) \in MR(U)$. That is, $g(U, A \cup \{D\}) \geqslant \vee_{t=1}^{k-1}(\wedge A_t) \vee (\wedge A_k^{'}) \vee (\vee_{t=k+1}^{l}(\wedge A_t))$, which is a contradiction. It implies there is $mr(x_{i0}) \in MR(U)$ such that $A_k^{'} \cap mr(x_{i0}) = \emptyset$. Thus, A_k is a reduct of A.

For any $X \in RED^\sigma(A)$, we have $X \cap r(x_i) \neq \emptyset$ for every $mr(x_i) \in MR(U)$, so $f(U, A \cup \{D\}) \wedge (\wedge X) = \wedge(\vee mr(x_i)) \wedge (\wedge X) = \wedge X$, which implies $\wedge X \leqslant f(U, A \cup \{D\}) = g(U, A \cup \{D\})$. Suppose for every $k = 1, 2, \ldots, l$, we have $A_k - X \neq \emptyset$. Then, for every k, there is $C_k \in A_k - X$. By rewriting $g(U, A \cup \{D\}) = (\vee_{k=1}^{l} C_k) \wedge \Phi$, $\wedge X \leqslant \vee_{k=1}^{l} C_k$. Thus, there is C_{k_0} such that $\wedge X \leqslant C_{k_0}$, it means $C_{k_0} \in X$, which is a contradiction. So $A_{k_0} \subseteq X$ for some k_0, since both X and A_{k_0} are reducts, it is evident that $X = A_{k_0}$. Consequently, $RED^\sigma(A) = \{A_1, A_2, \ldots, A_l\}$. □

The monotonic related family is introduced to enhance the classification performance and robustness of monotonic models by generating diverse feature subsets, offering a theoretical basis for ensemble classifiers. While finding reducts via discernibility functions is NP-hard, the monotonic related function, despite being more computationally efficient, also presents an NP-hard challenge. To reduce the computation cost, a heuristic algorithm is designed to identify key features, with $RED^\sigma(A)$ denoting the set of all potential reducts.

Example 4. (Follow up Example 3)

$$\begin{aligned} f(U, A \cup \{D\}) =& \{b \vee f\} \wedge \{b \vee e \vee f\} \wedge \{b \vee d\} \wedge \\ & \{b \vee d \vee f\} \wedge \{b \vee d \vee e\} \wedge \\ & \{b \vee d \vee e \vee f\} \\ =& \{b \vee d\} \wedge \{b \vee f\} \\ =& \{b\} \vee \{d \wedge f\} \end{aligned}$$

Based on the absorption law in monotonic related function, the minimal elements which cannot be contained by other elements in the monotonic related family are sufficient to find all reducts. For instance, in Example 4, $\{b, d\}$ and $\{b, f\}$ are enough for finding all reducts. Deleting all non-minimal elements from monotonic related functions decreases the time complexity and the

computational load. Then all reducts, $\{b\}$ and $\{d \wedge f\}$, could be obtained.

Algorithm 3 Obtaining feature subsets by fitting degree matrix

Require: Fitting degree matrix=$(c_{ij})_{n\times m}$; parameter σ.
Ensure: A set of feature subsets $RED^{\sigma}(A)$.
1: initialize the feature subsets $RED^{\sigma}(A) = \emptyset$.
2: set c_{ij} as 1 if $c_{ij} \geqslant \sigma$ and the others as 0, denote the resulting matrix as $M^{\sigma}(U, A \cup \{D\}) = (e_{ij})_{n\times m}$.
3: sort the rows in ascending by each row sum $\sum_{j=1}^{m} e_{ij}$.
4: remove the rows that are all 0 or all 1.
5: absorb $M^{\sigma}(U, A \cup \{D\}) = (e_{ij})_{n\times m}$, denote the resulting matrix as $M^{*}(U, A \cup \{D\}) = (e^{*}_{ij})_{n^{'}\times m^{'}}$.
6: let $OFS = \{a_k \in A | e^{*}_{1k} = 1\}$. % find feature subsets
7: **for** a_k in OFS **do**
8: red = $\{a_k\}$.
9: update $M^{*}(U, A \cup \{D\})$: delete the rows satisfied $e^{*}_{ik} = 1$ and the kth column.
10: update $n^{'}$, $A = A - \{a_k\}$, $m^{'} = |A|$.
11: **while** $n^{'} \neq 0$ and $m^{'} \neq 0$ **do**
12: $a_k = argmax_{a_k}(\sum_{i=1}^{n^{'}} e^{*}_{ik})$.
13: $red \leftarrow a_k$.
14: update $M^{*}(U, A \cup \{D\})$: delete the rows satisfied $e^{*}_{ik} = 1$, delete the kth column.
15: update $n^{'}$, $A = A - \{a_k\}$, $m^{'} = |A|$.
16: **end while**
17: $RED^{\sigma}(A) \leftarrow red$.
18: **end for**
19: **return** $RED^{\sigma}(A)$.

Algorithm 3 selects the monotonic related set with the minimum number of elements as the Original Features Set (OFS) for efficient feature subset generation, then eliminated duplicate feature subsets. Constructing $M^{\sigma}(U, A \cup D)$ takes $O(mnlogn)$ time with parameter σ, and generating $M^{*}(U, A \cup D)$ requires $O(nlogn)$ time. The process of extracting feature subsets for OFS elements concludes with a time complexity of $O(mn)$. Overall, the complexity for subset acquisition is $O(mn(logn + l))$, with l as OFS's size. This approach significantly reduces feature selection time complexity.

4 Bi-direction Fusing Monotonic Decision Trees

Paper [6] shows that the RMI metric approaches zero with increasing feature correlation, where $RMI > 0$ suggests a positive correlation and $RMI < 0$ indicates a negative one. Both ARMI and DRMI can measure this monotonic correlation and have been used to build and refine monotonic decision trees for improved classification [4].

4.1 Decision Support Level (DSL)

A standalone decision tree classifier is prone to overfitting and underfitting complications. As a solution to this predicament, prior research has demonstrated the efficacy of ensemble learning in enhancing the generalization capacity of a classifier [7], [12], [14], [16], [19]. To amalgamate multiple classifications from disparate decision trees, we advocate for the incorporation of the Decision Support Level (DSL) as a pivotal factor.

Definition 19. [48] Let $Leaf_i$ is a leaf node of a monotonic decision tree, the set of samples in $Leaf_i$ is U_i, their decision class is $\{d_1, \ldots, d_m\}$, the DSL of d_k on $Leaf_i$ is defined as

$$Support(Leaf_i, d_k) = \frac{|\{x \in U_i | v(x, D) = d_k\}|}{|U_i|} \quad (34)$$

The DSL characterizes the bias for each classification based on impure leaf nodes within a decision tree. For an individual decision tree, it is known that $\sum_{k=1}^{m} Support(Leaf_i, d_k) = 1$, and the classification possessing the highest DSL is preferred as the ultimate classification. If there is a unique decision class d_1 in a leaf node (pure node), then $Support(Leaf_i, d_1) = 1$. In the case where the decision class d_n is absent in $Leaf_i$, it follows that $Support(Leaf_i, d_n) = 0$.

Upon making decisions using n monotonic decision trees, one sample belongs to n leaf nodes for n decision trees, represented by $\{Leaf_1, \ldots, Leaf_n\}$. The decision classes present in all leaf nodes are denoted by $\{d_1, \ldots, d_m\}$. The amalgamated leaf node is symbolized by $Leaf_{all}$. The DSL of each decision class is computed as follows

$$\begin{gathered} Support(Leaf_{all}, d_k) = \sum_{i=1}^{n} Support(Leaf_i, d_k) \\ k = 1, 2 \ldots, m \end{gathered} \quad (35)$$

Finally, we set the decision class with highest DSL as the final classification label for the sample.

$$\begin{gathered} d = argmax_{d_k}(Support(Leaf_{all}, d_k)) \\ k = 1, 2 \ldots, m \end{gathered} \quad (36)$$

All monotonic decision trees are used for making more robust and accurate classifications.

4.2 Fusing Monotonic Decision Tree Algorithm

Algorithm 3 selects feature subsets using a threshold σ, enabling the creation of multiple monotonic trees. With each subset, two trees are built using ARMI and DRMI, leading to $2h$ trees from h subsets. The leaf nodes from these trees are combined, with the classification having the highest DSL being chosen. This approach, termed BFMDT, has a time complexity of $O(hn + k_1 mvn^2 + k_2 n)$, depending on feature dimension m, the number of nodes k_1, k_2 and possible split points v. We provide a flowchart Fig. 2 to illustrate the overall algorithmic process.

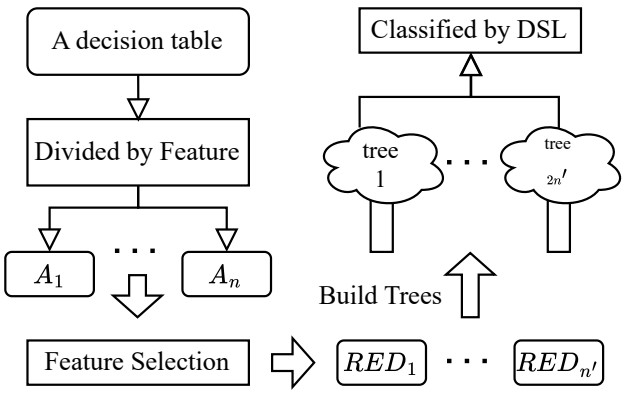


Fig. 2: A flowchart described the algorithmic process

Algorithm 4 BFMDT

Require: Ordinal decision table $ODT = (U, \{A \cup D\})$; stoping criterion δ; samples to be predicted $X = \{x_1, x_2, \ldots, x_n\}$.

Ensure: Highest classifcation accuracy AC_{max}, MAE_{min}; optimal threshold σ_{best}.

Initialize $AC_{max} = 0$, σ_{best}=0
obtain fitting degree matrix by Algorithm 1 and 2
for each σ **do**
  get $RED^{\sigma}(A)$ from Algorithm 3
  for each feature subset red in $RED^{\sigma}(A)$ **do**
    generate two trees by ARMI and DRMI
  end for
  for each $x \in X$ **do**
    compute DSL of all possible decisions by all trees
    compute the decision of x by Equation 36
  end for
  compute the accuracy and mean absolute error of X, marked as AC and MAE
  if $AC > AC_{max}$ **then**
    $AC_{max} = AC$, $\sigma_{best} = \sigma$, $MAE_{min} = MAE$
  end if
end for
return AC_{max}, MAE_{min} and σ_{best}.

Example 5. (Follow up Example 4)
Apply Algorithm 4 to build four decision trees based on feature subsets $\{b\}$ and $\{d \wedge f\}$ shown in Fig. 3.

5 Experiment Analysis

In this section, we demonstrate the efficacy of the BFMDT algorithm by comparing it to six representative monotonic classifiers.

5.1 Benchmark Methods and Datasets

We evaluated our method on 18 benchmark datasets, detailed in Table 4, using 5-fold cross-validation to ensure reliability. Parallel experiments were executed for each fold. The only exception is dataset PEMS-SF, due to the excessive number of features, we only compared the BFMDT algorithm with the REMT algorithm and employed an 8-2 split approach for the experiments. Experiments were run in parallel on an Apple Silicon M1 Pro CPU with 16G memory, using MATLAB 2023b on macOS.

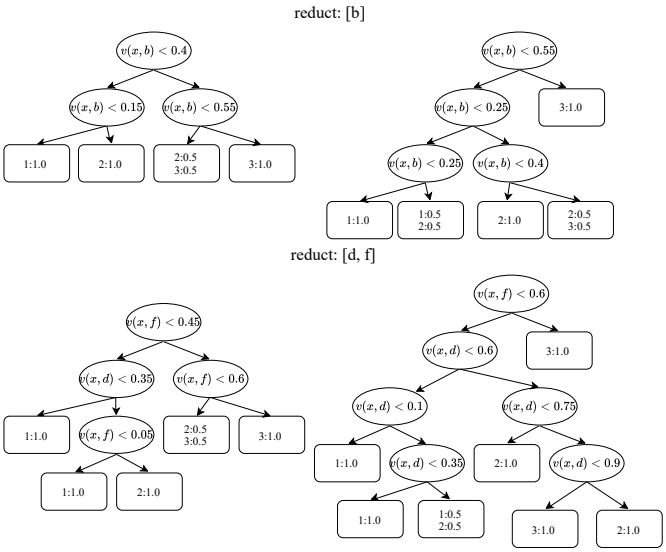


Fig. 3: Four trees built by ARMI (left) and DRMI (right), ←:Yes, →:No

TABLE 4: Datasets in the experimental analysis

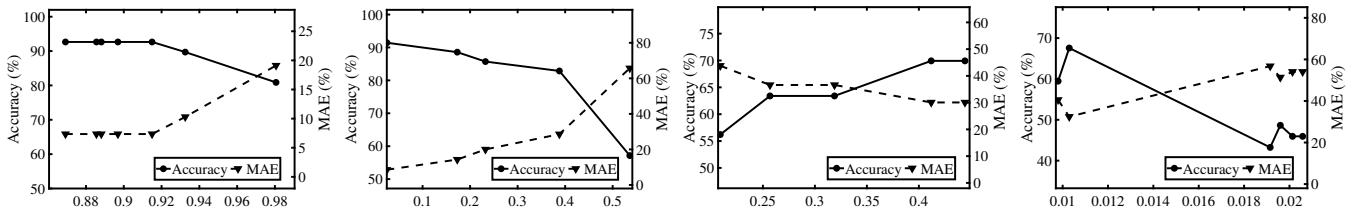
ID	dataset	objects	features	class	sources
1	breast-wisconsin	683	10	2	UCI
2	wine	178	14	3	UCI
3	breast-cancer	277	10	2	UCI
4	heart-disease	270	14	2	UCI
5	hepatitis	80	20	2	UCI
6	german-credit	1000	21	2	UCI
7	vehicle	946	19	4	UCI
8	wdbc	569	31	2	mclust
9	diabetes	768	9	2	UCI
10	wine-quality	4898	12	7	UCI
11	divorce	170	55	2	AIStudio
12	sonar	208	61	2	UCI
13	turkiye-student	5820	33	5	UCI
14	Yale	165	1025	15	jundongl
15	arcene	200	10001	2	UCI
16	SMK_CAN_187	187	19994	2	jundongl
17	DrivFace	606	6401	3	UCI
18	PEMS-SF	440	138673	7	UCI

5.2 Data Pre-processing

To guarantee a uniform dimension for each feature, range normalization is employed on all datasets during the training phase. In some specific datasets, certain features might encompass unique values that do not contribute to the classification process. We eliminated these features from the datasets. In scenarios where a particular feature might contain missing values, we employ the mean of all available values within that feature as an appropriate substitute. Instances with missing values in the decision are subsequently excluded from the dataset, without undergoing any additional processing.

5.3 Evaluation Measures

Experiments recorded classification accuracy (CA) and mean absolute error (MAE) to assess proposed method and reference models. We also timed the feature selection and classification time for each dataset.



(a) breast-wisconsin (b) wine (c) diabetes (d) SMK_CAN_187

Fig. 4: AC and MAE of datasets with parameter σ

5.4 Experimental Results and Analysis

5.4.1 Selecting of fitting degree values

The BFMDT algorithm assigns a unique σ value set for each dataset based on fitting degrees. This is beneficial for small datasets, allowing for an exhaustive search for optimal classification. However, for large datasets, this exhaustive approach can be prohibitively time-consuming due to the complexity of constructing decision trees. Therefore, it's crucial to find a σ selection method that balances classification performance with time efficiency, especially as higher-dimensional data increases tree construction time. In general, a larger MMI encompasses more samples and exhibits a higher fitting degree, ensuring an elevated likelihood of being regarded as a key MMI.

We propose an efficient parameter selection scheme. Initially, we identify all distinct values within the matrix $M(U, A \cup D)$. Each value v and its frequency in the matrix is represented by $freq(v)$. We set the initial value $sp = 1$, and $L = max\{m, n\}$, where n is the number of rows and m is the number of columns of the fitting matrix. Then all values v that satisfy $freq(v) > L/sp$ are considered as the potential candidates for the parameter σ. If the number of all potential values in the set exceeds the predefined range (for this study, between 5 and 30), we adjust the value of sp by either increasing it by a factor of 1.1 (i.e., $sp \times 1.1$) or decreasing it by a factor of 0.9 (i.e., $sp \times 0.9$), until the criteria are met. This method aids in reducing computational time while preserving a satisfactory classification performance. Nonetheless, a certain degree of performance degradation is inevitable in comparison to an approach utilizing all fitting degrees. Figure 4 demonstrate the relationship between σ and the corresponding classification accuracy for some datasets, utilizing an 80% training dataset proportion. As evident, the quantity of parameter σ tends to increase with the increasing scale of the data.

For each σ value, if the $RED^{\sigma}(A)$ generated by Algorithm 3 contains elements more than 50, we choose the first 50 feature subsets of $RED^{\sigma}(A)$ to build the trees. The purpose is to balance time cost and classification performance.

5.4.2 Analysis on AC and MAE

We compared BFMDT with three algorithms that incorporated feature selection: the fuzzy dominance discernibility matrix-based algorithm (FFREMT) [16]; the dominance discernibility matrix-based algorithm (FCMT) [19]; and three fuzzy significance measure-based algorithms (FPRS_up, FPRS_down, FPRS_global) [33]. Additionally, we evaluated two decision tree algorithms: the original monotonic decision tree algorithm (REMT) [6] and the ensemble learning-based algorithm (Bagging) [12]. To enhance the diversity and universality of the experiments, we also compared a monotonic kNN-based algorithm (FMKNN) [11]. Table 5 presents the complexity of BFMDT and five aforementioned comparison algorithms. It's worth noting that BFMDT reduces the time complexity of feature selection, in terms of both sample scale and feature scale. The results of experiments are presented in Tables 6 and 7.

High-dimensional datasets are those with ID: 14-18, as listed in Table 4. Due to their prolonged running times, significance-based algorithms failed to complete the feature selection tasks. Discernibility matrix-based approaches demand exponentially increasing amounts of memory when performing disjunction and conjunction operations on distinguishable features. Thus, both FFREMT and FCMT were unable to complete experiments on high-dimensional datasets 14-17 due to memory limitations. FCMT could not complete the experiment on dataset 11 owing to insufficient memory. Three FPRS algorithms on datasets 13-17, FFREMT on dataset 10, and the Bagging algorithm on dataset 17 failed to complete the experiments due to unacceptable execution time. In contrast, REMT, and BFMDT successfully executed all experiments except dataset 18, where Dataset 18 was utilized for comparison between BFMDT and REMT.

In the context of datasets 1-10, the result shows that BFMDT's average accuracy and MAE were 77.77% and 25.40%, respectively. The BFMDT algorithm achieved the highest classification accuracy on eight datasets and the optimal MAE on seven datasets, compared with other algorithms.

On the larger-scale datasets 11-18, the increased data scale led to the inabilities of some algorithms, which were listed as "/" (insufficient memory) and "−" (out of time) in Tables 9 and 10. On seven datasets (among eight large scale datasets), 11, 12, 13, 14, 16, 17, and 18, the BFMDT algorithm achieved the best classification performance. We conducted a comparison of the BFMDT algorithm with other algorithms on datasets where they could each complete the experiments. Table 8 shows the CA improvement and MAE decrease ratio of BFMDT compared to other algorithms.

These results underscore the competitiveness of the BFMDT algorithm in both classification accuracy and MAE, compared to other methods under the same experimental conditions. Especially, only BFMDT, REMT and FMKNN can handle dataset 18, and the BFMDT algorithm performs 8.45% and 10.65% better than REMT on classification accuracy and MAE. Due to the large number of features in

TABLE 5: The comparison of time and space complexity

Algorithms	Feature-selection		Classification	Space Complexity
	Time Complexity	Time Complexity Ratio	Time Complexity	
BFMDT	$O(mn(logn+l))$	1	$O(hn+k_1m^{'}vn^2+k_2n)$	$O(mn)$
FFREMT	$O((m+V)n^2)$	$(1+V/m)n/logn$	$O(hn+k_1m^{'}vn^2+k_2n)$	$O(mn^2)$
FCMT	$O(tmn^2)$	$tn/logn$	$O(hn+k_1m^{'}vn^2+k_2n)$	$O(mn^2)$
FPRS	$O((nlogn+kn)m^2)$	$m+km/logn$	$O(hn+k_1m^{'}vn^2+k_2n)$	$O(mn)$
Bagging	N/A	N/A	$O(hn+k_1mvn^2+k_2n)$	$O(n)$
REMT	N/A	N/A	$O(k_1mvn^2+k_2n)$	$O(n)$
FMKNN	N/A	N/A	$O(mn)$	$O(mn)$

1. m, n represent the numbers of features and samples of raw data, respectively. m' represents the feature number after feature selection.
2. In BFMDT, l denotes the cardinality of OFS in Algorithm 3, which is maximum of 50.
3. In FFREMT, V is the number of nodes in reduction algorithms [16].
4. In FCMT, t represents the number of decision classes.
5. In FPRS, k is the number of samples in the computation domain for each sample.
6. For decision trees, k_1, k_2 are the number of non-leaf nodes and leaf nodes respectively. v is the number of possible split points taken in the value domain of features. h is the number of decision trees.
7. The Time Complexity Ratio represents the ratio of time complexities of FFREMT, FCMT and FPRS divide BFMDT.

TABLE 6: Comparison on CA (%)

ID	BFMDT*	FFREMT*	FCMT*	FPRS_up*	FPRS_down*	FPRS_global*	Bagging	REMT	FMKNN
1	95.85+0.825	95.89+0.8941	94.67+4.472	95.02+1.3	95.17+1.51	95.32+1.324	95.22+1.579	95.02+2.175	91.35+3.737
2	**92.67+5.166**	68.53+5.495	65.75+1.983	68.48+11.94	67.34+9.994	67.34+9.994	72.41+10.13	68.48+6.763	48.89+4.099
3	**77.24+2.198**	69.33+4.079	61.78+7.426	64.99+7.347	64.29+7.413	65.36+8.039	67.88+4.868	51.96+9.256	74.02+2.237
4	71.11+5.003	**78.52+1.014**	67.15+5.918	75.19+4.829	74.81+5.493	75.19+4.829	77.41+6.058	77.41+4.793	74.81+6.495
5	83.86+2.603	75.31+15.92	83.77+3.183	80.96+10.58	75.94+15.7	78.45+12.93	77.9+13.17	78.13+12.98	83.86+2.603
6	**70.0+0.0**	63.6+3.11	53.6+4.219	55.8+1.891	56.4+2.104	56.2+1.891	62.0+3.142	58.8+4.907	54.1+2.302
7	**63.7+2.241**	55.92+0.8877	55.4+4.053	54.25+4.371	54.25+4.371	54.25+4.371	60.29+0.9972	56.28+5.89	25.06+0.2234
8	**94.55+1.585**	93.32+1.602	92.38+4.079	92.97+2.71	92.79+2.589	92.97+2.71	**94.73+1.636**	94.2+2.822	74.02+7.884
9	66.27+2.155	73.84+2.752	68.09+2.987	69.79+4.26	69.54+5.557	69.02+4.45	73.83+3.715	71.48+2.048	53.9+3.566
10	**62.52+1.289**	59.54+1.226	58.37+1.527	59.13+1.889	59.29+2.062	59.17+2.111	60.19+0.8406	57.74+1.516	5.104+1.537
Avg	**77.77**	73.38	70.10	71.66	70.98	71.33	74.19	70.95	58.51
11	99.43+1.278	/	97.06+3.176	97.02+2.986	95.24+4.017	96.43+3.931	96.45+4.83	95.29+2.631	99.39+1.355
12	**75.49+3.846**	72.1+9.334	-	71.67+5.218	75.02+2.513	72.64+5.261	73.83+6.222	71.64+5.478	72.14+6.385
13	84.3+0.2384	82.85+1.352	82.16+0.9696	/	/	/	84.53+1.18	76.55+2.887	30.03+0.5784
14	**47.42+10.9**	-	-	/	/	/	29.48+13.5	19.48+6.763	7.29+6.263
15	56.01+1.722	-	-	/	/	/	69.0+8.023	61.92+8.057	**83.5+7.216**
16	**71.12+4.392**	-	-	/	/	/	62.57+4.828	54.04+5.459	48.11+2.768
17	**93.57+2.102**	-	-	/	/	/	/	87.3+2.49	70.98+13.6
18	**99.55**	-	-	/	/	/	/	91.11	14.09+0.491
Avg_all		78.13||74.06	80.12||73.35	79.39||74.13	79.39||73.34	79.39||73.53	75.72||72.73	78.03||70.45	78.03||56.15
win-tie-lose	11-4-3	1-2-15	0-1-17	0-0-18	0-0-18	0-0-18	1-2-15	0-0-18	1-2-15

1. A "tie" is noted if two algorithms' accuracy differs by less than 0.3% on a dataset.
2. "win" and "tie" algorithms are highlighted or underlined for each dataset, respectively.
3. The * indicates algorithms with feature selection.
4. 3. The b in $a||b$ represents the average performance matrices of the comparison algorithm on all datasets that it can handle, and a represents the average performance matrices of BFMDT on the same datasets with the comparison algorithm.
5. The footnote 2-4 also applies to Table 7.

the dataset, the FMKNN algorithm performed poorly, indicating that the BFMDT algorithm maintains a high level of robustness when dealing with high-dimensional data. The above analysis illustrates that the BFMDT algorithm exhibits exceptional performance, especially in the context of large-scale data, where it demonstrates a superior enhancement in calculation scale.

5.4.3 Analysis on the number of feature selected

Table 9 presents the number of features selected by various algorithms on the datasets where they can perform feature selection. Considering that these algorithms ultimately yield multiple subsets of features, we have represented the results using the average number of features.

It is evident that the BFMDT algorithm produces feature subsets with a notably lower average number of features across various datasets, and has consistently demonstrated superior classification performance on each dataset. Furthermore, the feature selection processes of the FCMT and FFREMT algorithms appear to be ineffective on some datasets. These results underscore the applicability and efficacy of the BFMDT algorithm in the task of feature selection.

5.4.4 Analysis on time cost

The time complexity of the BFMDT algorithm can be divided into two components: the time overhead associated with feature selection and the time overhead involved in constructing the trees. Two decision tree algorithms, Bagging and REMT, and FMKNN do not necessitate any feature selection time overhead.

Fig. 6 and Table 12-14 present the total feature selection time and total classification time of five-fold cross-validation on different datasets. Classification methods employing feature selection tasks typically required longer execution times. Among all comparison algorithms, FMKNN had the shortest execution time. In tree-based algorithms,

TABLE 7: Comparison on MAE

ID	BFMDT*	FFREMT*	FCMT*	FPRS_up*	FPRS_down*	FPRS_global*	Bagging	REMT	FMKNN
1	4.15+0.825	**4.11+0.8941**	5.333+4.472	4.976+1.3	4.83+1.51	4.684+1.324	4.78+1.579	4.98+2.175	8.647+3.737
2	**9.603+7.428**	47.71+6.181	54.14+1.672	37.09+12.64	37.66+11.52	37.66+11.52	30.98+12.11	36.54+5.741	51.11+4.099
3	**22.76+2.198**	30.67+4.079	38.22+7.426	35.01+7.347	35.71+7.413	34.64+8.039	32.12+4.868	48.04+9.256	25.98+2.237
4	28.89+5.003	**21.48+1.014**	32.85+5.918	24.81+4.829	25.19+5.493	24.81+4.829	22.59+6.058	22.59+4.793	25.19+6.495
5	**16.14+2.603**	24.69+15.92	16.23+3.183	19.04+10.58	24.06+15.7	21.55+12.93	22.1+13.17	21.87+12.98	**16.14+2.603**
6	**30.0+0.0**	36.4+3.11	46.4+4.219	44.2+1.891	43.6+2.104	43.8+1.891	38.0+3.142	41.2+4.907	45.9+2.302
7	55.2+3.16	70.94+3.414	**44.6+4.053**	70.57+5.95	70.57+5.95	70.57+5.95	60.87+2.294	64.27+10.04	147.9+0.4809
8	5.45+1.585	6.679+1.602	7.62+4.079	7.032+2.71	7.206+2.589	7.032+2.71	**5.271+1.636**	5.801+2.822	25.98+7.884
9	33.73+2.155	**26.16+2.752**	31.91+2.987	30.21+4.26	30.46+5.557	30.98+4.45	26.17+3.715	28.52+2.048	46.1+3.566
10	46.0+2.049	56.72+1.085	53.43+1.902	51.78+2.007	51.51+2.219	51.73+2.315	**45.28+1.003**	54.41+2.087	211.4+13.93
Avg	**25.20**	32.56	33.07	32.47	33.08	32.75	28.82	32.82	60.43
11	**0.5714+1.278**	/	2.919+3.176	2.977+2.986	4.759+4.017	3.566+3.931	3.547+4.83	4.707+2.631	0.6061+1.355
12	**24.51+3.846**	27.9+9.334	-	28.33+5.218	24.98+2.513	27.36+5.261	26.17+6.222	28.36+5.478	27.86+6.385
13	22.75+0.767	22.04+4.491	24.37+2.459	/	/	/	**20.16+1.542**	32.87+2.925	124.7+2.106
14	**285.4+43.96**	-	-	/	/	/	477.2+105.4	459.5+78.9	517.0+114.7
15	43.99+1.722	-	-	/	/	/	31.0+8.023	38.08+8.057	**16.5+7.216**
16	**28.88+4.392**	-	-	/	/	/	37.43+4.828	45.96+5.459	51.89+2.768
17	**6.429+2.102**	-	-	/	/	/	/	13.86+2.838	29.18+13.83
18	**0.4545**	-	-	/	/	/	/	11.11	304.3+1.742
Avg_all		24.94||31.29	22.94||29.84	23.09||29.67	23.09||30.04	23.09||29.87	41.13||53.23	36.94||53.48	36.94||93.13
win-lose	10-8	3-15	1-18	0-18	0-18	0-18	3-15	0-18	2-16

TABLE 8: Comparison of average CA and MAE

ID	1-10		11-18	
Algorithm	CA (%)	MAE (%)	CA (%)	MAE (%)
FFREMT	4.39	7.36	4.06	6.35
FCMT	7.67	7.87	6.77	6.89
FPRS_up	6.11	7.27	5.61	6.58
FPRS_down	6.79	7.88	6.05	6.96
FPRS_global	6.44	7.55	5.86	6.78
Bagging	3.58	3.62	4.41	16.14
REMT	6.82	7.62	7.65	16.54
FMKNN	19.26	35.23	21.89	56.19

TABLE 9: The average number of feature subsets

ID	BFMDT	FFREMT	FCMT	up	down	global
1	2	8	8	8.1	8.2	8.2
2	2	10.4	11	12	12.1	12.1
3	1.5	7.7	9	8.1	8.2	8.1
4	1	11.4	11	12.1	12.1	12.1
5	1	9.7	9	10.1	11.5	15.1
6	2	18.6	20	19.1	19.1	19.1
7	1.9	15.8	13.7	17.1	17.1	17.1
8	5	19.9	5.8	29	29	29
9	3	7	8	7.2	7.2	7.2
10	8	10	11	10.1	10.1	10.1
11	1.8	/	5.8	11.7	9	15.8
12	16	42.6	-	54.3	49	57
13	7	30.1	32	/	/	/
14	1	-	-	/	/	/
15	1.2	-	-	/	/	/
16	1.6	-	-	/	/	/
17	13	-	-	/	/	/
18	2.8	-	-	/	/	/

REMT had the shortest execution time, as it did not involve any feature selection or ensemble learning process. Even if BFMDT adopted both feature selection and ensemble learning processes, it exhibited efficiency advantage among all algorithms.

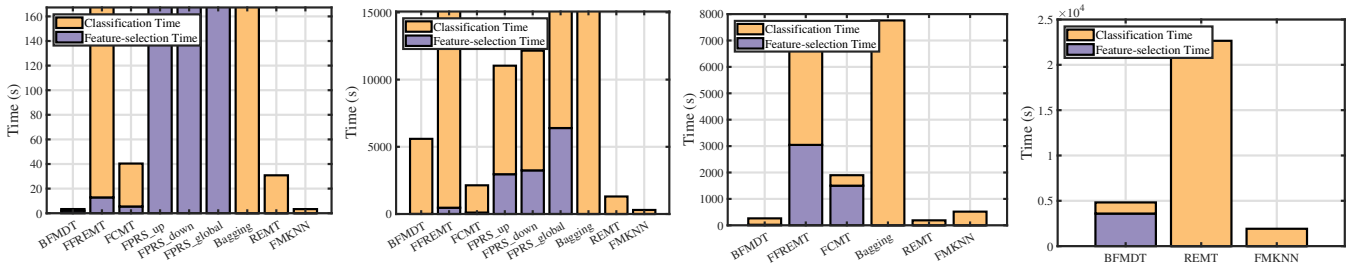
In step of feature selection, for all datasets (numbered 1-18), BFMDT achieved the shortest execution time, and the gap in computational efficiency was growing much larger as the data scale increased. For instance, BFMDT accelerated the feature selection by 146 times compared with FFREMT on dataset 13; 20952 times compared with FCMT on dataset 11; 1590 times compared with FPRS_up on dataset 12; 1399.95 times compared with FPRS_down on dataset 10; and 3482 times compared with FPRS_global on dataset 12. It is due to the lowest time complexity of BFMDT.

Among all the algorithms tested, FMKNN exhibited the shortest overall execution time, a characteristic attributed to its computational principle. However, when faced with high-dimensional data, the computation time of FMKNN significantly increased. Compared to tree-based algorithms, in step of tree-building, for small scale datasets 1-9, BFMDT achieved the shortest execution time on 3 datasets among all 8 classification algorithms. For 9 larger scale datasets (numbered 10-18), BFMDT achieved the shortest execution time on 4 datasets. In addition, BFMDT achieved the best average computation efficiency. It shows the same pattern as the feature selection step, as the scale increased, the BFMDT algorithm's advantages in terms of processable data scale and execution time become even more pronounced. For instance, BFMDT accelerated the tree-building by 687 times compared with FFREMT on dataset 6; 429 times compared with FCMT on dataset 11; 28 times compared with FPRS_up on dataset 8; 34 times compared with FPRS_down on dataset 8; 34 times compared with FPRS_global on dataset 8; 291 times compared with Bagging on dataset 15; and 28 times compared with REMT on dataset 17. This is primarily due to the fact that significant feature reduction substantially accelerates the tree-building process.

It's worth noting that most comparison algorithms are unable to process all datasets due to memory or time constraints, whereas BFMDT efficiently computes all datasets. The last column in Table 10 shows the percentage of time savings of the BFMDT algorithm compared to other algorithms. Compared to FMKNN, the BFMDT algorithm took 4.33 times longer to compute. However, as indicated by the CA and MAE experimental results, the efficiency of FMKNN comes at the cost of some loss in algorithm robustness. The aforementioned analysis demonstrates that BFMDT is capable of performing feature selection for high-dimensional dataset, effectively reducing feature dimensions and enhancing classification performance and com-

TABLE 10: Time cost of feature selection and classification (in seconds)

Datasets	1-10	11	12	13	14-16	17	18	decreased by (%)
BFMDT	5990.93	23.89	133.37	262.89	3255.38	688.18	4820.32	
FFREMT	31010.26	/	134.28	7362.95	-	-	-	83.41
FCMT	165692.58	41028.36	-	1897.79	-	-	-	96.99
FPRS_up	21440.66	190.26	3604.34	/	/	/	/	75.64
FPRS_down	24047.84	114.17	3146.74	/	/	/	/	77.49
FPRS_global	31913.29	485.40	7659.15	/	/	/	/	84.65
Bagging	41146.29	4.02	144.80	7759.91	38308.04	/	/	88.94
REMT	1480.98	2.16	11.95	188.01	3951.46	13180.89	22652.18	63.41
FMKNN	311.04	0.06	0.13	517.68	38.16	63.59	1914.09	-



(a) german-credit (b) wine-quality (c) turkiye-student (d) PEMS-SF

Fig. 5: Time of feature selections and classifications

putation efficiency.

6 CONCLUSION AND FUTURE WORK

In this paper, we introduce a novel feature selection method for monotonic datasets and enhance ordinal classification through the fusion of multiple feature subsets. The proposed algorithm improves average classification accuracy by 3.36%-7.65% and reduces average execution time by 63.41%-96.99%, showing efficiency in handling large-scale data. While there is room for improvement in dealing with continuous features and handling large sample sizes, future research will address these limitations.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (No.2021ZD0112400), the Key Program of the National Natural Science Foundation of China (No.62136005), National Natural Science Foundation of China (No.11201490, 62102428 and 61976089), the Hunan Provincial Natural Science Foundation of China (No.2021JJ20037), the Training Program for Excellent Young Innovators of Changsha (No.kq1905031).

REFERENCES

[1] Y. W. Kerk, K. M. Tay, and C. P. Lim, "Monotone Interval Fuzzy Inference Systems," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 11, pp. 2255–2264, 2019.

[2] T. B. Iwinski, "Ordinal Information Systems, I," *Bulletin of The Polish Academy of Sciences Mathematics*, vol. 36, pp. 467–475, 1988.

[3] ——, "Ordinal Information Systems, II," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 39, pp. 157–170, 1991.

[4] Q. H. Hu, W. W. Pan, L. Zhang, D. Zhang, Y. P. Song, M. Z. Guo, and D. R. Yu, "Feature Selection for Monotonic Classification," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 69–81, 2012.

[5] A. Ben-David, L. Sterling, and T. D. Tran, "Adding Monotonicity to Learning Algorithms May Impair Their Accuracy," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6627–6634, 2009.

[6] Q. H. Hu, X. J. Che, L. Zhang, D. Zhang, M. Z. Guo, and D. R. Yu, "Rank Entropy-Based Decision Trees for Monotonic Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 11, pp. 2052–2064, 2012.

[7] O. Sagi and L. Rokach, "Approximating XGBoost with an Interpretable Decision Tree," *Information Sciences*, vol. 572, pp. 522–542, 2021.

[8] P. Zhou, S. Zhao, Y. T. Yan, and X. D. Wu, "Online Scalable Streaming Feature Selection Via Dynamic Decision," *ACM Transactions on Knowledge Discovery from Data*, vol. 16, no. 5, pp. 1–20, 2022.

[9] P. M. Granitto, P. F. Verdes, and H. A. Ceccatto, "Neural Network Ensembles: Evaluation of Aggregation Algorithms," *Artificial Intelligence*, vol. 163, no. 2, pp. 139–162, 2005.

[10] H. Daniels and M. Velikova, "Monotone and Partially Monotone Neural Networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 6, pp. 906–917, 2010.

[11] H. Zhu, X. Z. Wang, and R. Wang, "Fuzzy Monotonic K-Nearest Neighbor Versus Monotonic Fuzzy K-Nearest Neighbor," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 9, pp. 3501–3513, 2022.

[12] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[13] P. Bartlett, Y. Freund, W. S. Lee, and R. E. Schapire, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.

[14] Y. H. Qian, H. Xu, J. Y. Liang, B. Liu, and J. T. Wang, "Fusing Monotonic Decision Trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 10, pp. 2717–2728, 2015.

[15] H. F. Zhou, J. W. Zhang, Y. Q. Zhou, X. J. Guo, and Y. M. Ma, "A Feature Selection Algorithm of Decision Tree Based on Feature Weight," *Expert Systems with Applications*, vol. 164, p. 113842, 2021.

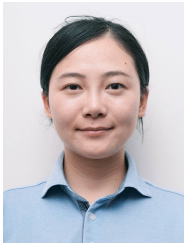
[16] J. T. Wang, Y. H. Qian, F. J. Li, J. Y. Liang, and W. P. Ding, "Fusing Fuzzy Monotonic Decision Trees," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 887–900, 2020.

[17] Z. Pawlak, "Rough Sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[18] Q. H. Hu, M. Z. Guo, D. R. Yu, and J. F. Liu, "Information Entropy for Ordinal Classification," *Science China Information Sciences*, vol. 53, no. 6, pp. 1188–1200, 2010.

[19] H. Xu, W. J. Wang, and Y. H. Qian, "Fusing Complete Monotonic

Decision Trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2223–2235, 2017.
[20] Q. H. Hu, D. R. Yu, Z. X. Xie, and X. D. Li, "EROS: Ensemble Rough Subspaces," *Pattern Recognition*, vol. 40, no. 12, pp. 3728–3739, 2007.
[21] S. Y. Yang, H. Y. Zhang, B. De Baets, M. Jah, and G. Shi, "Quantitative Dominance-Based Neighborhood Rough Sets Via Fuzzy Preference Relations," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 3, pp. 515–529, 2021.
[22] F. Nosheen, U. Qamar, and M. S. Raza, "A Parallel Rule-Based Approach to Compute Rough Approximations of Dominance Based Rough Set Theory," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105285, 2022.
[23] ——, "Redefining Preliminaries of Dominance-Based Rough Set Approach," *Soft Computing*, vol. 26, no. 3, pp. 977–1002, 2022.
[24] W. T. Li, X. P. Xue, W. H. Xu, T. Zhan, and B. J. Fan, "Double-Quantitative Variable Consistency Dominance-Based Rough Set Approach," *International Journal of Approximate Reasoning*, vol. 124, pp. 1–26, 2020.
[25] S. Singh, S. Shreevastava, T. Som, and G. Somani, "A Fuzzy Similarity-Based Rough Set Approach for Attribute Selection in Set-Valued Information Systems," *Soft Computing*, vol. 24, no. 6, pp. 4675–4691, 2020.
[26] W. T. Li, H. X. Zhou, W. H. Xu, X. Z. Wang, and W. Pedrycz, "Interval Dominance-Based Feature Selection for Interval-Valued Ordered Data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 6898–6912, 2023.
[27] Y. H. Qian, J. Y. Liang, and C. Y. Dang, "Interval Ordered Information Systems," *Computers & Mathematics with Applications*, vol. 56, no. 8, pp. 1994–2009, 2008.
[28] B. B. Sang, H. M. Chen, L. Yang, T. R. Li, W. H. Xu, and C. Luo, "Feature Selection for Dynamic Interval-Valued Ordered Data Based on Fuzzy Dominance Neighborhood Rough Set," *Knowledge-Based Systems*, vol. 227, p. 107223, 2021.
[29] Q. Q. Huang, T. R. Li, Y. Y. Huang, X. Yang, and H. Fujita, "Dynamic Dominance Rough Set Approach for Processing Composite Ordered Data," *Knowledge-Based Systems*, vol. 187, p. 104829, 2020.
[30] S. Greco, B. Matarazzo, and R. Slowinski, "A New Rough Set Approach to Multicriteria and Multiattribute Classification," in *Rough Sets and Current Trends in Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 60–67.
[31] P. A. Gutiérrez and S. García, "Current Prospects on Ordinal and Monotonic Classification," *Progress in Artificial Intelligence*, vol. 5, no. 3, pp. 171–179, 2016.
[32] B. B. Sang, H. M. Chen, L. Yang, J. H. Wan, T. R. Li, and W. H. Xu, "Feature Selection Considering Multiple Correlations Based on Soft Fuzzy Dominance Rough Sets for Monotonic Classification," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 12, pp. 5181–5195, 2022.
[33] Q. H. Hu, D. R. Yu, and M. Z. Guo, "Fuzzy Preference Based Rough Sets," *Special Issue on Intelligent Distributed Information Systems*, vol. 180, no. 10, pp. 2003–2022, 2010.
[34] T. Yang, Q. G. Li, and B. L. Zhou, "Related Family: A New Method for Attribute Reduction of Covering Information Systems," *Information Sciences*, vol. 228, pp. 175–191, 2013.
[35] T. Yang, Y. F. Deng, B. Yu, Y. H. Qian, and J. H. Dai, "Local Feature Selection for Large-scale Data Sets Limited Labels," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 7152–7163, 2023.
[36] T. Yang, X. R. Zhong, G. M. Lang, Y. H. Qian, and J. H. Dai, "Granular Matrix: A New Approach for Granular Structure Reduction and Redundancy Evaluation," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3133–3144, 2020.
[37] T. Yang, Y. J. Li, Y. H. Qian, and F. Y. Wang, "Consistent Matrix: A Feature Selection Framework for Large-Scale Data Sets," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 11, pp. 4024–4038, 2023.
[38] M. Hu, E. C. Tsang, Y. T. Guo, D. G. Chen, and W. H. Xu, "A Novel Approach to Attribute Reduction Based on Weighted Neighborhood Rough Sets," *Knowledge-Based Systems*, vol. 220, p. 106908, 2021.
[39] Q. H. Hu, D. R. Yu, J. F. Liu, and C. X. Wu, "Neighborhood Rough Set Based Heterogeneous Feature Subset Selection," *Information Sciences*, vol. 178, no. 18, pp. 3577–3594, 2008.
[40] X. L. Yang, H. M. Chen, T. R. Li, J. H. Wan, and B. Sang, "Neighborhood Rough Sets with Distance Metric Learning for Feature Selection," *Knowledge-Based Systems*, vol. 224, p. 107076, 2021.
[41] W. T. Li, W. H. Xu, X. Y. Zhang, and J. Zhang, "Updating Approximations with Dynamic Objects Based on Local Multigranulation Rough Sets in Ordered Information Systems," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 1821–1855, 2022.
[42] X. Zhang, C. L. Mei, D. G. Chen, Y. Y. Yang, and J. H. Li, "Active Incremental Feature Selection Using a Fuzzy-Rough-Set-Based Information Entropy," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 901–915, 2020.
[43] D. Dubois, H. Fargier, and P. Perny, "Corrigendum to "Qualitative Decision Theory with Preference Relations and Comparative Uncertainty: An Axiomatic Approach"," *Artificial Intelligence*, vol. 171, no. 5-6, pp. 361–362, 2007.
[44] G. M. Lang, M. J. Cai, H. Fujita, and Q. M. Xiao, "Related Families-Based Attribute Reduction of Dynamic Covering Decision Information Systems," *Knowledge-Based Systems*, vol. 162, pp. 161–173, 2018.
[45] D. G. Chen, S. Y. Zhao, L. Zhang, Y. P. Yang, and X. Zhang, "Sample Pair Selection for Attribute Reduction with Rough Set," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 11, pp. 2080–2093, 2012.
[46] A. Skowron and C. Rauszer, "The Discernibility Matrices and Functions in Information Systems," in *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*. Dordrecht: Springer Netherlands, 1992, pp. 331–362.
[47] Z. Pawlak and A. Skowron, "Rough Sets and Boolean Reasoning," *Zdzislaw Pawlak life and work (1926–2006)*, vol. 177, no. 1, pp. 41–73, 2007.
[48] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann, 1993.



Tian Yang received the Ph.D. degree in applied mathematics from Hunan University, Changsha, China. She is an Associate Professor at Hunan Normal University, Changsha, China.
Her current research areas include granular computing, intelligent information processing, fuzzy systems, data mining and topology.

Fansong Yan is currently pursuing the Master Degree with the College of Information Science and Engineering, Hunan Normal University, Changsha, China.
His main research interests include machine learning, granular computing and data mining .

Jieting Wang received the MS and PhD degrees in computers with applications from Shanxi University, Taiyuan, China, in 2015 and 2021, respectively.
She is currently a teacher with the Institute of Big Data Science and Industry, Shanxi University. Her research interest includes statistical machine learning and ensemble learning.

Yuhua Qian (Member, IEEE) received the M.S. and Ph.D. degrees in computers with applications from Shanxi University, Taiyuan, China, in 2005 and 2011, respectively. He is currently a Director at the Institute of Big Data Science and Industry, Shanxi University, where he is also a Professor with the Key Laboratory of Computational Intelligence and Chinese Information Processing, Ministry of Education. He is best known for artificial intelligence, machine learning and machine vision. He has published more than 120 papers in his research fields, including the journals of Artificial Intelligence, the Journal of Machine Learning Research, IEEE Transactions on Pattern Analysis and Machine Intelligence, and so on. Prof. Qian served on the Editorial Board of the International Journal of Knowledge-Based Organizations and Artificial Intelligence Research, the Program Chair or the Special Issue Chair for the Conference on Rough Sets and Knowledge Technology, the Joint Rough Set Symposium, and the Conference on Industrial Instrumentation and Control, and a PC member for many machine learning, data mining, and granular computing conferences.