Knowledge-Based Systems 309 (2025) 112781

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Enhancing information fusion and feature selection efficiency via the PROMETHEE method for multi-source dynamic decision data sets

Weihua Xu *, Yigao Li

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

ARTICLE INFO

Keywords:
Feature selection
Information fusion
Multi-source decision information system
PROMETHEE
Dynamic method

ABSTRACT

With the surge in big data, the complexity of synthesizing information from multiple sources has become a critical challenge for feature selection methodologies. Feature selection is the process of reducing the number of attributes in data. Traditional single-source centric approaches are inefficient, requiring extensive preprocessing for multi-source data consolidation prior to feature selection. At the same time, an information fusion method is needed to transform the multi-source information system with selected features into a single-source information system. This paper introduces a novel multi-source information fusion and feature selection approach that seamlessly integrates the Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) with a dynamic adaptation mechanism. This method is adept at addressing the complexities introduced by the evolving nature of feature and information source dimensions. The Attribute Evaluation Matrix (AEM) and the Attribute Preference Degree Matrix (APDM) are proposed to systematically assess and rank the significance of attributes within a static decision-making framework. Following this, an information fusion method using the source center is proposed. The dynamic feature selection and information fusion methods are proposed to deal with the condition when number of attributes and samples change. Extensive experimental validation confirms that this method not only reduces the computational overhead associated with multi-source feature selection but also significantly enhances the efficiency as the volume and variety of data sources increase.

1. Introduction

As the information age progresses, we are continuously confronted with an overwhelming influx of diverse data types. The imperative to efficiently distill pivotal features from a multitude of attributes is paramount, thereby enhancing the efficacy of the information acquisition process. The process of reducing the number of attributes in data is called feature selection. Feature selection methodologies have become pivotal in a spectrum of critical domains, including but not limited to: medical diagnostics [1–3], fraud detection [4], text feature selection [5] and credit scoring [6]. A compendium of additional applications can be found in [7,8].

The field of feature selection has witnessed extensive scholarly inquiry, with a plethora of research achievements summarized herein. Pioneering studies have proposed feature selection techniques tailored for single-source, single-value information systems [9–14], as well as for single-source interval-valued [15–17], multi-scale [18–20], and incomplete information systems [21–24]. Beyond these static feature extraction methodologies, a burgeoning body of research has emerged on dynamic feature selection approaches [25–27]. For an exhaustive exploration of feature selection methods, the reader is directed to [28–30], which present a diverse array of techniques catering to various preferences and requirements.

For single-source single-value information systems, Hu [9] invented a method combining Overlap Degree (OD) & K-Nearest-Neighbor (KNN) for feature selection based on rough set theory, Li [10] presented a novel binary relation based on improved Fuzzy C-Means (FCM) instead of equivalence relation. Khodadadi [12] introduces a binary variant of the Arithmetic Optimization Algorithm (BAOA) designed to address the feature selection challenge in classification. Abdelhamid [14] present a new hybrid binary meta-heuristic algorithm for solving the feature selection problem by integrating two algorithms: Dipper Throated Optimization (DTO) and the Sine Cosine (SC) algorithm. Ganesh [13] presents an innovative K-nearest neighbor (KNN)-based wrapper system for feature selection, which utilizes the iterative enhancement capability of the weighted superposition attraction (WSA). Alhussan [11] introduces a new algorithm inspired by the hunting behavior of waterwheel plants and how they adjust their locations during exploration and exploitation processes. For single-source interval value information systems, Xu's graph-based method [15] considered the left and right

* Corresponding author.
E-mail addresses: chxuw@swu.edu.cn (W. Xu), lyg040826@email.swu.edu.cn (Y. Li).

https://doi.org/10.1016/j.knosys.2024.112781
Received 9 July 2024; Received in revised form 23 October 2024; Accepted 21 November 2024
Available online 28 November 2024

boundaries of interval values when defining the distance between two samples. Each feature was used as a point in the graph to establish the graph, and the features were sorted through the graph. Dai [17] adopted a method utilizing kernel density estimation specifically designed for interval-valued data. In [16], algorithm for deriving the interval-valued dominance relation and the feature selection method was established by interval-valued ordered decision system (IV-ODS). For single-source multi-scale information system, Huang [19] made a new data analysis model with multi-scale coverings by extending partitions to coverings. For single-source incomplete information systems, Shen [21] explored a novel Unsupervised Feature Selection (UFS) method for performing UFS on incomplete data sets to address the previously mentioned issues. Maghsoodi [22] introduces a machine learning-based DDDM approach to address LSDM problems involving incomplete data and numerous decision attributes. Sun [23] introduced a new feature selection approach that utilizes neighborhood rough sets, incorporating Lebesgue and entropy measures within incomplete neighborhood decision systems. For dynamic feature selection methods, Fahy [25] introduced a dynamic feature mask for clustering high-dimensional data streams that updates in real-time by masking redundant features and unmasking relevant ones as their importance changes. Shu [27] focused on incrementally updating and selecting a new feature subset as multiple objects undergo variation. These articles provide some inspiration for our research.

Most of these previous studies have provided effective solutions for single-source information systems in different forms. However, in today's era, where information sources are increasingly diverse, these single-source feature selection methods are evidently inadequate for handling multiple information sources. For multi-source decision information systems, there are some articles that study the information fusion of these systems, such as: [31–36]. The process of information fusion is combining the same type of information from multiple sources into a single source of information. Xu [31] used statistical distribution principles and KL divergence to create a metric for evaluating the similarity between intervals. Zhang [32] introduced a new information fusion method based on information entropy for multi-source incomplete interval-valued data, along with four incremental fusion mechanisms that account for changes in information sources and attributes. Xiao [36] introduces a new MSIF method for decision-making, based on a newly defined generalized evidential divergence measure across multiple sources of evidence. If you want to learn more about articles on information fusion, Zhang [34] has summarized the existing information fusion methods based on rough set theory for your reference. These articles address some of the issues related to information fusion in multi-source information systems. At present, there are few feature selection methods for multi-source, and it is troublesome to solve the problem of multi-source feature selection. One approach is to first use an information fusion method to convert the multi-source information system into a single-source system, and then apply feature selection methods to the single-source system. The feature selection result from this method is based on the data constructed through information fusion, rather than the original data, which means the selected features may not represent the important features in the original information system. This is not conducive to subsequent information collection based on the feature selection results. The other approach involves applying feature selection methods to each source within the multi-source information system. The feature selection results from each source are then combined using methods such as intersection or union to obtain the final feature selection result for the multi-source system. This approach ensures that the feature selection results are based on the original data, but it requires a significant amount of time. In this article, we first propose a multi-source feature selection method that can handle multiple information sources. In contrast to the first approach, our method allows us to obtain feature selection results based on the original data, ensuring that the selected features are important within the original dataset. Additionally, compared to the second approach mentioned above, our method saves a significant amount of time while achieving feature selection results that can surpass those of the second approach. After feature selection, information fusion can be carried out based on the multi-source decision information system with selected features. So we also proposed an information fusion method that can integrate the results after feature selection. However, this information fusion and feature selection method is static, meaning it requires re-computation when information sources change, which also consumes considerable time. In the information age, the forms and sources of information can change at any time, rendering static methods inadequate. Therefore, in this article, we also propose dynamic methods that allow our feature selection approach to adapt to changes in information sources. The dynamic methods tend to deal with the condition when number of attributes and samples change. The main contributions of this article can be summarized as follows:

• We defined the Attribute Evaluation Matrix (AEM) to evaluate the various features of each information source, thereby determining the impact of each feature of each information source on the classification effectiveness.

• To comprehensively consider the impact of different features from different information sources on classification effectiveness, we defined the Attribute Preference Degree Matrix (APDM). The APDM compares each feature with other features and evaluates the superiority or inferiority of each feature relative to others in terms of their performance in classification.

• We also defined the leaving flow, the entering flow, and the net flow to provide a comprehensive evaluation of attributes. Finally, based on the net flow values of attributes, we sorted the attributes, making it convenient to extract different quantities of attributes according to specific needs.

• To integrate the multi-source information system with selected features into a single-source information system, we proposed an information fusion method that selects the optimal information source through the source center.

• In order to make our method adaptable to changes in information sources, we developed a dynamic method that considers variations in both the number of information sources and the number of attributes. The dynamic feature selection method is divided into four scenarios to cover all possible variations in information sources.

The article is organized into several sections. In the introduction, we present the background and motivation for the study. The preliminaries section covers the concepts of the multi-source information system, rough set theory, and the Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) method. In Section 3, Our research findings are divided into five parts: defining the AEM, introducing the APDM and the net flow, proposing the information fusion method, explaining the steps of our static information fusion and feature selection method, and providing a small case study as an illustration. The dynamic information fusion and feature selection method is detailed in Section 4, where we present it in four scenarios with accompanying code for clarity. The experimental section discusses our experimental setup, presents the results of the experiments, and analyzes the effectiveness of our methods. In the conclusion and future work section, we summarize our findings, highlight key points, and propose avenues for future research. Finally, the references section lists all the cited sources in the article.

2. Preliminaries

In this section, we will first briefly recall the concept of Decision Information System (DIS), Multi-Source Decision Information System (MSDIS), rough set theory and PROMETHEE. Then, we will explain the combinations between each preliminaries and their application in our study.

Table 1
An example of MSDIS.

	$f^1_{a_1}$	$f^1_{a_2}$	$f^1_{a_3}$	$f^1_{a_4}$	$f^2_{a_1}$	$f^2_{a_2}$	$f^2_{a_3}$	$f^2_{a_4}$	$f^3_{a_1}$	$f^3_{a_2}$	$f^3_{a_3}$	$f^3_{a_4}$	f_d
x_1	0.33	0.67	0.5	0.99	0.33	0.68	0.5	1	0.29	0.58	0.43	0.85	N
x_2	0.83	0.74	0	0.19	0.99	0.88	0	0.23	0.99	0.88	0	0.23	N
x_3	0.34	0	0.5	1.01	0.33	0	0.49	0.98	0.32	0	0.48	0.96	O
x_4	0	0.64	0	0.41	0	0.56	0	0.36	0	0.63	0	0.4	O
x_5	1.03	0.62	0.51	0.31	1.18	0.71	0.59	0.36	0.94	0.57	0.47	0.29	N
x_6	0	0.75	0	0.67	0	0.65	0	0.57	0	0.69	0	0.61	O
x_7	0.56	0.85	0.85	0	0.58	0.87	0.87	0	0.65	0.97	0.97	0	N
x_8	1.03	0.7	0.52	0.24	1.19	0.81	0.6	0.28	1.16	0.79	0.58	0.27	O
x_9	0	0.65	0.96	0.29	0	0.69	1.01	0.31	0	0.72	1.06	0.32	N
x_{10}	0.62	0.83	0.47	0.43	0.5	0.67	0.38	0.35	0.65	0.87	0.49	0.45	O
x_{11}	0.35	0	0.53	1.06	0.35	0	0.52	1.03	0.35	0	0.53	1.05	O

2.1. MSDIS and rough set theory

A decision information system can be denoted as DIS=$\{U, A, D, V_A, f_A, V_D, f_D\}$, where U is the set of samples; A represents a set of attributes these samples have; D represents a set of decision attribute these samples have; V_A is the domain of A; f_A is a function which can get the value of attribute $a \in A$ in sample $x \in U$. f_A can be represent as $f_A : U \times A = V_A$; V_D is the domain of D; f_D is a function which can get the value of decision attribute $d \in D$ in sample $x \in U$. f_D can be represented as $f_D : U \times D \to V_D$;

A multi-sources decision information system is a set of several DIS which can be denoted as MSDIS = $\{DIS_i | DIS_i = \{U, A, D, V_A, f_A, V_D, f_D\}, i = 1, 2, 3, \dots, n\}$. An example for MSDIS = $\{DIS_i | DIS_i = \{U, A, D, V_A, f_A, V_D, f_D\}, i = 1, 2, 3, \dots, n\}$ is shown in Table 1.

After we defined the DIS and MSDIS, we will have a brief review of rough set theory. Rough Set Theory, proposed by Polish mathematician Zdzisław Pawlak in the early 1980s, is a mathematical tool for handling uncertainty and vagueness. The theory is primarily used for analyzing data sets with incomplete or uncertain information. Before reviewing rough set theory, let us first revisit the basic theory of sets. With a DIS=$\{U, A, D, V_A, f_A, V_D, f_D\}$, we can get a relation between any two samples $a, b \in U$ using Cartesian product

$$R = U \times U = \{(x_i, x_j) | x_i \in U \cap x_j \in U\}. \tag{1}$$

If the relation R satisfies the condition mentioned below, it can be called equivalence relation.

- For any $x_i \in U$, there is $x_i R x_i$.
- If $x_i R x_j$, there is $x_j R x_i$.
- If $x_i R x_j$ and $x_j R x_k$, there is $x_i R x_k$.

With the definition of equivalence relation, we can get equivalence class of every samples $x \in U$ which can be denoted as follow

$$[x_i]_R = \{x_j | (x_i, x_j) \in R\}. \tag{2}$$

Because the relation is equivalent, we can get several quotient sets according to the equivalence class and these quotient sets will divide the samples into partitions which can be represented as U/R=$\{P_1, P_2, P_3, \dots\}$, $P_1, P_2, P_3, \dots$ are partitions we get using the equivalence relation. Samples within the same parts are considered equivalent.

After we recall the basic theory of sets, we can introduce rough set theory. According to the [37], when we randomly draw a subset $X \subseteq U$, if X can be represented as the union of several parts P_i, then X is considered a precise set. Otherwise, X is a rough set.

To represent the roughness of the set $X \subseteq U$, we define the upper approximation, the lower approximation, and the accuracy of approximation. For a subset of the sample set $X \in U$, a lower and upper approximation can be define as follow

$$\underline{R}(X) = \{x | [x]_R \subseteq X\} \tag{3}$$

$$\overline{R}(X) = \{x | [x]_R \cap X \neq \emptyset\}. \tag{4}$$

With $\underline{R}(X)$ and $\overline{R}(X)$, we can find that $\underline{R}(X) \subseteq X \subseteq \overline{R}(X)$. With the definition of lower and upper approximation, the approximate accuracy is denoted as follow

$$AP_R(X) = \frac{|\underline{R}(X)|}{|\overline{R}(X)|}, X \subseteq U, \tag{5}$$

where $|\underline{R}(X)|$ is the amount of units in lower approximation $\underline{R}(X)$, $|\overline{R}(X)|$ is the amount of units in upper approximation $\overline{R}(X)$. The higher approximate accuracy is, the less roughness set X have.

To assess the significance of each attribute in classification, Now we define two kinds of equivalence relations for the decision attribute set D and the attribute set A as follow

$$R_D = \{(x_i, x_j) | x_i \in U \cap x_j \in U \cap f_D(x_i, d) = f_D(x_j, d)\} \tag{6}$$

$$R_A(a) = \{(x_i, x_j) | x_i \in U \cap x_j \in U \cap |f_A(x_i, a) - f_A(x_j, a)| < \alpha\}, a \in A, \tag{7}$$

where α is a parameter to be defined. With the equivalence relations for the decision attribute set R_D, we can get partitions $U/R_D = \{Y_1, Y_2, Y_3, \dots\}$. Using the partitions derived from R_D, we then determine the approximation accuracy under the equivalence relations of the attribute set R_A as follow

$$AP_{R_A(a)}(Y_i) = \frac{|\underline{\underline{R_A}}(a)(Y_i)|}{|\overline{\overline{R_A}}(a)(Y_i)|}, a \in A, \tag{8}$$

where $|\underline{\underline{R_A}}(a)(Y_i)|$ is the amount of units in lower approximation $\underline{\underline{R_A}}(a)(Y_i)$, $|\overline{\overline{R_A}}(a)(Y_i)|$ is the amount of units in upper approximation $\overline{\overline{R_A}}(a)(Y_i)$. The higher approximate accuracy is, the better this attribute behave in classification. With $AP_{R_A(a)}(Y_i)$, we can find out the role of attributes in classification. Next, we will introduce PROMETHEE method to comprehensively evaluate and rank the attributes based on attributes' effectiveness on classification.

2.2. PROMETHEE

PROMETHEE is a decision-making method used in the field of multi-criteria decision analysis. It was developed by J.P. Brans and B. Mareschal in the late 1980s in the article [38]. This method is designed to help decision-makers evaluate and rank a set of alternative options based on multiple criteria. It considers both the positive and negative aspects of each alternative and provides a preference ranking.

Next, we will explain PROMETHEE method in details. PROMETHEE method need to input a decision matrix including alternatives and criteria. The purpose of this method is to derive a ranking from best to worst for each alternative based on all the criteria. Firstly, we need to normalize the decision matrix using the formula below

$$R_{ij} = \frac{x_{ij} - minx_{ij}}{maxx_{ij} - minx_{ij}} i = 1, 2, \dots, n, j = 1, 2, \dots, m, \tag{9}$$

where x_{ij} is the element in matrix, n represents the number of alternatives and m represents the number of criteria. Then we will calculate the preference function $P_j(i, i')$. This function represents the difference between alternative i and alternative i' under criteria j. Through the preference function of each criteria, we will calculate the aggregated preference function between alternative i and alternative i'.

$$\pi(i,i') = \frac{\sum_{j=1}^{m} \omega_j \times P_j(i,i')}{\sum_{j=1}^{m} \omega_j}. \tag{10}$$

With aggregated preference function between every two alternatives, we can calculate the leaving flow and entering flow as follow

$$\phi_{leaving}(i) = \frac{1}{n-1}\sum_{i'=1}^{n} \pi(i,i') \tag{11}$$

$$\phi_{entering}(i) = \frac{1}{n-1}\sum_{i'=1}^{n} \pi(i',i). \tag{12}$$

We can calculate the net flow by the leaving flow and entering flow as follow

$$\phi_{net}(i) = \phi_{leaving}(i) - \phi_{entering}(i). \tag{13}$$

The larger the net flow value is, the better the alternative is. We can rank all the alternatives from highest to lowest based on their net flow values, thereby forming the ranking of the alternatives.

PROMETHEE is devoted to solve the problem of multi-criteria decision analysis. Because the multi-criteria decision matrix is similar to the MSDIS, we can use this method to deal with the MSDIS and generate the result of sample's rank. However, there are still some differences between MSDIS and multi-criteria decision table, we will enhance PROMETHEE to make it applicable for MSDIS.

To provide readers with a deeper understanding of the preliminary concepts, we will next introduce the connections between each preliminary and how they contribute to our research. MSDIS is the subject of our research, where we will focus on feature selection and information fusion. Rough set theory is employed to evaluate the contribution of each attribute in MSDIS to classification. Specifically, by using the relationships R_D and $R_A(a)$, we derive the upper approximation sets $\overline{R_A(a)}(Y_i)$ and the lower approximation sets $\underline{R_A(a)}(Y_i)$ and calculate the approximate accuracy $AP_{R_A(a)}(Y_i)$, which allows us to assess the significance of each attribute in classification. The PROMETHEE method is then used to comprehensively consider the impact of each attribute across different information sources, resulting in a ranking of attribute importance, thus achieving effective feature selection.

3. Information fusion and feature selection based on PROMETHEE for MSDIS

In this section, we will introduce the AEM and the feature selection method based on PROMETHEE inspired by [39] and the information fusion method.

3.1. Attribute evaluation matrix

To comprehensively reflect the impact of various attributes on classification, we defined the AEM to evaluate every attribute in each DIS. A DIS=$\{U,A,D,V_A,f_A,V_D,f_D\}$ is given. To form an AEM, firstly, we need to calculate the $U/R_D = \{Y_1,Y_2,Y_3,\dots\}$ by the relation for the decision attribute set R_D defined in Eq. (6). Secondly, with U/R_D and the relation for the attribute set $R_A(a_i)$ defined in Eq. (7), we can calculate the lower approximation $\underline{R_A(a_i)}(Y_j)$ and upper approximation $\overline{R_A(a_i)}(Y_j)$ of every attributes separately according to Eq. (3) and Eq. (4). Thirdly, with $\underline{R_A(a_i)}(Y_j)$ and $\overline{R_A(a_i)}(Y_j)$, we can calculate the approximation accuracy $AP_{R_A(a_i)}(Y_j)$ by Eq. (8). Finally, using the approximation accuracy of every attributes a_i in every partitions Y_j, We can create an AEM just like in Table 2.

After defining the AEM, each DIS_i corresponds to a AEM_i. The number of rows in AEM corresponds to the number of attributes, while the number of columns corresponds to the number of elements in the set U/R_D. For a MSDIS=$\{DIS_i|DIS_i = \{U,A,D,V_A,f_A,V_D,f_D\}, i = 1,2,3,\dots,n\}$, there is a set of AEMs=$\{AEM_1, AEM_2,\dots,AEM_n\}$. Next, we propose a method based on PROMETHEE to integrate these AEM and derive a ranking of the attributes.

Table 2
Attribute evaluation matrix.

$AP_{R_A(a_i)}(Y_j)$	Y_1	Y_2	...	Y_n
a_1	$AP_{R_A(a_1)}(Y_1)$	$AP_{R_A(a_1)}(Y_2)$	...	$AP_{R_A(a_1)}(Y_n)$
a_2	$AP_{R_A(a_2)}(Y_1)$	$AP_{R_A(a_2)}(Y_2)$	...	$AP_{R_A(a_2)}(Y_n)$
...	...	...	...	...
a_n	$AP_{R_A(a_n)}(Y_1)$	$AP_{R_A(a_n)}(Y_2)$	...	$AP_{R_A(a_n)}(Y_n)$

Table 3
The AEM calculated in the example.

	Y_1^1	Y_2^1	Y_1^2	Y_2^2	Y_1^3	Y_2^3
a_1	0.09	0	0	0	0	0
a_2	0	0.18	0	0.18	0	0.15
a_3	0.18	0	0.18	0	0.18	0
a_4	0	0.07	0.1	0.1	0.125	0.36

Example 3.1. To better explain the creation of AEM, we provide an example for understanding purpose. The data for this example is taken from the fertility data set in UCI. We select 4 features and 11 objects from this data set, which are used to generate 3 sources of information. The data is shown on Table 1. In this example, we calculate AEM_1 corresponding to the first information source DIS_1 of the entire information system MSDIS in detail.

Firstly, get the U/D and U/R_A.the α is set to 0.2. The $U/D = \{\{x_1,x_2,x_5,x_7,x_9\},\{x_3,x_4,x_6,x_8,x_{10},x_{11}\}\}$. The $U/R_A(a_1)$ of DIS_1 is $U/R_A(a_1) = \{\{x_1,x_3,x_{11}\},\{x_2\},\{x_5,x_8\},\{x_4,x_6,x_9\},\{x_7,x_{10}\}\}$

Secondly, calculate the $AP_{R_A(a_i)}(Y_j)$ for every $a_i \in A$ in every $Y_j \in U/D$. For example, we calculate the $AP_{R_A(a_1)}(Y_1)$ in DIS_1, $\underline{R_A(a_1)}(Y_1) = \{x_2\}$, $\overline{R_A(a_1)}(Y_1) = \{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11}\}$, $AP_{R_A(a_1)}(Y_1) = \frac{|\underline{R_A(a_1)}(Y_1)|}{|\overline{R_A(a_1)}(Y_1)|} = \frac{1}{11} \approx 0.09$.

Finally, we can create the AEM_1 of the first information source DIS_1. We place the calculated $AP_{R_A(a_i)}(Y_j)$ in the corresponding positions according to the AEM structure given in Table 2, thereby forming the AEM_1. Using the same method, we create two additional AEMs through the other two DIS sources. All the AEMs we constructed can be combined to form a single set of AEMs=$\{AEM_1, AEM_2,\dots,AEM_n\}$. The AEMs we created is shown on Table 3.

3.2. Feature selection based on PROMETHEE

Given a MSDIS = $\{DIS_i|DIS_i = \{U,A,D,V_A,f_A,V_D,f_D\}, i = 1,2,3,\dots\}$, every DIS in MSDIS can produce an AEM. Firstly, we should define a preference function $P(x)$ as follow

$$P(x) : U \times U \to (0,1). \tag{14}$$

By using $P(x)$, we can constrain the range of x within $(0,1)$. According to the PROMETHEE method, we need to define the attribute preference degree as follow

$$APD_k^i(a,b) = P(AP_{R_A(a)}(Y_k) - AP_{R_A(b)}(Y_k)), a,b \in A, \tag{15}$$

where i represents the number of information source in the MSDIS. To comment the extent to which attribute $a \in A$ is better than $b \in A$ in the ith information source, the calculate formula is as follow

$$APD^i(a,b) = \sum_{k=1}^{n} w_k * APD_k^i(a,b), \tag{16}$$

where w_k is Y_k's weight. In order to comprehensively consider multiple information sources, the calculate formula is as follow

$$APD(a,b) = \sum_{i=1}^{n} APD^i(a,b). \tag{17}$$

$APD(a,b)$ reflect the extent to which attribute $a \in A$ is better than attribute $b \in A$ in classification. The higher $APD(a,b)$ is, the greater

Table 4
Attribute preference degree matrix.

$APD(a,b)$	a_1	a_2	...	a_n
a_1	0	$APD(a_1,a_2)$	...	$APD(a_1,a_n)$
a_2	$APD(a_2,a_1)$	0	...	$APD(a_2,a_n)$
...	...	...	...	...
a_n	$APD(a_n,a_1)$	$APD(a_n,a_2)$	...	0

Table 5
The APDM calculated in the example.

	a_1	a_2	a_3	a_4
a_1	0	0.25	0	0.25
a_2	1.21	0	1.21	0.52
a_3	1.25	1.5	0	0.88
a_4	1.59	0.9	0.97	0

extent a is better than b. Since every two attributes can compute a attribute preference degree, we can form an APDM. The number of row and column in this matrix is both the quantity of attributes. The matrix is shown in Table 4.

Through the APDM, we can comprehensively understand the advantageous relationships between any two attributes. Next, we need to evaluate each attribute based on these advantages and ultimately obtain scores for each attribute. To obtain the score we defined the leaving flow, entering flow and net flow. The leaving flow's definition is as follow

$$\alpha_{leaving}(a_i) = \sum_{j=1}^{n} APD(a_i, a_j), a_i, a_j \in A. \tag{18}$$

The higher leaving flow is, the greater attribute a_i surpass other attributes. The entering flow is defined as follow

$$\alpha_{entering}(a_i) = \sum_{j=1}^{n} APD(a_j, a_i), a_i, aj \in A. \tag{19}$$

The higher entering flow is, the more attribute a_i backward other attributes. We fixed leaving flow and entering flow together, and we denote the net flow

$$\alpha_{net}(a_i) = \alpha_{leaving}(a_i) - \alpha_{entering}(a_i). \tag{20}$$

With the definition of net flow, we can evaluate the goodness or badness of a attribute. Then, we can sort the attributes by the net flow and choose the attributes do well in classification to classify the samples.

Example 3.2. To better explain the creation of APDM and feature selection process, we provide an example for understanding purposes. In this example, we construct the APDM using the set of AEMs calculated in the example Example 3.1. To create an APDM, we have to define a preference function in the beginning. The preference function of information source k in column Y_j $P(x)^k_{Y_j}$ defined as follow

$$P(x)^k_{Y_j} = \begin{cases} 0, & x < 0 \\ \frac{x}{m^k_{Y_j}}, & 0 < x < m^k_{Y_j} \\ 1, & x > m^k_{Y_j} \end{cases},$$

where $m^k_{Y_j} = \max AP_{R_A(a)}(Y_j) - \min AP_{R_A(b)}(Y_j), a, b \in A$.

Next, we can calculate $APD(a,b)$ between every $a, b \in A$. For ease of understanding, in this example, all weights ω_k for the partition Y_k are 0.5. For instance, we calculate the $APD(a_1,a_2)$. The first step is calculating $APD^i_k(a_1,a_2)$, the results of $APD^i_k(a_1,a_2)$ in all conditions are listed below. $APD^1_1(a_1,a_2) = \frac{0.09-0}{0.18} = 0.5$, $APD^1_2(a_1,a_2) = 0$, $APD^2_1(a_1,a_2) = 0$, $APD^2_2(a_1,a_2) = 0$, $APD^3_1(a_1,a_2) = 0$, $APD^3_2(a_1,a_2) = 0$. The second step is to calculate $APD^i(a_1,a_2)$, the results of $APD^i(a_1,a_2)$ in all conditions are listed below. $APD^1(a_1,a_2) = 0.5 * 0.5 + 0.5 * 0 = 0.25$, $APD^2(a_1,a_2) = 0$, $APD^3(a_1,a_2) = 0$. The final step is to calculate $APD(a_1,a_2) = 0.25 + 0 + 0 = 0.25$.

Then, we can form the APDM by $APD(a,b)$. According to the structure of APDM demonstrate in Table 4, the APDM can be created. The APDM of this example is shown in Table 5.

Finally, we calculate the leaving flow $\alpha_{leaving}(a_i)$, the entering flow $\alpha_{entering}(a_i)$ and the net flow $\alpha_{net}(a_i)$. The net flow of this example is shown below. $\alpha_{net}(a_1) \approx -3.55$, $\alpha_{net}(a_2) \approx 0.29$, $\alpha_{net}(a_3) \approx 1.45$, $\alpha_{net}(a_4) \approx 1.81$. So the rank of attributes is $a_4 > a_3 > a_2 > a_1$.

3.3. Information fusion

In this subsection, we will talk about our information fusion method. With attributes selected A_{select}, we can turn the original MSDIS to $MSDIS_{select} = \{DIS_i | DIS_i = \{U, A_{select}, D, V_{A_{select}}, f_{A_{select}}, V_D, f_D\}, i = 1,2,3,\dots,n\}$. Next, we will integrate $MSDIS_{select}$ into an information table using information fusion methods.

The main idea of this method is to choose the best DIS_i in $MSDIS_{select}$ to represent the $MSDIS_{select}$. In order to choose the best DIS_i, we firstly calculate the center of MSDIS. Dropping the decision attributes, the $MSDIS_{select}$ become a $MSIS_{select} = \{IS_i | IS_i = \{U, A, V_A, f_A\}, i = 1,2,3,\dots,n\}$. We can put every IS_i in MSIS as a matrix, whose number of column is the quantity of samples and number of row is the quantity of attributes. With MSIS, we can calculate the center of $MSDIS_{select}$ V as follow:

$$v_{pq} = (\sum_{i=1}^{n} IS^{pq}_i)/n, \tag{21}$$

where v_{pq} represents the element in the pth row and q-th column of the center of $MSDIS_{select}$ V matrix, and IS^{pq}_i represents the element in the pth row and q-th column of the IS_i matrix. After calculate the center of $MSDIS_{select}$, we begin to choose the best DIS_i. The best DIS_i is correspond to the IS_i which is closest to the center of MSDIS. To judge which IS_i is closest to the center of MSDIS, we defined the function to measure the distance between two matrices.

$$L(A,B) = \|A - B\|_F, \tag{22}$$

where A and B are the two matrices, $\| * \|_F$ is F-norm of the matrix. With the $L(A,B)$, we can get the best DIS_i of MSDIS. The best DIS_i is defined as follow.

$$DIS_{best} = argmin_{DIS_i} L(V, IS_i), i = 1,2,\dots,n. \tag{23}$$

Finally the result of information confusion is DIS_{best}.

3.4. The steps of information fusion and feature selection

According to the content above, we can conclude the steps of this method. The main method is shown in Fig. 1. A MSDIS = $\{DIS_i | DIS_i = \{U, A, D, V_A, f_A, V_D, f_D\}, i = 1,2,3,\dots,n\}$ is the input of this algorithm. This method will output the top few attributes and the best DIS. Information fusion and feature selection method based of PROMETHEE in multi-sources information decision system is shown in the Algorithm 1.

In Algorithm 1, the time complex of step 1–11 is $O(n \times |A| \times |U/R_D| \times |U| \times |U|)$. The time complex of step 12–24 is $O(|A| \times |A| \times n \times |U/R_D|)$. The time complex of step 25–28 is $O(|A| \times |A|)$. The time complex of step 29–32 is $O(n)$. In conclusion, the time complex of Algorithm 1 is $O(n \times |A| \times |U/R_D| \times (|U| \times |U| + |A|))$.

4. Dynamic feature selection and information fusion based on PROMETHEE with the change of attributes and information sources

With the change of attributes and information sources, the method is different from the method we talked in Section 3. In this section, we will first talk about the information fusion method and then will discuss the feature selection methods in four conditions. The four conditions is shown in Fig. 2. Finally, we will summarize all the proposed algorithms to provide readers with a comprehensive understanding of the entire approach.

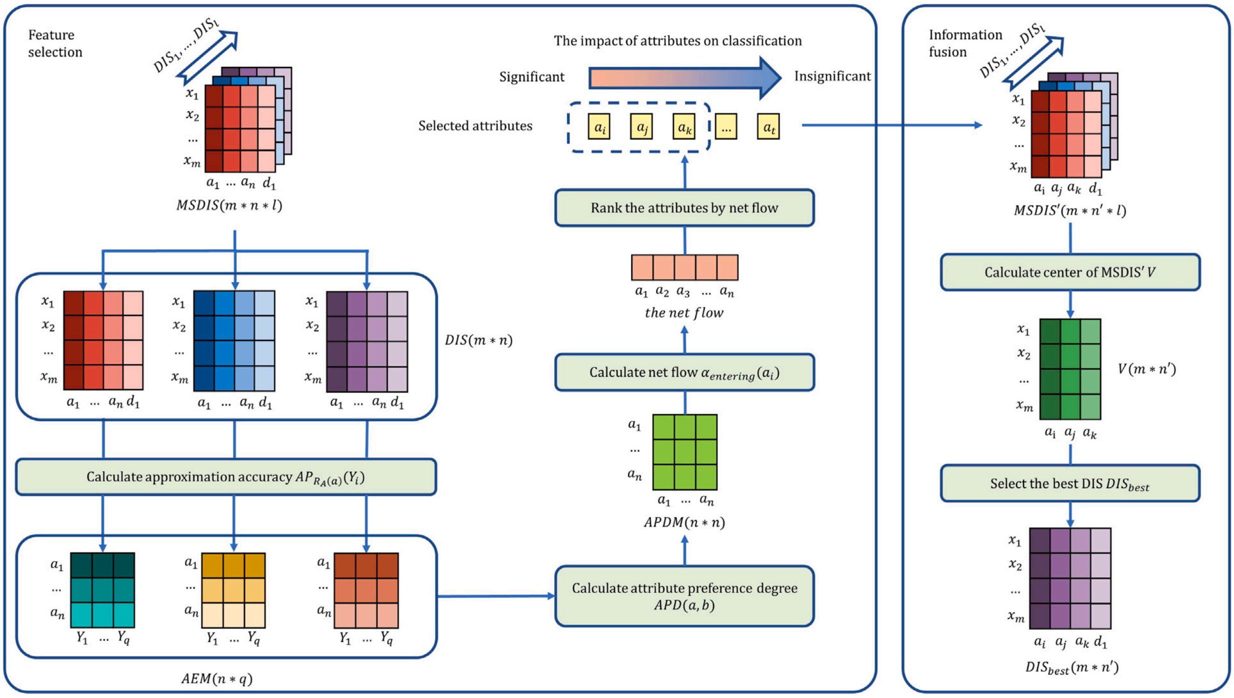

Fig. 1. The framework of static information fusion and feature selection method based on PROMETHEE in MSDIS.

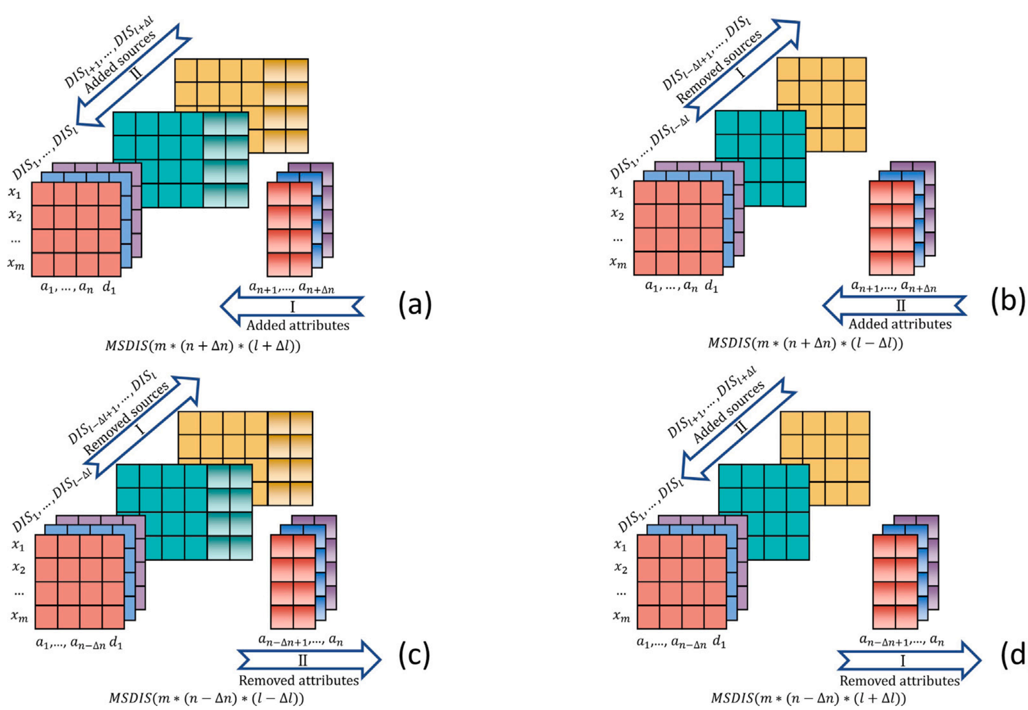

Fig. 2. Demonstration of dynamic changes in number of features and sources.

4.1. Dynamic information fusion

The approach to dynamic information fusion is similar to that of static information fusion. Both seek to find an optimal DIS within MSDIS as the result of information fusion. In dynamic situations, the original $MSDIS = \{DIS_i | DIS_i = \{U, A, D, V_A, f_A, V_D, f_D\}, i = 1, 2, 3, \ldots, n\}$ is transformed into $MSDIS' = \{DIS_i | DIS_i = \{U, A', D, V_{A'}, f_{A'}, V_D, f_D\}, i = 1, 2, 3, \ldots, n'\}$ due to changes in features and the number of information sources. Since the information fusion in this paper occurs after feature selection, the selected features A_{select} are obtained before the fusion process. The dynamic information fusion method first converts the transformed MSDIS' to $MSDIS_{select} = \{DIS_i | DIS_i = \{U, A_{select}, D, V_{A_{select}}, f_{A_{select}}, V_D, f_D\}, i = 1, 2, 3, \ldots, n'\}$. Similar to the static information fusion method, we can

Algorithm 1: Information fusion and feature selection based of PROMETHEE in multi-sources information decision system

Input: $MSDIS = \{DIS_i | DIS_i = \{U, A, D, V_A, f_A, V_D, f_D\}, i = 1, 2, 3, ..., n\}$
Output: $DIS_{best} = \{U, A_{select}, D, V_{A_{select}}, f_{A_{select}}, V_D, f_D\}$

$AEM_{group} \Leftarrow \emptyset$
for *each* $DIS_i \in MSDIS$ **do**
    $AEM \Leftarrow \emptyset$
    calculate $U/R_D \Leftarrow \{Y_1, Y_2, ..., Y_m\}$
    for *each* $a_p \in A$ **do**
        for *each* $Y_q \in U/R_D$ **do**
            $AEM[p,q] \Leftarrow \frac{|\underline{R_A(a_p)}(Y_q)|}{|\overline{R_A(a_p)}(Y_q)|}, a \in A$
        end
    end
    $AEM_{group} \Leftarrow AEM_{group} \cup AEM$
end
$APDM \Leftarrow \emptyset$
for *each* $a_p \in A$ **do**
    for *each* $a_q \in A$ **do**
        for *each* $AEM_i \in AEM_{group}$ **do**
            for *each* $Y_h \in U/R_D$ **do**
                calculate $APD_h^i(a_p, a_q)$
            end
            $APD^i(a_p, a_q) \Leftarrow \sum_{h=1}^{n} w_h * APD_h^i(a, b)$
        end
        $APD(a_p, a_q) \Leftarrow \sum_{i=1}^{n} APD^i(a, b)$
        $APDM[p,q] \Leftarrow APD(a_p, a_q)$
    end
end
for *each* $a_p \in A$ **do**
    calculate $\alpha_{net}(a_p) \Leftarrow \alpha_{leaving}(a_p) - \alpha_{entering}(a_p)$
end
rank the attributes by the net flow $\alpha_{net}(a_p)$ and choose the top few as A_{select}
$MSDIS_{select} \Leftarrow \{U, A_{select}, D, V_{A_{select}}, f_{A_{select}}, V_D, f_D\}, i = 1, 2, 3, ..., n\}$
$MSIS_{select} \Leftarrow \{IS_i | IS_i = \{U, A, V_A, f_A\}, i = 1, 2, 3, ..., n\}$
$V \Leftarrow \sum_{i=0}^{n} IS_i / n$
$DIS_{best} \Leftarrow argmin_{DIS_i} L(V, IS_i), i = 1, 2, ..., n$

obtain $MSIS_{select} = \{IS_i | IS_i = \{U, A, V_A, f_A\}, i = 1, 2, 3, \dots, n'\}$ through $MSDIS_{select}$. Then, we calculate center of $MSDIS_{select}$ through the formula as follow

$$V = \sum_{i=0}^{n} IS_i / n'. \tag{24}$$

Then, we can get DIS_{best} by the following equation

$$DIS_{best} = argmin_{DIS_i} L(V, IS_i), i = 1, 2, \dots, n'. \tag{25}$$

The DIS_{best} is the result of dynamic information fusion method.

4.2. Dynamic information fusion and feature selection in case of addition of attributes and sources

With the change of attributes and information sources, the AEM and the APDM will change. When the attributes and information sources increase, firstly, we will deal with the added attributes. Then, we will deal with the added information sources.

Given a MSDIS $= \{DIS_i | DIS_i = \{U, A, D, V_A, f_A, V_D, f_D\}, i = 1, 2, 3, \dots, n\}$, some attributes $A_{add} = \{a_1, a_2, \dots, a_k\}$ and information sources $\{DIS_j | DIS_j = \{U, A \cup A_{add}, D, V_{A \cup A_{add}}, f_{A \cup A_{add}}, V_D, f_D\}, j = 1, 2, 3, \dots, m\}$ are tend to join in the MSDIS. With the addition of attributes and information sources, the former MSDIS is changed to MSDIS'$=\{DIS_t | DIS_t = \{U, A \cup A_{add}, D, V_{A \cup A_{add}}, f_{A \cup A_{add}}, V_D, f_D\}, t = 1, 2, 3, \dots, n, n+1, n+2, \dots, n+m\}$. With the former MSDIS's AEMs, we just have to deal with the added attributes.

Deal with added attributes. For every DIS in the former MSDIS, calculating the added attributes' approximate accuracies $AP_{R_{A_{add}}(a)}(Y)$ and add them to the former DIS's AEM. Now the number of rows in AEM is increasing, while the number of columns in the matrix is still remain. To better explain the state of the AEM in cases where the number of attributes increases, the $AEM_{modified}$ is shown in Table 6.

Table 6
Modified attribute evaluation matrix when the attributes add.

$AP_{R_A(a_i)}(Y_j)$	Y_1	Y_2	...	Y_n
a_1	$AP_{R_A(a_1)}(Y_1)$	$AP_{R_A(a_1)}(Y_2)$	...	$AP_{R_A(a_1)}(Y_n)$
a_2	$AP_{R_A(a_2)}(Y_1)$	$AP_{R_A(a_2)}(Y_2)$	...	$AP_{R_A(a_2)}(Y_n)$
...	...	...	...	...
a_n	$AP_{R_A(a_n)}(Y_1)$	$AP_{R_A(a_n)}(Y_2)$	...	$AP_{R_A(a_n)}(Y_n)$
a_{n+1}	$AP_{R_A(a_{n+1})}(Y_1)$	$AP_{R_A(a_{n+1})}(Y_2)$	...	$AP_{R_A(a_{n+1})}(Y_n)$
...	...	...	...	...
a_{n+k}	$AP_{R_A(a_{n+k})}(Y_1)$	$AP_{R_A(a_{n+k})}(Y_2)$	...	$AP_{R_A(a_{n+k})}(Y_n)$

After calculating the $AEM_{modified}$, we need to reformulate the APDM of the former MSDIS. The method to calculate a APDM is similar to the method proposed in Section 3.2. As the preference function $P(x)$ is defined when using static feature selection method, the same preference function $P(x)$ should be used when calculate $APDM_{modified}$ in dynamic cases. Then, we can calculate $APD(a,b)$ of every $a, b \in A \cup A_{add}$ with Eqs. (15)–(17). Since the number of attributes increases, the shape of APDM changes. To better explain the state of the APDM when the number of attributes increases, the $APDM_{modified}$ is shown in Table 7. According to Table 7, we can generate a $APDM_{modified}$ with $APD(a,b)$ of every $a, b \in A \cup A_{add}$.

After dealing with the added attributes, we begin to deal with the added information sources.

Deal with added information sources. To deal with added information sources, we need to calculate the APDM of added information sources with the method in Section 3.2 and get the final APDM using the formula as follow.

$$APDM_{final} = APDM_{former} + APDM_{add}, \tag{26}$$

where $APDM_{final}$ is the final APDM, $APDM_{former}$ is the $APDM_{modified}$, the $APDM_{add}$ is the APDM we calculate according to the added information sources.

With the $APDM_{final}$, we calculate the net flow with Eqs. (18)–(20). With the $APDM_{final}$, we calculate the net flow with Eqs. (18)–(20). By the net flow of all attributes, we will get a rank of them. Finally, we can select any number of attributes from the top ranks as the final result. The information fusion and feature selection algorithm with the addition of attributes and information sources is shown in Algorithm 2.

In Algorithm 2, the time complex of step 1–9 is $O(n \times |A_{add}| \times |U/R_D| \times |U| \times |U|)$. The time complex of step 10–16 is $O(|A \cup A_{add}| \times |A \cup A_{add}| \times n \times |U/R_D|)$. The time complex of step 17–18 is $O(m \times |A \cup A_{add}| \times |U/R_D| \times |U| \times |U| + |A \cup A_{add}| \times |A \cup A_{add}| \times m \times |U/R_D|)$. The time complex of step 19–22 is $O(|A \cup A_{add}| \times |A \cup A_{add}|)$. The time complex of step 23–26 is $O(n+m)$. In conclusion, the time complex of Algorithm 2 is $O((n \times |A_{add}| + m \times |A \cup A_{add}|) \times |U| \times |U| + (m+n) \times |A \cup A_{add}| \times |A \cup A_{add}| \times |U/R_D|)$.

4.3. Dynamic information fusion and feature selection in case of addition of attributes and deletion of sources

This condition is somewhat different from the previous one. In the previous case, we first deal with the added attributes and then with

Table 7
Modified attribute preference degree matrix when attributes add.

$APD(a,b)$	a_1	a_2	...	a_n	a_{n+1}	...	a_{n+k}
a_1	0	$APD(a_1,a_2)$	...	$APD(a_1,a_n)$	$APD(a_1,a_{n+1})$	...	$APD(a_1,a_{n+k})$
a_2	$APD(a_2,a_1)$	0	...	$APD(a_2,a_n)$	$APD(a_2,a_{n+1})$	...	$APD(a_2,a_{n+k})$
...	...	...	...	...	...	...	...
a_n	$APD(a_n,a_1)$	$APD(a_n,a_2)$	...	0	$APD(a_n,a_{n+1})$	...	$APD(a_n,a_{n+k})$
a_{n+1}	$APD(a_{n+1},a_1)$	$APD(a_{n+1},a_2)$	...	$APD(a_{n+1},a_n)$	0	...	$APD(a_{n+1},a_{n+k})$
...	...	...	...	...	...	...	...
a_{n+k}	$APD(a_{n+k},a_1)$	$APD(a_{n+k},a_2)$	...	$APD(a_{n+k},a_n)$	$APD(a_{n+k},a_{n+1})$	...	0

Algorithm 2: The information fusion and feature selection with the addition of attributes and information sources

Input: $MSDIS=\{DIS_i|DIS_i=\{U,A,D,V_A,f_A,V_D,f_D\},i=1,2,3,...,n\}$; $A_{add}=\{a_{n+1},a_{n+2},...,a_{n+k}\}$;
$MSDIS_{add}=\{DIS_j|DIS_j=\{U,A\cup A_{add},D,V_{A\cup A_{add}},f_{A\cup A_{add}},V_D,f_D\},j=1,2,3,...,m\}$;
$AEM_{group}=\{AEM_i,i=1,2,3,...,n\}$;
Output: $DIS_{best}=\{U,A_{select},D,V_{A_{select}},f_{A_{select}},V_D,f_D\}$

1 **for** *each* $DIS_i\in MSDIS$ **do**
2 **for** *each* $a_{n+p}\in A_{add}$ **do**
3 calculate $U/R_D\Leftarrow\{Y_1,Y_2,...,Y_m\}$
4 **for** *each* $Y_q\in U/R_D$ **do**
5 calculate $AP_{R_{A(a_p)}}(Y_q)$
6 $AEM_i[n+p,q]\Leftarrow AP_{R_{A(a_p)}}(Y_q)$
7 **end**
8 **end**
9 **end**
10 $APDM_{former}\Leftarrow\emptyset$
11 **for** *each* $a_p\in A\cup A_{add}$ **do**
12 **for** *each* $a_q\in A\cup A_{add}$ **do**
13 calculate $APD(a_p,a_q)$
14 $APDM_{former}[p,q]\Leftarrow APD(a_p,a_q)$
15 **end**
16 **end**
17 calculate $APDM_{add}$ in $MSDIS_{add}$
18 $APDM_{final}\Leftarrow APDM_{former}+APDM_{add}$
19 **for** *each* $a\in A\cup A_{add}$ **do**
20 calculate $\alpha_{net}(a_i)$ by $APDM_{final}$
21 **end**
22 rank the attributes by the net flow $\alpha_{net}(a_p)$ and choose the top few as A_{select}
23 $MSDIS_{select}\Leftarrow\{U,A_{select},D,V_{A_{select}},f_{A_{select}},V_D,f_D\},i=1,2,3,...,n+m\}$
24 $MSIS_{select}\Leftarrow\{IS_i|IS_i=\{U,A,V_A,f_A\},i=1,2,3,...,n+m\}$
25 $V\Leftarrow\sum_{i=0}^{n}IS_i/(n+m)$
26 $DIS_{best}\Leftarrow argmin_{DIS_i}L(V,IS_i),i=1,2,...,n+m$

the added sources. In this case, we first deal with the removed sources and then with the added attributes. Since the steps for handling added attributes in the former MSDIS are the same as Section 4.2, we will not go into too much detail on that. Instead, we focus on the part dealing the delete information sources.

Given a MSDIS=$\{DIS_i|DIS_i=\{U,A,D,V_A,f_A,V_D,f_D\},i=1,2,3,\dots,n\}$, some attributes $A_{add}=\{a_1,a_2,\dots,a_k\}$ are tend to add to the MSDIS. Some information sources $\{DIS_j|DIS_j=\{U,A,D,V_A,f_A,V_D,f_D\},j=1,2,3,\dots,m\}$ are going to be deleted from the MSDIS. After adding the attributes and deleting the information resources, the MSDIS is changed to MSDIS'=$\{DIS_t|DIS_t=\{U,A\cup A_{add},D,V_{A\cup A_{add}},f_{A\cup A_{add}},V_D,f_D\},t=1,2,3,\dots,n-m\}$.

Deal with removed information sources. We have to delete the removed DIS's AEM_i from MSDIS's AEM group. After deleting, AEM_{group} is turned to be $\{AEM_i,i\in\{1,2,\dots,n-m\}\}$, where n is the number of DIS in former MSDIS, m is the number of removed DIS.

Deal with added attributes. The method of dealing with added attributes is same as the method in Section 4.2. According to the method in Section 4.2, we get a modified APDM. Then, we calculate the net flow with Eqs. (18)–(20) and rank all attributes by the net flow. The information fusion and feature selection algorithm with the addition of attributes and deletion of information sources is shown in the Algorithm 3.

Algorithm 3: The information fusion and feature selection with the addition of attributes and deletion of information sources

Input: $MSDIS=\{DIS_i|DIS_i=\{U,A,D,V_A,f_A,V_D,f_D\},i=1,2,3,...,n\}$; $A_{add}=\{a_1,a_2,...,a_k\}$;
$MSDIS_{del}=\{DIS_j|DIS_j=\{U,A\cup A_{add},D,V_{A\cup A_{add}},f_{A\cup A_{add}},V_D,f_D\},j=1,2,3,...,m\}$;
$AEM_{group}=\{AEM_i,i\in\{1,2,...,n\}\}$;
Output: $DIS_{best}=\{U,A_{select},D,V_{A_{select}},f_{A_{select}},V_D,f_D\}$

1 **for** *each* $DIS_j\in MSDIS_{del}$ **do**
2 $AEM_{group}\Leftarrow AEM_{group}-AEM_j$
3 **end**
4 calculate AEM as Algorithm 2's step 1-9
5 $APDM_{modified}\Leftarrow\emptyset$
6 **for** *each* $a_p\in A\cup A_{add}$ **do**
7 **for** *each* $a_q\in A\cup A_{add}$ **do**
8 calculate $APD(a_p,a_q)$
9 $APDM_{modified}[i,j]\Leftarrow APD(a_p,a_q)$
10 **end**
11 **end**
12 **for** *each* $a_p\in A\cup A_{add}$ **do**
13 calculate $\alpha_{net}(a_p)$ by $AEPM_{modified}$
14 **end**
15 rank the attributes by the net flow $\alpha_{net}(a_p)$ and choose the top few as A_{select}
16 $MSDIS_{select}\Leftarrow\{U,A_{select},D,V_{A_{select}},f_{A_{select}},V_D,f_D\},i=1,2,3,...,n-m\}$
17 $MSIS_{select}\Leftarrow\{IS_i|IS_i=\{U,A,V_A,f_A\},i=1,2,3,...,n-m\}$
18 $V\Leftarrow\sum_{i=0}^{n}IS_i/(n-m)$
19 $DIS_{best}\Leftarrow argmin_{DIS_i}L(V,IS_i),i=1,2,...,n-m$

In Algorithm 3, the time complex of step 1–3 is $O(m)$. The time complex of step 4 is $O(n\times|A_{add}|\times|U/R_D|\times|U|\times|U|)$. The time complex of step 5–11 is $O(|A\cup A_{add}|\times|A\cup A_{add}|\times n\times|U/R_D|)$. The time complex of step 12–15 is $O(|A\cup A_{add}|\times|A\cup A_{add}|)$. The time complex of step 16–19 is $O(n-m)$. In conclusion, the time complex of Algorithm 3 is $O(m+(|A_{add}|\times|U|\times|U|+|A\cup A_{add}|\times|A\cup A_{add}|)\times n\times|U/R_D|)$

4.4. Dynamic information fusion and feature selection in case of deletion of attributes and sources

In this subsection, we first deal with removed information sources. Then, we will deal with removed attributes. When the attribute decrease, We only need to delete the rows and columns corresponding to the removed attributes in the AEM to form a new AEM. For the deletion of information sources, the method is same as the method in Section 4.3.

Given a MSDIS=$\{DIS_i|DIS_i=\{U,A,D,V_A,f_A,V_D,f_D\},i=1,2,3,\dots,n\}$, some attributes $A_{del}=\{a_1,a_2,\dots,a_k\}$ and some information

Table 8
Modified attribute evaluation matrix when attributes delete.

$AP_{R_A(a_i)}(Y_j)$	Y_1	Y_2	...	Y_n
a_1	$AP_{R_A(a_1)}(Y_1)$	$AP_{R_A(a_1)}(Y_2)$	...	$AP_{R_A(a_1)}(Y_n)$
a_2	$AP_{R_A(a_2)}(Y_1)$	$AP_{R_A(a_2)}(Y_2)$	...	$AP_{R_A(a_2)}(Y_n)$
...	...	...	...	...
a_{n-k}	$AP_{R_A(a_{n-k})}(Y_1)$	$AP_{R_A(a_{n-k})}(Y_2)$	...	$AP_{R_A(a_{n-k})}(Y_n)$

Table 9
Modified attribute preference degree matrix when attributes delete.

$APD(a,b)$	a_1	a_2	...	a_{n-k}
a_1	0	$APD(a_1,a_2)$	...	$APD(a_1,a_{n-k})$
a_2	$APD(a_2,a_1)$	0	...	$APD(a_2,a_{n-k})$
...	...	...	...	...
a_{n-k}	$APD(a_{n-k},a_1)$	$APD(a_{n-k},a_2)$	...	0

sources $\{DIS_j|DIS_j = \{U,A,D,V_A,f_A,V_D,f_D\}, j = 1,2,3,\dots,m\}$ are going to be deleted. Then, attribute set will be $A' = A - A_{del}$ and the MSDIS will be MSDIS'=$\{DIS_t|DIS_t = \{U,A',D,V_{A'},f_{A'},V_D,f_D\}, t = 1,2,3,\dots,n-m\}$.

Deal with removed information sources. The method to remove information sources is same as the method mentioned in Section 4.3. We have to delete AEMs corresponding to removed DIS in MSDIS.

Deal with removed attributes. With the attributes to be deleted, we will firstly delete the rows corresponding to the removed attributes in the former MSDIS's AEMs. To better explain the state of the AEM when the number of attributes decreases, the $AEM_{modified}$ is shown in Table 8. Next, we can reformulate the APDM using $AEM_{modified}$. The method of reformulating the APDM is similar to the method introduced in Section 3.2. As the preference function $P(x)$ is defined when using static feature selection method, the same preference function $P(x)$ should be used while formulate $APDM_{modified}$ in dynamic cases. With the preference function $P(x)$, $APD(a,b)$ of every $a,b \in A'$ can be calculated. Since the number of attributes decreases, the shape of APDM changes. To better explain the shape of APDM when the number of attributes decreases, the $APDM_{modified}$ is shown in Table 9. According to Table 9, we can generate an $APDM_{modified}$ using $APD(a,b)$ of every $a,b \in A'$.

Finally, we calculate the net flow with Eqs. (18)–(20) and rank all attributes by the net flow. The information fusion and feature selection algorithm with the deletion of attributes and information sources is shown in the Algorithm 4.

In Algorithm 4, the time complex of step 1–3 is $O(m)$. The time complex of step 4–6 is $O(n)$. The time complex of step 7–13 is $O(|A'|\times|A'|\times n\times|U/R_D|)$. The time complex of step 14–17 is $O(|A'|\times|A'|)$. The time complex of step 18–21 is $O(n-m)$. In conclusion, the time complex of Algorithm 4 is $O(m+|A'|\times|A'|\times n\times|U/R_D|)$.

4.5. Dynamic information fusion and feature selection in case of deletion of attributes and addition of sources

In this section, we will first deal with removed attributes and then with added information sources. Since the method of deal with removed attributes and added information sources is told in Sections 4.2 and 4.4, we will not explain the details of these methods in this section.

Given a MSDIS $= \{DIS_i|DIS_i = \{U,A,D,V_A,f_A,V_D,f_D\}, i = 1,2,3,\dots,n\}$, some attributes $A_{del} = \{a_1,a_2,\dots,a_k\}$ are going to be deleted, and some information sources $\{DIS_j|DIS_j = \{U,A,D,V_A,f_A,V_D,f_D\}, j = 1,2,3,\dots,m\}$ are tend to add to the MSDIS. Then, the attributes set will be $A' = A - A_{del}$ and the MSDIS will be MSDIS'=$\{DIS_t|DIS_t = \{U,A',D,V_{A'},f_{A'},V_D,f_D\}, t = 1,2,3,\dots,n+m\}$.

Deal with removed attributes. The method to remove attributes is introduced in Section 4.4. We have to remove the rows corresponding to the removed attributes in AEM and calculate a $APDM_{former}$ by the new AEM.

Algorithm 4: The information fusion and feature selection algorithm with the deletion of attributes and information sources

Input: $MSDIS = \{DIS_i|DIS_i = \{U,A,D,V_A,f_A,V_D,f_D\}, i = 1,2,3,...,n\}$; $A_{del} = \{a_1,a_2,...,a_k\}$; $MSDIS_{del} = \{DIS_j|DIS_j = \{U,A,D,V_A,f_A,V_D,f_D\}, j = 1,2,3,...,m\}$; $AEM_{group} = \{AEM_i, i \in \{1,2,...,n\}\}$;
Output: $DIS_{best} = \{U,A_{select},D,V_{A_{select}},f_{A_{select}},V_D,f_D\}$

1 **for** *each* $DIS_j \in MSDIS_{del}$ **do**
2 $\quad AEM_{group} \Leftarrow AEM_{group} - AEM_j$
3 **end**
4 **for** *each* $AEM_i \in AEM_{group}$ **do**
5 $\quad$ Delete the rows corresponding to removed attributes in AEM_i
6 **end**
7 $APDM_{modified} \Leftarrow \emptyset$
8 **for** *each* $a_p \in A'$ **do**
9 $\quad$ **for** *each* $a_q \in A'$ **do**
10 $\quad\quad$ calculate $APD(a_p,a_q)$
11 $\quad\quad$ $APDM_{modified}[i,j] \Leftarrow APD(a_p,a_q)$
12 $\quad$ **end**
13 **end**
14 **for** *each* $a_p \in A'$ **do**
15 $\quad$ calculate $\alpha_{net}(a_p)$ by $APDM_{modified}$
16 **end**
17 rank the attributes by the net flow $\alpha_{net}(a_p)$ and choose the top few as A_{select}
18 $MSDIS_{select} \Leftarrow \{U,A_{select},D,V_{A_{select}},f_{A_{select}},V_D,f_D\}, i = 1,2,3,...,n-m\}$
19 $MSIS_{select} \Leftarrow \{IS_i|IS_i = \{U,A,V_A,f_A\}, i = 1,2,3,...,n-m\}$
20 $V \Leftarrow \sum_{i=0}^{n} IS_i/(n-m)$
21 $DIS_{best} \Leftarrow argmin_{DIS_i} L(V,IS_i), i = 1,2,...,n-m$

Deal with added information sources. Its method is introduced in Section 4.2. Briefly, we calculate the APDM of added information sources and use Eq. (26) in order to get $APDM_{final}$. Then, we calculate the net flow with Eqs. (18)–(20) and rank all attributes by the net flow.

The information fusion and feature selection algorithm with the deletion of attributes and the addition of information sources is shown in the Algorithm 5.

In Algorithm 5, the time complex of step 1–3 is $O(m)$. The time complex of step 4–10 is $O(|A'|\times|A'|\times n\times|U/R_D|)$. The time complex of step 11–12 is $O(m\times|A'|\times|U/R_D|\times|U|\times|U|+|A'|\times|A'|\times m\times|U/R_D|)$. The time complex of step 13–16 is $O(|A'|\times|A'|)$. The time complex of step 17–20 is $O(n+m)$. In conclusion, the time complex of Algorithm 5 is $O(m+|A'|\times|A'|\times n\times|U/R_D|+m\times|A'|\times|U/R_D|\times(|U|\times|U|+|A'|))$.

The above provides a detailed introduction to both the static and dynamic feature selection and information fusion algorithms. Next, we will provide a brief summary of the proposed algorithms to help readers easily locate the corresponding methods.

• Algorithm 1 is the static feature selection and information fusion algorithm.

Algorithms 2–5 are dynamic feature selection and information fusion algorithms, each corresponding to a dynamic scenario as shown in Fig. 2:

• Algorithm 2 handles the case where both the number of attributes and sources increase.

• Algorithm 3 addresses the case where the number of attributes increases while the number of sources decreases.

• Algorithm 4 deals with the scenario where both the number of attributes and sources decrease.

• Algorithm 5 is used when the number of attributes decreases while the number of sources increases.

Algorithm 5: The information fusion and feature selection with the deletion of attributes and the addition of information sources

Input: $MSDIS = \{DIS_i | DIS_i = \{U, A, D, V_A, f_A, V_D, f_D\}, i = 1, 2, 3, ..., n\}$;$A_{del} = \{a_1, a_2, ..., a_k\}$;
$MSDIS_{add}\{DIS_j | DIS_j = \{U, A, D, V_A, f_A, V_D, f_D\}, j = 1, 2, 3, ..., m\}$; $AEM = \{AEM_i, i \in \{1, 2, ..., n\}\}$;
Output: $DIS_{best} = \{U, A_{select}, D, V_{A_{select}}, f_{A_{select}}, V_D, f_D\}$

```
for each AEM_i ∈ AEM do
    Delete the rows corresponding to removed attributes in AEM_i.
end
APDM_former ⇐ ∅
for each a_p ∈ A' do
    for each a_q ∈ A' do
        calculate APD(a_p, a_q)
        APDM_former[p, q] ⇐ APD(a_p, a_q)
    end
end
calculate APDM_add in MSDIS_add
APDM_final ⇐ APDM_former + APDM_add
for each a ∈ A' do
    calculate α_net(a_i) by APDM_final
end
rank the attributes by the net flow α_net(a_p) and choose the top few as A_select
MSDIS_select ⇐ {U, A_select, D, V_{A_select}, f_{A_select}, V_D, f_D}, i = 1, 2, 3, ..., n + m}
MSIS_select ⇐ {IS_i | IS_i = {U, A, V_A, f_A}, i = 1, 2, 3, ..., n + m}
V ⇐ Σ_{i=0}^{n} IS_i/(n + m)
DIS_best ⇐ argmin_{DIS_i} L(V, IS_i), i = 1, 2, ..., n + m
```

5. Experiment and results

In this section, some experiment are done to prove the effect of our static method and the compare between the static method and the dynamic method. The data set we use are shown on Table 10. All the experiment this time are run by personal computer, the operating environment is listed in Table 11. Because the MSDIS is hard to find, so we use some method to turn DIS to MSDIS.

The method is discussed below. Given a $DIS = \{U, A, D, V_A, f_A, V_D, f_D\}$, Firstly, standardization and normalization are used to DIS in order to make the values in DIS between 0 to 1. Next, we generate n random numbers $r = \{r_1, r_2, \dots, r_n\}$ satisfy the normal distribution N(0, 0.1),where n is the quantity of samples. Let i be the number of DIS in MSDIS and j be the number of sample in U, the value of sample x on attribute a in DIS_i is $f_{Ai}(x, a) = f_A(x, a) * (1 + r_j), x \in U, a \in A$. Additionally, the value of sample x's decision attribute is $f_{Di}(x, d) = f_D(x, d), x \in U, d \in D$. Then we can get a $MSDIS = \{DIS_i | DIS_i = \{U, A, D, V_A, f_A, V_D, f_D\}, i = 1, 2, 3, \dots, n\}$.

In the experiment, the preference function of information source k in column Y_j $P(x)^k_{Y_j}$ defined as follow.

$$P(x)^k_{Y_j} = \begin{cases} 0, & x < 0 \\ \frac{x}{m^k_{Y_j}}, & 0 < x < m^k_{Y_j} \\ 1, & x > m^k_{Y_j} \end{cases},$$

where $m^k_{Y_j} = \max AP_{R_{A(a)}}(Y_j) - \min AP_{R_{A(b)}}(Y_j), a, b \in A$.

5.1. Experiment design

5.1.1. The comparative experiment of effectiveness of parameters in feature selection and information fusion method

In this subsection, we will analysis the effectiveness of the static information fusion and feature selection method. We will divide the subsection into two parts. In the first part, We vary the threshold value alpha from 0.05 to 0.5 with a step size of 0.05 and observe the accuracy of attributes selected under different alpha values in the KNN, SVM and DT algorithm for classification on various datasets. We need to control the number of selected features in order to explore the impact of different alpha values on classification accuracy. Choosing an overly large feature selection ratio makes it difficult to observe the effect of feature selection, while selecting too small a ratio significantly affects the original data structure and classification accuracy. Therefore, we choose 0.5 as the feature selection rate for this part of the experiment. In the second part, we set the threshold value alpha to 0.2. If the alpha value is too large or too small, the feature selection effectiveness of rough set theory will be relatively low, as can be observed from Fig. 3. We vary the attribute selection rate from 0.1 to 1 with a step size of 0.1 and observe the accuracy after classification using the KNN, SVM and DT algorithm under different attribute selection rates for different datasets. The Fig. 3 and Fig. 4 show the results of the experiment. The classification accuracy obtained in the experiment is the result of taking the average of five runs.

5.1.2. The comparative experiment of classification results with four other algorithms

In this section, we will compare the proposed static multi-source feature selection algorithm (RST-P) with other feature selection algorithms in order to analyze the effectiveness of the proposed algorithm. Since there are currently few multi-source feature selection algorithms, we have selected a total of four single-source feature selection algorithms for comparison with the proposed algorithm. The four selected algorithms are CE [40], WD [41], OD-KNN [9], FCM [10].

In this comparative experiment, for each dataset, we randomly generated a total of 10 information sources using the aforementioned method of source generation. For the selected four single-source feature selection algorithms, we sequentially input the 10 generated information sources into the single-source feature selection algorithms, yielding 10 feature selection results. These 10 feature selection results are then combined using the following method:

• Intersection: We take the intersection of the 10 feature selection results in sequence to obtain the final feature selection result for the multi-source information system. This method can identify features that are important in every information source, thereby ensuring the high quality of the selection results.

• Union: We take the union of the 10 feature selection results in sequence to obtain the final feature selection result for the multi-source information system. This method can identify all features that are important across the various information sources, thereby ensuring the comprehensiveness of the feature selection results.

Using the proposed information fusion method, the 10 generated information sources are fused into one source. Based on the results obtained from the single-source feature selection algorithm applied to the fused source, several features are selected from the result, thereby generating the data to be input into the classification algorithm.

For the proposed feature selection method, the 10 generated information sources are directly input to obtain the final feature selection result. Similarly, by using the proposed information fusion algorithm, we can apply the feature selection results to the fused information source, thereby obtaining the final data to be input into the classification algorithm. Since our algorithm is capable of ranking the features and selecting different numbers of features, we select 40%, 60%, and 80% of the features for comparison with the other four feature selection algorithms. The selection of these feature selection rates is based on the experimental results shown in Fig. 4. From the analysis of Fig. 4, we determined that the optimal feature selection rate for our algorithm is 0.6. Therefore, we conducted experiments with three feature selection rates close to this value. Additionally, based on Fig. 3 and the corresponding analysis, we selected a moderate alpha value of 0.2 as the alpha parameter for this experiment.

Table 10
The description of data sets.

No.	Data sets	Abbreviation	Samples	Attributes	Classes
1	Abalone	Abalone	4177	8	3
2	Avila	Avila	20 867	10	12
3	Connectionist Bench(Sonar, Mines vs. Rocks)	CB	208	60	2
4	Ecoli	Ecoli	336	7	8
5	Musk	Musk	476	168	2
6	Period Changer	PC	90	1177	2
7	QSAR	QSAR	1055	41	2
8	Waveform	Waveform	5000	21	3
9	Wine	Wine	178	13	3

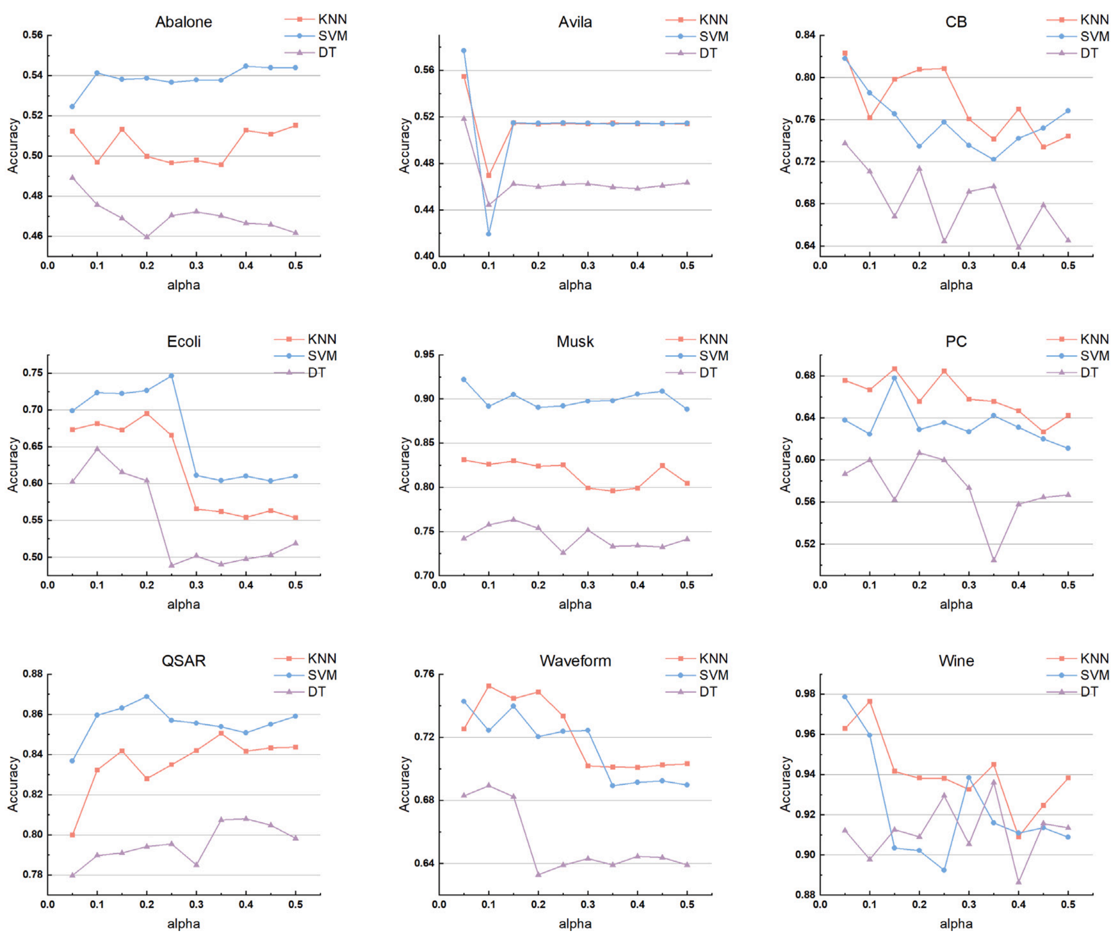

Fig. 3. Classification accuracy of input feature selection results of three algorithms under changes in alpha.

Table 11
Operating ambient.

Name	Model	Parameter
CPU	12th Gen Intel(R) Core(TM) i7-12700H	2.30 GHz
Platform	Python	3.9
System	Windows11	64bit
Memory	DDR5	16 GB;4800MKz
Hard Disk	Micron MTFDKBA512TFH	477 GB

We compare these feature selection algorithms based on classification accuracy. Additionally, we use three classification algorithms (KNN, Support Vector Machine (SVM) and Decision Tree (DT))to classify the data, thereby obtaining a more comprehensive result. Additionally, we used a 5-fold cross-validation method to input the feature-selected data into three different classifiers. The final classification accuracy is calculated as the average of the results from the 5-fold cross-validation. Since using the intersection as the fusion method for the feature selection results of the single-source feature selection algorithm can result in an empty set, we have marked such occurrences with a '\' symbol. The results of this experiment are displayed in Tables 12–17 (see Tables 13–16).

5.1.3. The comparative experiment of the efficiency of dynamic information fusion and feature selection method

In this subsection, we analysis the efficiency of dynamic information fusion and feature selection method. Because every case have different dynamic method, we will analysis the efficiency in four cases. The α is choose to be 0.2. As shown in Fig. 3, if the alpha value is either too high or too low, the effectiveness of feature selection using rough set theory will be significantly reduced. We selected 20%, 40%, 60%, 80%, and all features to explore the performance of the dynamic feature selection and information fusion algorithm under small-scale, medium-scale, and large-scale feature conditions. Similarly, we selected 12, 14, 16, 18, and 20 information sources to investigate the algorithm's performance with a small, medium, and large number of sources. The number of

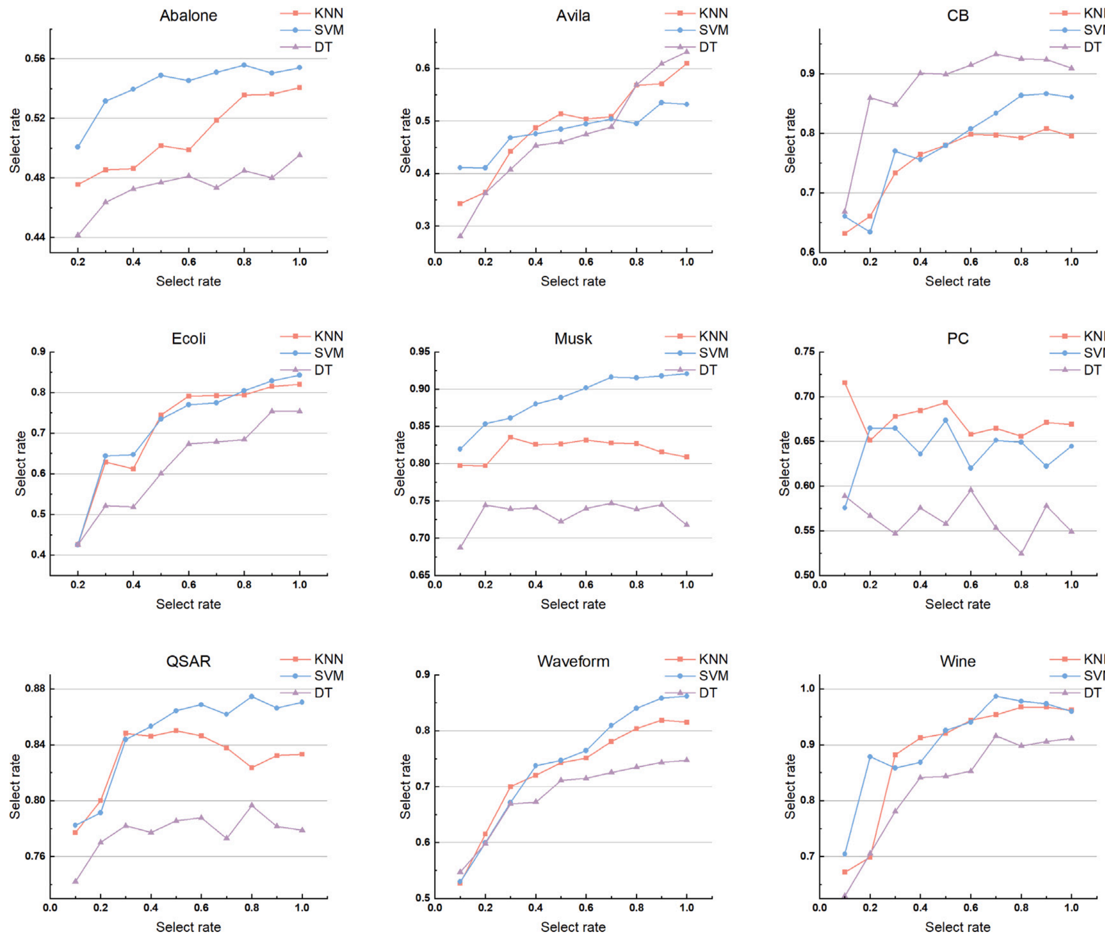

Fig. 4. Classification accuracy of input feature selection results of three algorithms under changes in select rate.

Table 12
The comparison of classification results with different algorithms on KNN classifier in the case of using intersection method to combine feature selection results of single-source feature selection algorithms.

	raw	∩CE	∩WD	∩OD-KNN	∩FCM	RST-P (40%)	RST-P (60%)	RST-P (80%)
abalone	0.5204	0.4537	0.4764	0.4858	0.5083	0.5010	0.5002	**0.5177**
avila	0.6073	0.5194	0.5175	0.5219	0.5248	0.4857	0.5066	**0.5668**
CB	0.7884	\	0.6317	\	0.6893	**0.7247**	**0.7891**	**0.7803**
ecoli	0.8299	**0.8297**	0.8274	0.8219	0.7488	0.6065	0.6953	0.7612
musk	0.8013	0.5520	0.6513	0.6374	0.7774	**0.8340**	**0.8378**	**0.8252**
PC	0.6644	0.7000	0.6854	0.6644	0.6975	**0.7067**	0.6956	0.6489
QSAR	0.8427	0.6973	0.7994	\	0.8328	**0.8580**	**0.8298**	**0.8402**
waveform	0.8119	0.7851	0.7652	0.8129	0.7682	0.6594	0.7560	**0.8166**
wine	0.9527	0.8055	0.6281	0.8935	0.8999	**0.9382**	**0.9527**	**0.9639**

Table 13
The comparison of classification results with different algorithms on KNN classifier in the case of using union method to combine feature selection results of single-source feature selection algorithms.

	raw	∪CE	∪WD	∪OD-KNN	∪FCM	RST-P (40%)	RST-P (60%)	RST-P (80%)
abalone	0.5204	**0.5347**	0.5247	0.5219	0.5103	0.5010	0.5002	0.5177
avila	0.6073	0.5392	0.5193	0.5247	0.5259	0.4857	0.5066	**0.5668**
CB	0.7884	**0.7934**	0.6902	0.8232	0.7633	0.7247	0.7891	0.7803
ecoli	0.8299	**0.8291**	0.8202	0.8094	0.7501	0.6065	0.6953	0.7612
musk	0.8013	0.7832	0.7866	0.7803	0.7799	**0.8340**	**0.8378**	**0.8252**
PC	0.6644	0.6511	0.6652	0.6756	0.6682	**0.7067**	**0.6956**	0.6489
QSAR	0.8427	0.8398	0.8398	0.8358	0.8400	**0.8580**	0.8298	**0.8402**
waveform	0.8119	0.7874	0.7756	0.8108	0.7701	0.6594	0.7560	**0.8166**
wine	0.9527	0.9573	0.9571	0.9574	0.9562	0.9382	0.9527	**0.9639**

features and sources in each configuration can also be adjusted as long as they meet the conditions of different cases, allowing us to explore the algorithm's performance under various scenarios. Just in this subsection, we chose these specific parameters for our experiments.

case a. Addition of attributes and information sources

We make five different configurations from the data set and performed both static and dynamic feature selection and information fusion algorithms on them. The specific steps are as follows:

Table 14
The comparison of classification results with different algorithms on SVM classifier in the case of using intersection method to combine feature selection results of single-source feature selection algorithms.

	raw	∩CE	∩WD	∩OD-KNN	∩FCM	RST-P (40%)	RST-P (60%)	RST-P (80%)
abalone	0.5503	0.5071	0.5076	0.5114	0.5476	**0.5378**	**0.5390**	**0.5522**
avila	0.5337	0.5106	0.5264	0.5049	0.4833	0.4727	0.4941	**0.5754**
CB	0.8058	\	0.6844	\	0.6136	**0.7288**	**0.8018**	**0.7933**
ecoli	0.8506	0.8416	**0.8429**	0.8362	0.7703	0.6750	0.7501	0.7845
musk	0.8685	0.5693	0.6753	0.6676	0.7711	**0.8643**	**0.8727**	**0.8685**
PC	0.7400	0.7000	0.7144	0.7000	0.7026	**0.7444**	**0.7289**	**0.7222**
QSAR	0.8510	0.7282	0.7846	\	0.8292	**0.8353**	**0.8408**	**0.8400**
waveform	0.8606	0.8205	0.8245	0.8428	0.8010	0.7078	0.7970	**0.8561**
wine	0.9797	0.8279	0.5942	0.8922	0.9043	**0.9538**	**0.9551**	**0.9742**

Table 15
The comparison of classification results with different algorithms on SVM classifier in the case of using union method to combine feature selection results of single-source feature selection algorithms.

	raw	∪CE	∪WD	∪OD-KNN	∪FCM	RST-P (40%)	RST-P (60%)	RST-P (80%)
abalone	0.5503	0.5503	0.5367	0.5421	0.5484	0.5378	0.5390	**0.5522**
avila	0.5337	0.5237	0.5324	0.5106	0.4833	0.4727	0.4941	**0.5754**
CB	0.8058	0.7923	0.6942	0.7941	0.7963	0.7288	**0.8018**	0.7933
ecoli	0.8506	**0.8434**	0.8400	0.8345	0.7697	0.6750	0.7501	0.7845
musk	0.8685	0.8264	0.8467	0.8009	0.8029	**0.8643**	**0.8727**	**0.8685**
PC	0.7400	0.7156	0.6924	0.7089	0.7103	**0.7444**	**0.7289**	**0.7222**
QSAR	0.8510	0.8451	**0.8521**	0.8440	0.8482	0.8353	0.8408	0.8400
waveform	0.8606	0.8320	0.8326	0.8560	0.7999	0.7078	0.7970	**0.8561**
wine	0.9797	0.9799	**0.9820**	0.9606	0.9686	0.9538	0.9551	0.9742

Table 16
The comparison of classification results with different algorithms on DT classifier in the case of using intersection method to combine feature selection results of single-source feature selection algorithms.

	raw	∩CE	∩WD	∩OD-KNN	∩FCM	RST-P (40%)	RST-P (60%)	RST-P (80%)
abalone	0.4863	0.4346	0.4413	0.4487	0.4723	0.4603	0.4636	**0.4813**
avila	0.6357	0.4816	0.4862	0.4792	0.4749	0.4540	0.4727	**0.5754**
CB	0.7413	\	0.5427	\	0.5999	**0.7146**	**0.7232**	**0.7231**
ecoli	0.7548	0.7376	**0.7529**	0.7452	0.7012	0.5590	0.6797	0.6982
musk	0.7420	0.5185	0.5974	0.6071	0.7395	**0.7408**	**0.7496**	**0.7336**
PC	0.5778	**0.6733**	0.6488	0.6133	0.6243	0.6000	0.6178	0.5622
QSAR	0.7945	0.6900	0.7350	\	0.7816	**0.7989**	**0.7900**	**0.7992**
waveform	0.7387	0.7301	0.7394	0.7422	0.6989	0.6056	0.6946	**0.7437**
wine	0.9113	0.7440	0.6148	0.8449	0.8765	0.8696	0.8707	**0.9261**

Table 17
The comparison of classification results with different algorithms on DT classifier in the case of using union method to combine feature selection results of single-source feature selection algorithms.

	raw	∪CE	∪WD	∪OD-KNN	∪FCM	RST-P (40%)	RST-P (60%)	RST-P (80%)
abalone	0.4863	0.4795	0.4781	0.4765	0.4732	0.4603	0.4636	**0.4813**
avila	0.6357	0.4864	0.4967	0.4827	0.4745	0.4540	0.4727	**0.5754**
CB	0.7413	0.7488	0.5827	**0.7597**	0.7507	0.7146	0.7232	0.7231
ecoli	0.7548	**0.7595**	0.7554	0.7542	0.6792	0.5590	0.6797	0.6982
musk	0.7420	0.7390	0.7479	**0.7618**	0.7517	0.7408	0.7496	0.7336
PC	0.5778	0.6044	0.5938	0.5889	0.5843	0.6000	**0.6178**	0.5622
QSAR	0.7945	0.7983	0.7900	0.7917	0.7896	0.7989	0.7900	**0.7992**
waveform	0.7387	0.7300	0.7376	0.7409	0.6974	0.6056	0.6946	**0.7437**
wine	0.9113	0.9055	0.9090	0.9224	0.9044	0.8696	0.8707	**0.9261**

The five configurations of the data are:
• First configuration: 20% of attributes with 12 sources of data
• Second configuration: 40% of attributes with 14 sources of data
• Third configuration: 60% of attributes with 16 sources of data
• Fourth configuration: 80% of attributes with 18 sources of data
• Fifth configuration: All attributes with 20 sources of data

For the dynamic scenario, we recorded the time required for computations by inputting data from the first configuration to the fifth configuration sequentially into the dynamic algorithm. For the static scenario, we separately calculated these five configurations using a static feature selection and information fusion algorithm and recorded the time taken. Finally, we compared the time taken for static and dynamic computations for the same configurations and represented the results in a bar chart. The result of experiment is shown in Fig. 5. This chart is designed to show a comparison of the time consumption for dynamic and static feature selection and information fusion algorithms as the number of attributes and information sources increases.

case b. Addition of attributes and deletion of information sources

The same as case a, we make five different configurations from the data set and performed both static and dynamic feature selection and information fusion algorithms on them.

The five configurations of the data are:
• First configuration: 20% of attributes with 20 sources of data
• Second configuration: 40% of attributes with 18 sources of data
• Third configuration: 60% of attributes with 16 sources of data
• Fourth configuration: 80% of attributes with 14 sources of data
• Fifth configuration: All attributes with 12 sources of data

The result of experiment is shown in Fig. 6. This chart is designed to show a comparison of the time consumption for dynamic and static

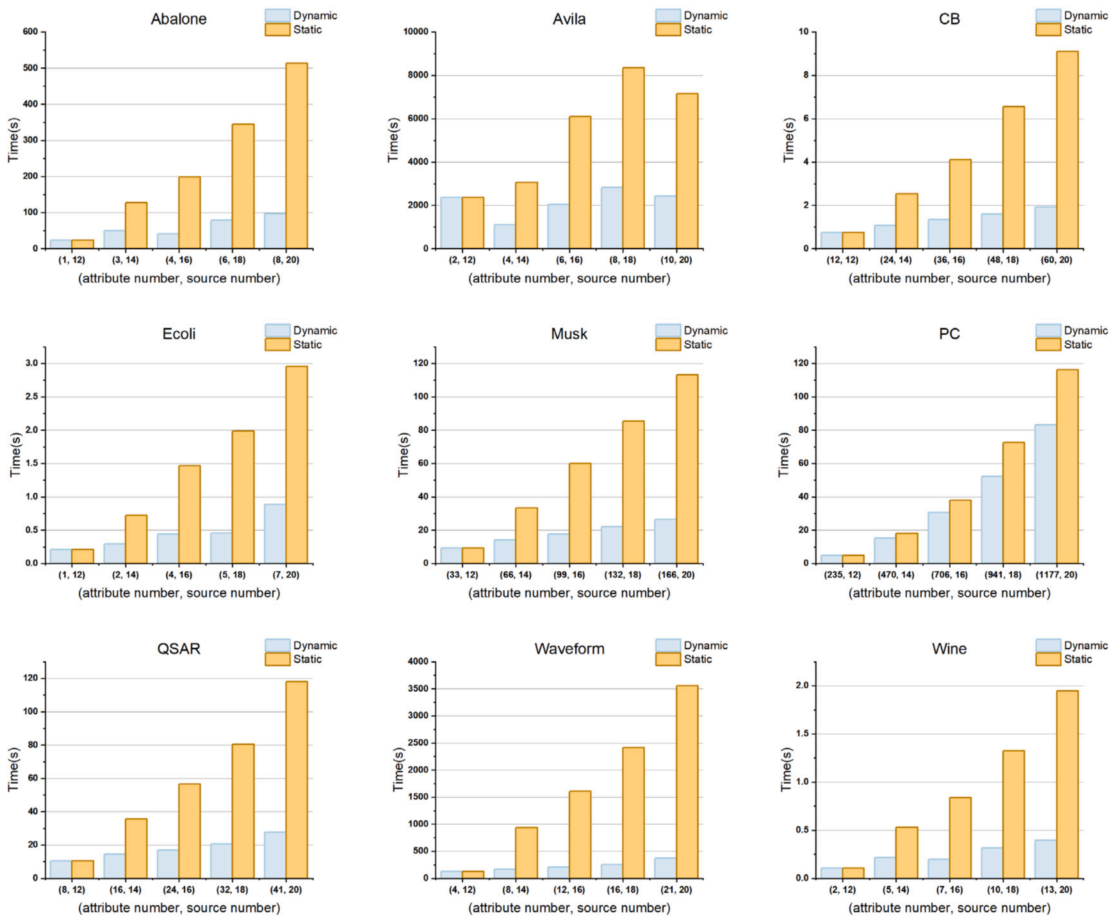

Fig. 5. The time Static and Dynamic methods use in case of addition of attributes and sources.

feature selection and information fusion algorithms as the number of attributes increases and number of information sources decreases.

case c. Deletion of attributes and information sources

The five configurations of the data are:

• First configuration: All of attributes with 20 sources of data
• Second configuration: 80% of attributes with 18 sources of data
• Third configuration: 60% of attributes with 16 sources of data
• Fourth configuration: 40% of attributes with 14 sources of data
• Fifth configuration: 20% attributes with 12 sources of data

The result of experiment is shown in Fig. 7. This chart is designed to show a comparison of the time consumption for dynamic and static feature selection and information fusion algorithms as the number of attributes and information sources decreases.

case d. Deletion of attributes and addition of information sources

The five configurations of the data are:

• First configuration: All of attributes with 12 sources of data
• Second configuration: 80% of attributes with 14 sources of data
• Third configuration: 60% of attributes with 16 sources of data
• Fourth configuration: 40% of attributes with 18 sources of data
• Fifth configuration: 20% attributes with 20 sources of data

The result of experiment is shown in Fig. 8. This chart is designed to show a comparison of the time consumption for dynamic and static feature selection and information fusion algorithms as the number of attributes decreases and the number of information sources increases.

5.2. Experiment results

5.2.1. The analysis of effectiveness of parameters in feature selection and information fusion method

From Fig. 3 and Fig. 4, we can see that for most datasets, the optimal range for the alpha value for feature selection is between 0.05 and 0.2. From the experimental results, it is evident that selecting an alpha value that is too large or too small leads to a decline in the final feature selection performance. This can be explained theoretically: When the alpha value is too small, different samples become independent of each other, and the algorithm is unable to identify relationships between the samples. As a result, it cannot leverage these relationships to identify the best-performing features. When the alpha value is too large, all samples become interconnected, making it impossible to separate samples into distinct classes, similar to how classification attributes divide samples. Without these class distinctions, the algorithm cannot find optimal features based on classification relationships. For feature selection rate, it can be observed from the images that the accuracy for most datasets increases with an increase in the number of attributes. Additionally, for most datasets, there is a clear distinction at a feature selection rate of 0.4: when the feature selection rate is less than 0.4, the decrease in classification accuracy is more noticeable compared to when the feature selection rate is greater than 0.4. Therefore, considering both classification performance and runtime, a feature selection rate of 0.6 is considered optimal. Meanwhile, this to some extent reflects the significant impact of the first few features selected using feature selection method and source selected using information fusion method on classification accuracy, reflecting the effectiveness of our information fusion and feature selection method.

5.2.2. The comparative analysis of classification results with four other algorithms

From Tables 12–17, we can observe that our feature selection algorithm generally outperforms other feature selection algorithms, providing more effective results in most cases. The table shows that our method performs particularly well when the feature selection rate is

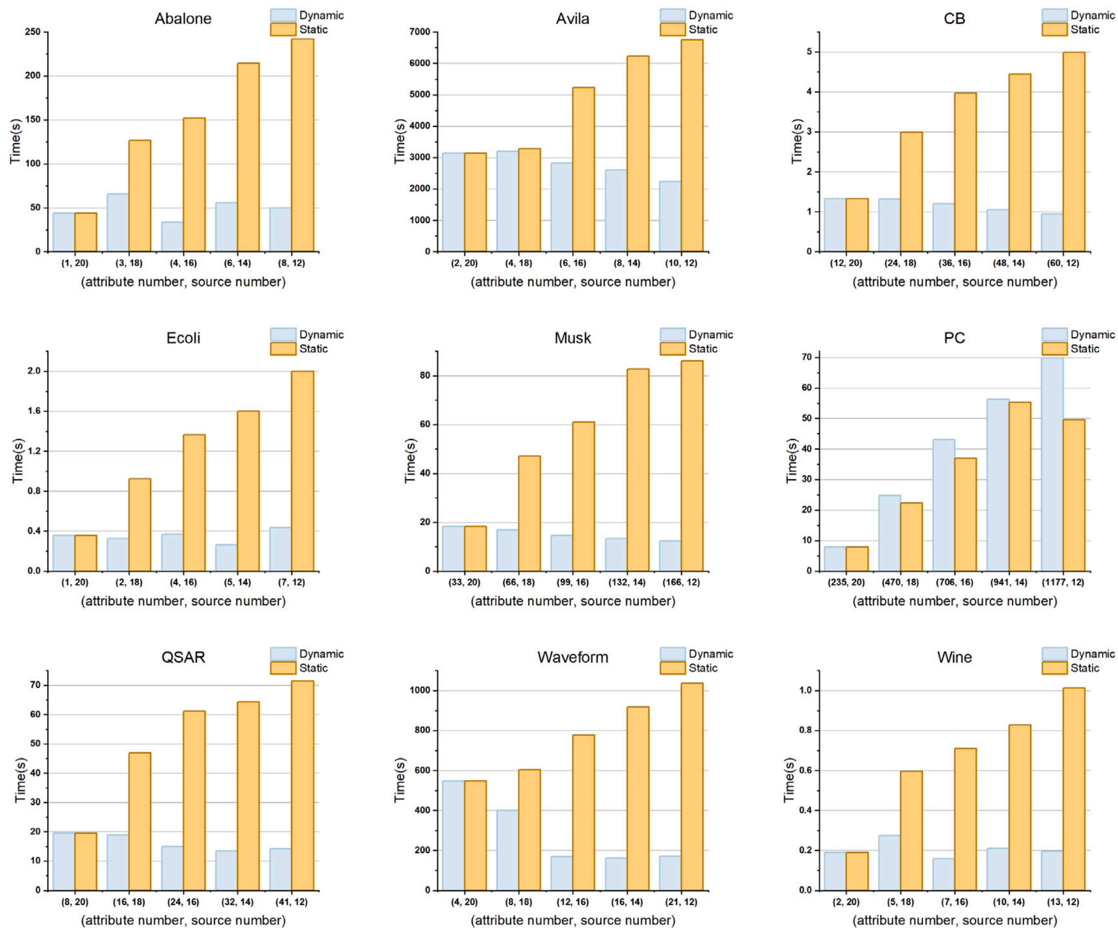

Fig. 6. The time Static and Dynamic methods use in case of addition of attributes and deletion of sources.

between 60% and 80%, but its performance declines at a 40% selection rate. This indicates that our feature selection method is more suited for selecting over half of the original features, while it struggles when fewer features are selected. During the experiments, we also observed that using single-source feature selection algorithms to select features from each source individually and then merging the results takes a considerable amount of time. In comparison, our multi-source feature selection method requires significantly less time while still outperforming the aforementioned methods in most cases. This also demonstrates that our method greatly enhances the efficiency of multi-source feature selection.

5.2.3. The analysis of the efficiency of dynamic information fusion and feature selection method

From Fig. 5-Fig. 8, it can be seen that our dynamic method can reduce program runtime to a certain extent. Comparing the performance of the same dataset in four different scenarios, we can see that our dynamic method's effect is particularly evident in case c. The ratios for the two larger datasets Avila and Waveform are relatively high, indicating that the dynamic method's effect is more pronounced when the dataset has a larger sample size. However, observing the ratios for datasets with many features like Pendigits, there are a few ratios less than 1, indicating that the dynamic method may have a disadvantage when dealing with datasets with many features but few samples.

6. Conclusion and contribution

In conclusion, we proposed a multi-source static and dynamic information fusion and feature selection method based on PROMETHEE. For the static information fusion and feature selection method, we first defined the AEM to evaluate the merits of each attribute. Then, we defined the APDM to merge the AEM of various sources, thereby obtaining the preference degrees of each attribute relative to others. Additionally, we defined the net flow, which comprehensively considers the preferences of each attribute for others, and used the net flow to rank attributes, achieving the goal of feature selection. Finally, we propose an information fusion method that calculates the source center and uses it to find the optimal information source. For the dynamic information fusion and feature selection method, we firstly introduce the dynamic information fusion method. Then, we considered four scenarios and provided dynamic algorithms for each scenario, enabling us to handle changes in both attribute and source quantities.

The final experimental section evaluated our static and dynamic information fusion and feature selection methods using nine datasets. On the static side, we first fixed the threshold alpha for varying feature selection rates and observed the algorithm's performance by inputting the results into KNN, SVM and DT. Additionally, we fixed the threshold alpha for varying feature selection rates and also inputted the results into KNN, SVM and DT to observe the outcome. We also compare our method with other four feature selection methods with classification accuracy to confirm the efficiency of our method. On the dynamic side, we compared the runtime of the dynamic algorithms in four scenarios with that of the static algorithm. From the comparison, we found that our dynamic algorithm significantly improves feature selection efficiency when source variations occur.

For the contribution, This paper's theory addresses the problem of multi-source feature selection. As mentioned in the introduction, solving the multi-source feature selection problem is quite challenging due to the limited availability of methods in this area. Our proposed theory ensures that feature selection results are based on the original

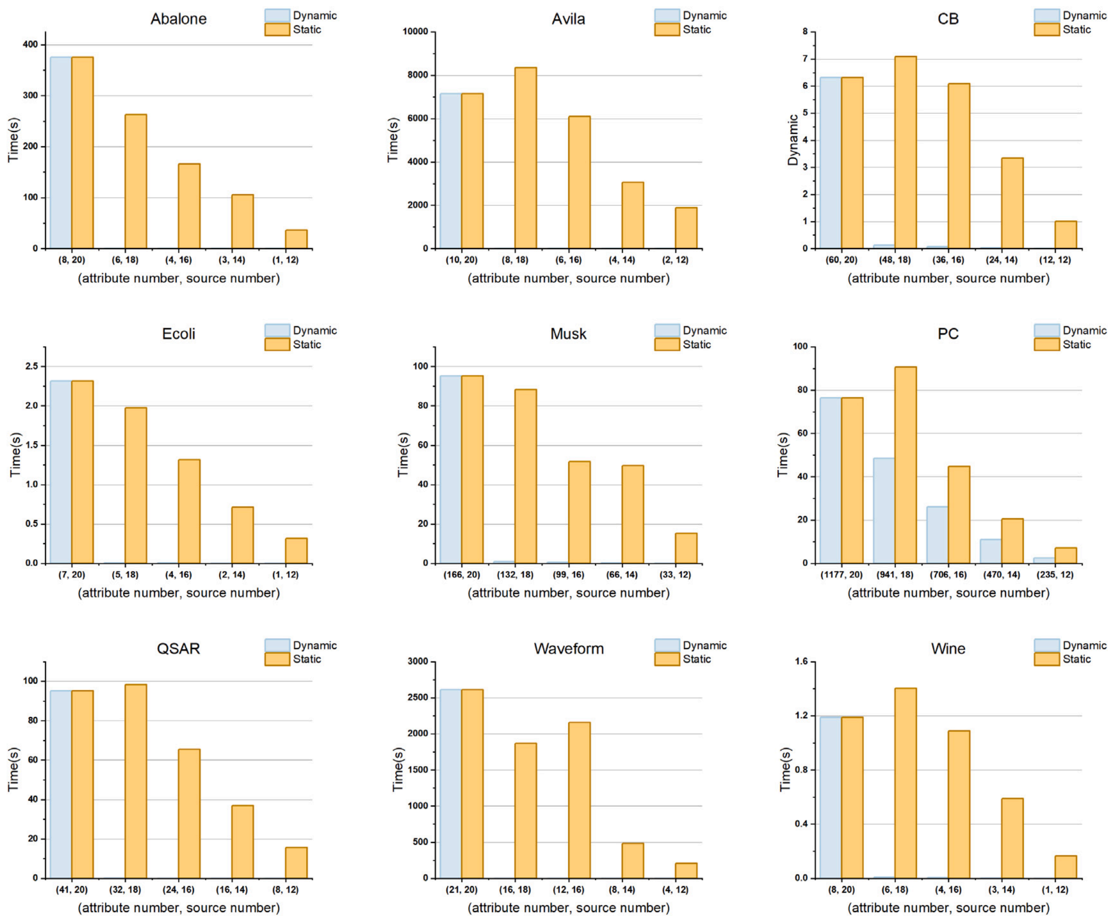

Fig. 7. The time Static and Dynamic methods use in case of deletion of attributes and sources.

data, saves a significant amount of time, and yields effective feature selection outcomes. In real-world big data environments, multi-source feature selection is also a common issue. For instance, when evaluating something annually, each year's data can serve as an information source, forming a multi-source information system. If feature selection and information fusion are needed for such a system, our static feature selection and information fusion methods can be effectively applied. Additionally, as the evaluation data for each year changes, our dynamic feature selection and information fusion methods can greatly reduce the time required, making them highly efficient for such scenarios.

7. Future work

In this section, we will discuss about the limitations and the future work, which are listed below:

• Our static method is designed specifically for multi-source single-value information systems. Further research is needed to address these different forms of information systems which is common in practical applications and develop appropriate feature selection methods tailored to each type.

• Our dynamic method currently only accommodates changes in the number of information sources and the number of attributes, while it cannot handle changes in the number of samples. Therefore, our dynamic method requires further improvement to adapt to all variations in information systems, including changes in the number of samples.

• In the dynamic experiments, we discovered that our dynamic method did not perform well when handling datasets with many features but few samples. Therefore, our next goal is to continue improving the dynamic algorithm to ensure it performs well even with datasets that have many features but few samples.

CRediT authorship contribution statement

Weihua Xu: Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Yigao Li:** Writing – review & editing, Writing – original draft, Visualization, Software, Data curation.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 62376229, and Natural Science Foundation of Chongqing, China under Grant CSTB2023NSCQ-LZX0027.

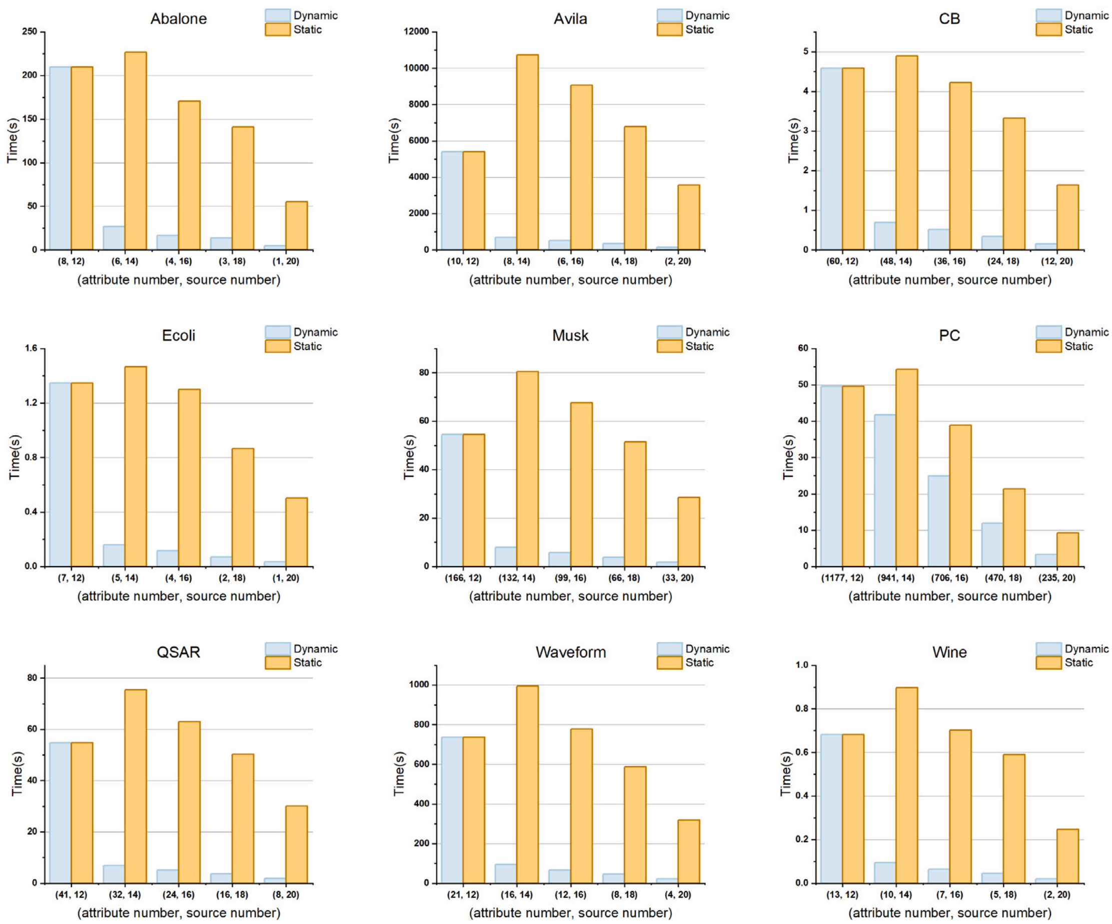

Fig. 8. The time Static and Dynamic methods use in case of deletion of attributes and addition of sources.

Data availability

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

References

[1] B. Remeseiro, V. Bolon-Canedo, A review of feature selection methods in medical applications, Comput. Biol. Med. 112 (2019) 103375.

[2] Z. Zainuddin, K.H. Lai, P. Ong, An enhanced harmony search based algorithm for feature selection: Applications in epileptic seizure detection and prediction, Comput. Electr. Eng. 53 (2016) 143–162.

[3] X. Li, J. Zhang, F. Safara, Improving the accuracy of diabetes diagnosis applications through a hybrid feature selection algorithm, Neural Process. Lett. 55 (1) (2023) 153–169.

[4] B. Omar, F. Rustam, A. Mehmood, G.S. Choi, et al., Minimizing the overlapping degree to improve class-imbalanced learning under sparse feature selection: application to fraud detection, IEEE Access 9 (2021) 28101–28110.

[5] X. Liu, S. Wang, S. Lu, Z. Yin, X. Li, L. Yin, J. Tian, W. Zheng, Adapting feature selection algorithms for the classification of Chinese texts, Systems 11 (9) (2023) 483.

[6] S. Maldonado, J. Pérez, C. Bravo, Cost-based feature selection for support vector machines: An application in credit scoring, European J. Oper. Res. 261 (2) (2017) 656–665.

[7] A. Jović, K. Brkić, N. Bogunović, A review of feature selection methods with applications, in: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, IEEE, 2015, pp. 1200–1205.

[8] J. Wang, P. Zhao, S.C. Hoi, R. Jin, Online feature selection and its applications, IEEE Trans. Knowl. Data Eng. 26 (3) (2013) 698–710.

[9] M. Hu, E.C. Tsang, Y. Guo, D. Chen, W. Xu, Attribute reduction based on overlap degree and k-nearest-neighbor rough sets in decision information systems, Inform. Sci. 584 (2022) 301–324.

[10] W. Li, S. Zhai, W. Xu, W. Pedrycz, Y. Qian, W. Ding, T. Zhan, Feature selection approach based on improved fuzzy c-means with principle of refined justifiable granularity, IEEE Trans. Fuzzy Syst. (2022).

[11] A.A. Alhussan, A.A. Abdelhamid, E.S.M. El-Kenawy, A. Ibrahim, M.M. Eid, D.S. Khafaga, A.E. Ahmed, A binary waterwheel plant optimization algorithm for feature selection, IEEE Access (2023).

[12] N. Khodadadi, E. Khodadadi, Q. Al-Tashi, E.S.M. El-Kenawy, L. Abualigah, S.J. Abdulkadir, A. Alqushaibi, S. Mirjalili, BAOA: binary arithmetic optimization algorithm with K-nearest neighbor classifier for feature selection, IEEE Access 11 (2023) 94094–94115.

[13] N. Ganesh, R. Shankar, R. Čep, S. Chakraborty, K. Kalita, Efficient feature selection using weighted superposition attraction optimization algorithm, Appl. Sci. 13 (5) (2023) 3223.

[14] A.A. Abdelhamid, E.S.M. El-Kenawy, A. Ibrahim, M.M. Eid, D.S. Khafaga, A.A. Alhussan, S. Mirjalili, N. Khodadadi, W.H. Lim, M.Y. Shams, Innovative feature selection method based on hybrid sine cosine and dipper throated optimization algorithms, IEEE Access 11 (2023) 79750–79776.

[15] W. Xu, M. Huang, Z. Jiang, Y. Qian, Graph-based unsupervised feature selection for interval-valued information system, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[16] W. Li, H. Zhou, W. Xu, X.Z. Wang, W. Pedrycz, Interval dominance-based feature selection for interval-valued ordered data, IEEE Trans. Neural Netw. Learn. Syst. (2022).

[17] J. Dai, Y. Liu, J. Chen, X. Liu, Fast feature selection for interval-valued data through kernel density estimation entropy, Int. J. Mach. Learn. Cybern. 11 (2020) 2607–2624.

[18] W.Z. Wu, D. Niu, J. Li, T.J. Li, Rule acquisition in generalized multi-scale information systems with multi-scale decisions, Internat. J. Approx. Reason. 154 (2023) 56–71.

[19] Z. Huang, J. Li, Multi-scale covering rough sets with applications to data classification, Appl. Soft Comput. 110 (2021) 107736.

[20] S. Nemani, D. Cote, B. Misiuk, E. Edinger, J. Mackin-McLaughlin, A. Templeton, J. Shaw, K. Robert, A multi-scale feature selection approach for predicting benthic assemblages, Estuar. Coast. Shelf Sci. 277 (2022) 108053.

[21] H.T. Shen, Y. Zhu, W. Zheng, X. Zhu, Half-quadratic minimization for unsupervised feature selection on incomplete data, IEEE Trans. Neural Netw. Learn. Syst. 32 (7) (2020) 3122–3135.

[22] A.I. Maghsoodi, A.E. Torkayesh, L.C. Wood, E. Herrera-Viedma, K. Govindan, A machine learning driven multiple criteria decision analysis using LS-SVM feature

elimination: sustainability performance assessment with incomplete data, Eng. Appl. Artif. Intell. 119 (2023) 105785.
[23] L. Sun, L. Wang, Y. Qian, J. Xu, S. Zhang, Feature selection using Lebesgue and entropy measures for incomplete neighborhood decision systems, Knowl.-Based Syst. 186 (2019) 104942.
[24] C. Luo, T. Li, Z. Yi, An incremental feature selection approach based on information entropy for incomplete data, in: 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, DASC/PiCom/CBDCom/CyberSciTech, IEEE, 2019, pp. 483–488.
[25] C. Fahy, S. Yang, Dynamic feature selection for clustering high dimensional data streams, IEEE Access 7 (2019) 127128–127140.
[26] S. Li, L. Yang, J. Huang, X.S. Hua, L. Zhang, Dynamic anchor feature selection for single-shot object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6609–6618.
[27] W. Shu, W. Qian, Y. Xie, Incremental approaches for feature selection from dynamic data with the variation of multiple objects, Knowl.-Based Syst. 163 (2019) 320–331.
[28] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, J. Saeed, A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction, J. Appl. Sci. Technol. Trends 1 (1) (2020) 56–70.
[29] B. Venkatesh, J. Anuradha, A review of feature selection and its methods, Cybern. Inf. Technol. 19 (1) (2019) 3–26.
[30] N. Pudjihartono, T. Fadason, A.W. Kempa-Liehr, J.M. O'Sullivan, A review of feature selection methods for machine learning-based disease risk prediction, Front. Bioinform. 2 (2022) 927312.
[31] W. Xu, K. Cai, D.D. Wang, A novel information fusion method using improved entropy measure in multi-source incomplete interval-valued datasets, Internat. J. Approx. Reason. 164 (2024) 109081.
[32] X. Zhang, X. Chen, W. Xu, W. Ding, Dynamic information fusion in multi-source incomplete interval-valued information system with variation of information sources and attributes, Inform. Sci. 608 (2022) 1–27.
[33] Y. Tang, Y. Chen, D. Zhou, Measuring uncertainty in the negation evidence for multi-source information fusion, Entropy 24 (11) (2022) 1596.
[34] P. Zhang, T. Li, G. Wang, C. Luo, H. Chen, J. Zhang, D. Wang, Z. Yu, Multi-source information fusion based on rough set theory: A review, Inf. Fusion 68 (2021) 85–117.
[35] J.A. Massignan, J.B. London, M. Bessani, C.D. Maciel, R.Z. Fannucchi, V. Miranda, Bayesian inference approach for information fusion in distribution system state estimation, IEEE Trans. Smart Grid 13 (1) (2021) 526–540.
[36] F. Xiao, GEJS: A generalized evidential divergence measure for multisource information fusion, IEEE Trans. Syst. Man Cybern. A 53 (4) (2022) 2246–2258.
[37] Z. Pawlak, Rough set theory and its applications, J. Telecommun. Inf. Technol. (2002) 7–10.
[38] J.P. Brans, P. Vincke, B. Mareschal, How to select and how to rank projects: The PROMETHEE method, European J. Oper. Res. 24 (2) (1986) 228–238.
[39] J. Deng, J. Zhan, W.Z. Wu, A ranking method with a preference relation based on the PROMETHEE method in incomplete multi-scale information systems, Inform. Sci. 608 (2022) 1261–1282.
[40] W. Xu, K. Yuan, W. Li, W. Ding, An emerging fuzzy feature selection method using composite entropy-based uncertainty measure and data distribution, IEEE Trans. Emerg. Top. Comput. Intell. 7 (1) (2022) 76–88.
[41] Y. Pan, W. Xu, Q. Ran, An incremental approach to feature selection using the weighted dominance-based neighborhood rough sets, Int. J. Mach. Learn. Cybern. 14 (4) (2023) 1217–1233.