Contents lists available at ScienceDirect

Information Fusion

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

Full length article

Scalable data fusion via a scale-based hierarchical framework: Adapting to multi-source and multi-scale scenarios

Xiaoyan Zhang [*], Jiajia Lin

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

ARTICLE INFO

Keywords:
Data fusion
Information gain
Multi-source integration
Scale-based tree

ABSTRACT

Multi-source information fusion addresses challenges in integrating and transforming complementary data from diverse sources to facilitate unified information representation for centralized knowledge discovery. However, traditional methods face difficulties when applied to multi-scale data, where optimal scale selection can effectively resolve these issues but typically lack the advantage of identifying the optimal and simplest data from different data source relationships. Moreover, in multi-source, multi-scale environments, heterogeneous data (where identical samples have different features and scales in different sources) is prone to occur. To address these challenges, this study proposes a novel approach in two key stages: first, aggregating heterogeneous data sources and refining datasets using information gain; second, employing a customized **S**cale-**b**ased **T**ree (SbT) structure for each attribute to help extract specific scale information value, thereby achieving effective data fusion goals. Extensive experimental evaluations cover ten different datasets, reporting detailed performance across multiple metrics including **A**pproximation **P**recision (AP), **A**pproximation **Q**uality (AQ) values, classification accuracy, and computational efficiency. The results highlight the robustness and effectiveness of our proposed algorithm in handling complex multi-source, multi-scale data environments, demonstrating its potential and practicality in addressing real-world data fusion challenges.

1. Introduction

Multi-source fusion technology has practical and real-world applications, including environmental monitoring, intelligent transportation systems, healthcare, and urban planning and management [1–3]. In data fusion, various methodologies can merge separate multi-source data [4–6]. One strategy involves using multi-granularity information fusion and researching how to merge information from multiple sources using multi-granularity rough sets. This method primarily entails extracting adequate information from multi-source information systems using multi-granularity concepts [7–11] and doing computations at each granularity to address fusion challenges across varying multi-source information systems. For instance, in 2023, Zhang et al. [12] established an information fusion approach utilizing multi-granularity fusion operators' matrix operations. Another approach involves utilizing evidence theory. The Dempster–Shafer evidence theory [13,14] helps handle unknowable data fusion. Its fundamental principle is to apply belief functions for evidence reasoning and synthesis, making essential improvements to information fusion applications in fields such as artificial intelligence [15], pattern recognition [16], and diagnostic detection [17]. In 2017, Liu et al. [18] proposed an architecture based on the weighted fuzzy Dempster–Shafer framework to adjust the weights of inconsistent evidence generated by different classification methods, thereby realizing a multi-modal information fusion system. In 2020, Chen et al. [19] introduced a data fusion method based on weighted belief entropy and negated primary probability assignment to gauge the relative importance of evidence. In 2021, Wang et al. [20] considered employing quality function trust metrics and rationality metrics to reflect the relevance of different types of subsets and proposed a novel multi-source data fusion method. Additionally, a fusion methodology based on information entropy exists. These methodologies utilize information entropy [21] to measure the information content of each information source, aiming to resolve information asymmetry issues within multi-source data. This facilitates the acquisition and fusion of information or rules from multiple origins. For instance, in 2022, Zhang et al. [22] devised a dynamic fusion method tailored for incomplete interval-valued information systems. In 2023, Xu et al. [23] presented a novel dynamic interval-valued information fusion method. In 2024, Cai et al. [24] enhanced interval-valued information fusion by designing enhanced information entropy based on statistical distribution principles and KL divergence, Chen et al. [25] proposed a dynamic information fusion method for interval-valued data with changing objects. Even though many methods can integrate multiple data sources, they are not suitable for multiscale data.

[*] Corresponding author.
E-mail addresses: zhangxiaoyan@swu.edu.cn (X. Zhang), pnkaka8007@email.swu.edu.cn (J. Lin).

https://doi.org/10.1016/j.inffus.2024.102694
Received 2 July 2024; Received in revised form 30 August 2024; Accepted 10 September 2024
Available online 12 September 2024

Table 1
The summary of existing methods.

Methods	Categories	Disadvantages
Multi-source methods	Using multi-granularity information fusion [12] Utilizing evidence theory [18–20] Utilizing information entropy [22–25]	Not applicable to multi-scale data
Optimal scale selection methods	Using attribute significance [37] Handling dynamic multi-scale data [38]	Only one scale can be chosen for the same feature, multiple scales cannot be selected
	Optimal scale cuts [39]	Multiple scales can be chosen for the same feature, but it cannot handle multi-source heterogeneous data
Heterogeneous data methods	Convert into single-type data [42,43] Utilizing different rough set models [44–47]	Not combined with methods of multi-source fusion and multi-scale approaches.

There is no method to integrate multi-source, multi-scale heterogeneous data.

Based on specific multi-scale task requirements [26–28], the most appropriate scale for analyzing and processing multi-scale data can be selected using various methods, such as information granularity [29–31], feature selection [32,33], optimization algorithms [34] and others [35,36]. In 2017, Li et al. [37] introduced an optimal scale selection technique for multi-scale tables that combines attribute importance to achieve optimal selection. In the same year, Hao et al. [38] presented an approach that tackles local and global concerns in choosing the most suitable scale in dynamic multi-scale decision systems. This method also adjusts the ideal scale when objects are modified. In 2021, She et al. [39] proposed a strategy that leverages granularity trees for optimal the selection of cuts in totally multi-scale decision tables. In the same year, Bao et al. [40] presented an entropy-based approach to choose the most suitable mix of scales for generalized multi-scale information tables. In 2023, Zhang et al. [41] introduced a technique utilizing generalized multi-scale decision tables to achieve optimal scale selection. This method aims to alleviate limitations on decision attributes. Traditional multiscale methods typically only allow for selecting a single scale of data for different features, although in some cases, they can select different scales for the same feature [39]. However, in multi-source, multi-scale environments, heterogeneous data is prone to occur.

In real life, a large amount of heterogeneous data composed of numerical and categorical data exist. Early methods for handling such data involved converting it into a single numerical type [42,43]. Currently, neighborhood rough set models are widely used for feature selection and data processing. In 2008, Hu et al. [44] introduced the concept of domains and extended equivalence classes using distance functions to simplify the handling of heterogeneous data. In 2016, Zhang et al. [45] proposed a method using fuzzy rough entropy for feature selection of heterogeneous data, but this method had poor generalization capabilities. In 2024, Zhang et al. [46] optimized feature selection of heterogeneous data using domain combination entropy. Simultaneously, Dai et al. [47] introduced an improved domain combination entropy method specifically designed for handling heterogeneous data.

Table 1 has categorized three different data processing methods according to their respective characteristics. There are various approaches for handling multi-source data fusion. however, they do not effectively address multiscale data. On the other hand, many methods can handle multiscale data, but they are not suitable for dealing with multi-source heterogeneous scenarios. Heterogeneous data is only considered in our discussion when involving multiple sources. Currently, Existing methods do not have specialized countermeasures available for multi-source, multiscale heterogeneous data. In the integration process, these issues are taken into account.

1. The untapped potential of multi-source multi-scale data: In the process of integrating multi-source data, the fusion of multi-scale data has not been explored. Due to features having different scale information, existing information entropy methods may not be suitable for multi-scale structural models.
2. Neglecting heterogeneous features in multi-source multi-scale data: In multi-source multi-scale data, heterogeneous features arise due to changes in related structures. For example, the same feature of identical samples from different sources may vary in quantity scales, and identical samples from different sources may possess different features. These are all issues that require attention.
3. How to segment scales: Existing traditional methods for optimal scale selection typically involve choosing the same scale for all scales. For instance, selecting the same scale for all samples under a certain feature may lead to biases in data selection and evaluation.

This work introduces an innovative method for fusing multi-source multi-scale data using scale-based tree (SbT), solving the drawbacks of existing fusion approaches, as seen in Fig. 1. The proposed approach consists of two parts: rough selection and precise fusion. In the rough selection stage, information gain (IG) [48] is employed to select important information from each source, generating an optimal datasets, namely the standard multi-scale data. The fusion stage in question uses the attributes of SbT to make exact selections based on scale. The SbT selection method differs from many approaches by enabling the option of several scales for a single attribute, as opposed to selecting only one scale. This results in the extraction of the most optimal combined data. The primary advancements and contributions of this work are as follows:

1. Proposing an innovative approach for multi-source, multi-scale data fusion.
2. Applying IG to multi-source data fusion enables fast fusion and significantly reduces fusion time.
3. Utilizing SbT for precise scale selection allows for different scales for the same attribute. The final fused data can be represented in tuple-based and interval-based.
4. Experimental evaluations conducted on UCI datasets demonstrate the proposed fusion method's significant improvements in classification accuracy and time efficiency.

The paper is organized as follows. Section 2 provides an overview of foundational information, thoroughly defining and clarifying important ideas and terminologies. Section 3 introduces a data fusion architecture combining several sources and data scales using SbT. The system is separated into two main components, each with its algorithms. Section 4 subsequently assesses the efficacy and efficiency of the suggested approach by conducting experiments on the UCI dataset. Section 5 provides a concise overview of the primary discoveries of the study, shows the recognized limitations, and outlines possible paths for further research. Furthermore, Table 2 is included to list and explain all the terminology and abbreviations used in the work, attempting to enhance the readers' comprehension.

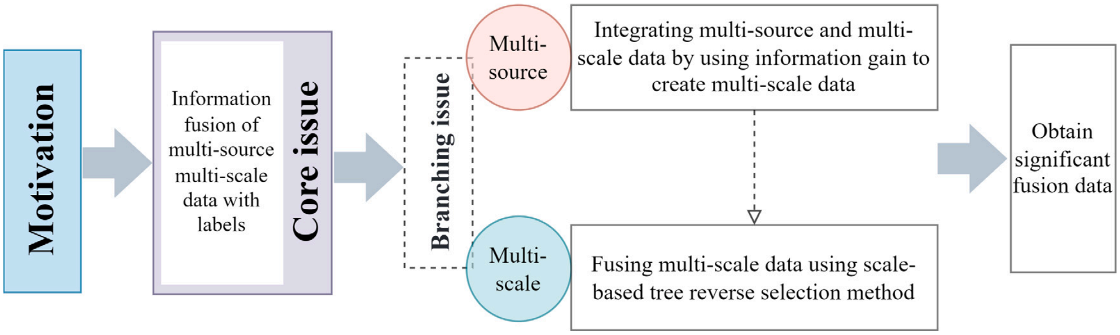

Motivation
Information fusion of multi-source multi-scale data with labels
Core issue
Branching issue
Multi-source
Integrating multi-source and multi-scale data by using information gain to create multi-scale data
Multi-scale
Fusing multi-scale data using scale-based tree reverse selection method
Obtain significant fusion data

Fig. 1. Motivation for our work.

Table 2
The abbreviations of terminologies.

Abbreviations	Terminologies	References
IG	Information gain	[48]
SbT	Scale-based tree	[39]
IT	Information table	[39]
DT	Decision table	[39]
AP	Approximation precision	[25]
AQ	Approximation quality	[22]
Ms-DT	Multi-source decision table	[25]
MsIT	Multi-scale information table	[39]
MsDT	Multi-scale decision table	[39]
Ms-MsDT	Multi-source multi-scale decision table	

Ms-MsDT combines multiple sources and scales.

2. Preliminaries

This part provides a gradual introduction to information table (IT), decision table (DT), multi-source decision table (Ms-DT), multi-scale information table (MsIT), multi-scale decision table (MsDT), multi-source multi-scale decision table (Ms-MsDT), and also includes an introduction to SbT.

2.1. Multi-source multi-scale information table

IT can be denoted as a triplet (O, A, F), where $O = \{o_1, o_2, \ldots, o_p\}$ is an object set, $A = \{a_1, a_2, \ldots, a_q\}$ is an attribute set, $F = \{f(o,a) \mid o \in O, a \in A\}$ is relationship set between O and A, $f(o,a)$ represents the actual value of the sample o on that attribute a.

Then, DT can be denoted as a quadruple $(O, A \bigcup \{d\}, F, G)$, where d is a decision attribute, $G = \{g(o,d) \mid o \in O\}$ is relationship set between O and d, $g(o,d)$ represents the classification label of the sample o on that decision attribute d.

For any $A_{sub} \subseteq A$, $R_{A_{sub}} \subseteq O \times O$ can be defined, it indicates that there is a relationship set $R_{A_{sub}}$ between two objects on the attribute set A_{sub}. And for any $o_1 \in O$, the relation class of o_1 is denoted as $[o_1]^{R_{A_{sub}}} = \{o_2 \in O \mid (o_1, o_2) \in R_{A_{sub}}\}$.

Definition 2.1. Given IT as a triplet (O, A, F), $O_{sub} \subseteq O$, $A_{sub} \subseteq A$, the lower and upper approximations of O_{sub} with respect to A_{sub} are denoted as

$$\underline{R_{A_{sub}}}(O_{sub}) = \{o \in O \mid [o]^{R_{A_{sub}}} \subseteq O_{sub}\}, \tag{1}$$

$$\overline{R_{A_{sub}}}(O_{sub}) = \{o \in O \mid [o]^{R_{A_{sub}}} \cap O_{sub} \neq \emptyset\}. \tag{2}$$

Definition 2.2. Given DT as a quadruple $(O, A \bigcup \{d\}, F, G)$, let $O/d = \{D_1, D_2, \ldots, D_m\}$ be the decision partition of O, where $D_m = \{o \in O \mid g(o,d) = d_m\}$ denotes the collection of objects for the mth decision label d_m. For any $A_{sub} \subseteq A$, the approximation precision (AP) and approximation quality (AQ) with d is defined as

$$AP_{R_{A_{sub}}}(O/d) = \frac{\sum_{D \in O/d} |\underline{R_{A_{sub}}}(D)|}{\sum_{D \in O/d} |\overline{R_{A_{sub}}}(D)|}, \tag{3}$$

$$AQ_{R_{A_{sub}}}(O/d) = \frac{\sum_{D \in O/d} |\underline{R_{A_{sub}}}(D)|}{|O|}. \tag{4}$$

AP and AQ are metrics that quantify the accuracy of models in classifying data. They are frequently employed as assessment measures in rough set theory. Higher AP and AQ values show superior effectiveness in classifying approximations.

Ms-DT is defined as $\{DT_i \mid DT_i = (O, A_i \bigcup \{d_i\}, F_i, G_i), i = 1, 2, \ldots, s\}$, where DT_i represents ith DT of Ms-DT.

MsIT can be denoted as a triplet $(O, \{a_i^j, i \in 1, 2, \ldots, q, j \in 1, 2, \ldots, t\}, \{f(o,a) \mid o \in O, a \in A\})$, where a_i^j represents the jth scale of the ith attribute.

Then MsDT is defined as a quadruple $(O, \{a_i^j, i = 1, 2, \ldots, q, j = 1, 2, \ldots, t\} \bigcup \{d\}, \{f(o,a) \mid o \in O, a \in A\}, \{g(o,d) \mid o \in O\})$.

So, Ms-MsDT is defined as $\{MsDT_i \mid MsDT_i = (O, A_i \bigcup \{d_i\}, F_i, G_i), i = 1, 2, \ldots, s\}$.

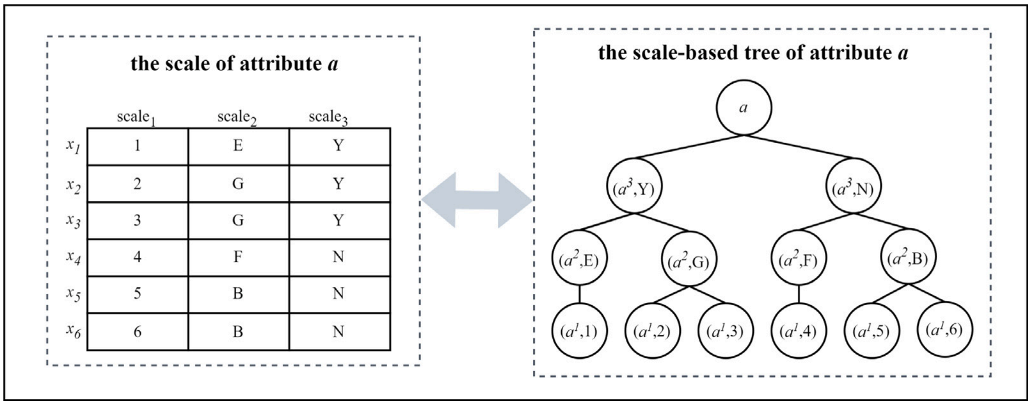
2.2. Scale-based tree

Within the Ms-MsDT framework, a tree structure can be used to express all the scales associated with each attribute. The attribute name is the tree's root, while the nodes are binary tuples. These tuples consist of a scale and all values that have appeared on this scale. The tree demonstrates a top-down progression, with the scales becoming increasingly finer as one moves downwards. SbT can be used to denote the granularity tree mentioned in the study by [39]. Next, the definition of SbT will be provided.

Definition 2.3. Given an MsDT as a quadruple $(O, \{a_i^j, i = 1, 2, \ldots, q, j = 1, 2, \ldots, t\} \bigcup \{d\}, \{f(o,a) \mid o \in O, a \in A\}, \{g(o,d) \mid o \in O\})$, an SbT can be constructed for an attribute a as follows.

(1) SbT has one root node labeled by a.
(2) All nodes except the root node are represented as binary tuples composed of scale and actual value. For example, the child node of the root node is denoted as $\{(t, f(o, a^t)) | o \in O\}$.
(3) All nodes, except the root node, have the following parent–child relationships. If the parent node is defined as $(j, f(o_1, a^j))$, $o_1 \in O$, then its child node is $\{(j-1, f(o, a^{j-1})) \mid f(o, a^j) = f(o_1, a^j), o \in O\}$. If the child node is defined as $(j, f(o_1, a^j))$, then its parent node is $(j+1, f(o_1, a^{j+1}))$.

Example 2.1. Fig. 2 demonstrates an instance of the conversion process from data to SbT. The graphic illustrates the data for attribute



	scale_1	scale_2	scale_3
x_1	1	E	Y
x_2	2	G	Y
x_3	3	G	Y
x_4	4	F	N
x_5	5	B	N
x_6	6	B	N

Fig. 2. Example of transformation between data and SbT.

a on the left side, including values for six objects across three scales. On the right side, the matching SbT for a is illustrated. The tree's root represents the attribute's name, while the leaf nodes are supplied as tuples. The values assigned to each node in a layer correspond to each scale.

3. Our approach

At this stage, information fusion involves two consecutive processes as depicted in Fig. 3. Firstly, we introduce an original dataset called $Ms - MsDT$, composed of multiple sources $MsDT_1$, $MsDT_2$, …, $MsDT_s$. Each source contains the same samples and decision attributes, but they may differ in terms of attributes and scales, exhibiting heterogeneous properties. Next, information gain calculations are performed separately for each source. For instance, in $MsDT_1$, we compute the information gain for each attribute across different scales and select the scale with the maximum information gain for each attribute. The selected data undergoes processing to form IT_1, IT_2, …, IT_s, which are then combined with the original decision attribute d to obtain a rough selection of the data $MsDT$. This process is relatively quick while retaining the maximum information content from the original dataset $Ms - MsDT$.

Next is the precise fusion process. For each feature in $MsDT$, scale-based trees SbT_1, SbT_2, …, SbT_s are built based on their scale values. These tree models help in better understanding the distribution of $MsDT$ data. Using coherence judgments, we reverse-select each SbT to choose appropriate scale values for different objects within the same attribute. This helps simplify data representation while maintaining accuracy. In this process, the form of the scale-based tree we use is derived from [39], but the method of cutting the scales is completely different from that in [39], where traditional granularity computing methods for judging consistency are employed.

Finally, combining the results yields data in tuple or interval value forms. Tuple form data is represented as each scale paired with its corresponding scale value in a tuple. Interval form data shows the range of values each scale can take.

3.1. Rough selection

Provided a MsDT represented as a quadruple $(O, \{a_i^j, i = 1,2,\dots,q, j = 1,2,\dots,t\} \bigcup d, \{f(o,a) \mid o \in O, a \in A\}, \{g(o,d) \mid o \in O\})$, the task is to calculate the IG for each attribute at every scale in a given information source. IG is a metric employed in decision trees and other machine-learning methods to choose features. The measure estimates the information acquired about a class variable (O/d) from observing a feature (a scale of an attribute). Below are some definitions linked to IG.

Definition 3.1. The entropy of O/d represents the average amount of information needed to classify a randomly chosen object in dataset. It can be calculated using the following formula:

$$H(O/d) = -\sum_{i\in O/d} P(i)\log_2 P(i), \tag{5}$$

where $P(i) = \dfrac{|i|}{|O|}$.

Proposition 3.1. *The entropy of O/d has those properties.*

(1) *Non-negativity: $H(O/d) \geq 0$, because conditional entropy is fundamentally a measure of information, and information cannot be negative.*
(2) *Maximizing Entropy Principle: $H(O/d) \leq \log_2 n$, where n is the number of possible values of the random variable O/d.*

Definition 3.2. The conditional entropy of O/d given a_i^j represents the entropy of O/d under the condition of knowing scale a_i^j. It can be calculated using the following formula:

$$H(O/d \mid a_i^j) = \sum_{k\in O/a_i^j} P(k)H(O/a_i^j), \tag{6}$$

where $P(k) = \dfrac{|k|}{|O|}$.

Proposition 3.2. *The conditional entropy of O/d given a_i^j has those properties.*

(1) *Non-negativity: $H(O/d \mid a_i^j) \geq 0$, because conditional entropy is fundamentally a measure of information, and information cannot be negative.*
(2) *Symmetry: Conditional entropy exhibits symmetry when exchanging conditional variables and target variables, meaning $H(O/d \mid a_i^j)$ can equivalently be expressed as $H(O/a_i^j \mid d)$. This symmetry arises from the definition of conditional entropy itself, which depends symmetrically on conditional variables and the target variable.*
(3) *Extremal Property: Conditional entropy achieves extreme values under specific minimum or maximum conditions. For instance, in the case of discrete distributions, conditional entropy reaches its minimum when a_i^j is completely correlated with d; it reaches its maximum when they are completely independent.*

Definition 3.3. Information gain $IG(a_i^j)$ represents the information about O/d gained by observing a_i^j. It can be calculated using the following formula:

$$IG(a_i^j) = H(O/d) - H(O/d \mid a_i^j), \tag{7}$$

$H(O/d)$ is O/d entropy and $H(O/d \mid a_i^j)$ is the conditional entropy given a_i^j.

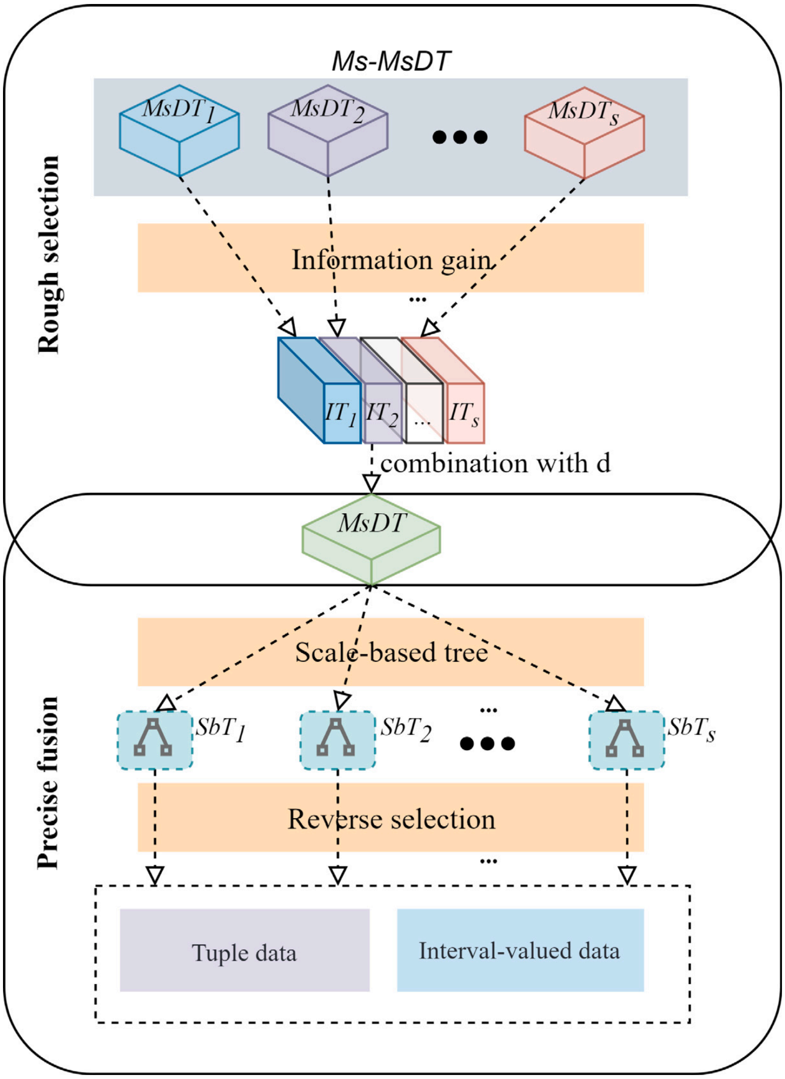


Fig. 3. The architecture of our work.

Proposition 3.3. *IG has those properties.*

(1) *Non-negativity: IG is always non-negative, implying that by observing a_i^j, information about O/d is always gained; at the very least, losing information is impossible.*
(2) *Dependence on Scale Scope: The magnitude of IG depends on the choice of a_i^j. Typically, if a_i^j can better assist in distinguishing O/d, then its information gain will be larger.*
(3) *Relationship with Entropy: $IG(a_i^j)$ can be viewed as the overall entropy $H(O/d)$ minus the conditional entropy $H(O/d \mid a_i^j)$. Therefore, IG describes the uncertainty that can be reduced by observing a_i^j.*

To assess the significance of a_i^j in O/d, $IG(a_i^j)$ can be calculated by following these steps. Within the MsDT framework, IG can be computed for every attribute at every scale. This allows us to determine the specific scale within each attribute that results in the highest IG. The chosen scales are merged to create an IT. Ms-MsDT allows for generating several information tables (ITs) from various sources. Ultimately, these individual ITs are merged to form a comprehensive MsDT, which contains shared decision data. This process is summarized in Algorithm 1.

The time complexity of algorithm 1 can be analyzed as follows:

Algorithm 1: Rough selection algorithm.

```
Input   : $Ms - MsDT = \{MsDT_i \mid MsDT_i = (O, A_i \bigcup d_i, F_i, G_i), i = 1, 2, \dots, s\}$.
Output  : $MsDT$.
begin
    for $i = 1 : s$ do
        obtain $DT_i$
        for each $I = 1 : |A|$ do
            find $a_I^J$, which $\{IG(a_I^J) \mid J = 1, 2, \dots, t\}$ is minist
            // $t$ represents the number of scales for
                $a_I$ .
        end
    end
    for $i = 1 : s$ do
        $MsDT = \{DT_i \mid i = 1, 2, \dots, s\}$
    end
    // The scales of each attribute are arranged
       from fine to coarse.
end
return  : $MsDT$.
```

Table 3
A MsDT example.

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

(1) Outer Loop (First for Loop): This loop iterates from $i = 1$ to s, fetching DT_i each time. Therefore, the time complexity of the outer loop is $O(s)$.
(2) Inner Loop (Second for Loop): For each DT_i, there is a nested loop that iterates over the attribute set A, which consists of $|A|$ attributes. Within each iteration, the algorithm finds the minimum $IG(a_I^J)$ for a specific attribute a_I, where $J = 1, 2, \dots, t$. Here, t denotes the number of scales for attribute a_I. The time complexity of this part is $O(|A| \cdot t)$.

Therefore, the time complexity of the inner loop is $O(|A| \cdot t)$, and since the outer loop executes s times, the overall time complexity of the algorithm is: $O(s \cdot |A| \cdot t)$. Here, s represents the number of data tables in the dataset, $|A|$ denotes the size of the attribute set, and t depends on the scale structure of different attributes and the data processing method.

Example 3.1. As shown in Table 3, an example of MSDT are presented. In this instance, there are ten objects, each with three attributes. Each attribute has three scales arranged from fine to coarse. Every object has a unique decision attribute value. This MSDT is considered as an information source, and the IG of each attribute at each scale is computed. The next step involves the computational process.

Given d, which contains ten data objects of different categories, the category is $O/d = \{C_1, C_2\}$, $C_1 = \{0, 0, 0, 0, 0, 0\}$, $C_2 = \{1, 1, 1, 1\}$.

Apply the formula for Definition 3.1, the value of $H(O/d)$ is 0.970951. Next, IG can be calculated using the entropy computation method. The calculated results are as follows.

(1) $H(O/d)$: 0.970951. $H(O/d \mid a_1^1)$: 0.360964. $IG(a_1^1)$: 0.609987.
(2) $H(O/d)$: 0.970951. $H(O/d \mid a_1^2)$: 0.360964. $IG(a_1^2)$: **0.609987**.
(3) $H(O/d)$: 0.970951. $H(O/d \mid a_1^3)$: 0.8. $IG(a_1^3)$: 0.170951.
(4) $H(O/d)$: 0.970951. $H(O/d \mid a_2^1)$: 0.6. $IG(a_2^1)$: **0.370951**.
(5) $H(O/d)$: 0.970951. $H(O/d \mid a_2^2)$: 0.760964. $IG(a_2^2)$: 0.209987.
(6) $H(O/d)$: 0.970951. $H(O/d \mid a_2^3)$: 0.8. $IG(a_2^3)$: 0.170951.
(7) $H(O/d)$: 0.970951. $H(O/d \mid a_3^1)$: 0.604184. $IG(a_3^1)$: **0.366766**.
(8) $H(O/d)$: 0.970951. $H(O/d \mid a_3^2)$: 0.649022. $IG(a_3^2)$: 0.321928.
(9) $H(O/d)$: 0.970951. $H(O/d \mid a_3^3)$: 0.826466. $IG(a_3^3)$: 0.144484.

When the information gain values for a_1^1 and a_1^2 are equal, we favor larger-scale data due to its ease of interpretation and reduced complexity in data representation, which also saves storage space. In such cases where the information gains are identical, a_1^2 is typically selected as the final result. So according to IG, the final fusion result we can obtain is a_1^2, a_2^1, a_3^1.

3.2. Precise fusion

Multi-scale data can be obtained from different sources according to the rough selection. However, using multi-scale data may result in varying decisions for data with the same attribute in different scale. Hence, it is imperative to preprocess the data and provide labels to differentiate it.

Give an example of multi-scale data under a single attribute. The scales are arranged from fine to coarse. If data at two scales, the finer scale and the coarser scale, cannot form a containment relationship, the finer scale is artificially classified according to the coarser scale to establish a containment relationship.

Definition 3.4. Given a MsDT as a quadruple $(O, \{a_i^j, i = 1, 2, \dots, q, j = 1, 2, \dots, t\} \bigcup \{d\}, \{f(o, a) \mid o \in O, a \in A\}, \{g(o, d) \mid o \in O\})$, for $X \in O/a_i^j$, if $X \nsubseteq Y$ $(\forall Y \in O/d)$, making the following adjustments to them:

$$h(f(o_x, a_i^j)) = f(o_x, a_i^j) \star f(o_x, a_i^{j+1}), \tag{8}$$

$$f(o_x, a_i^j) = h(f(o_x, a_i^j)). \tag{9}$$

$o_x \in X$, $\star$ represents an operation that labels the scale value with the mark of the previous scale to differentiate it.

Proposition 3.4. *Definition 3.4 has those properties.*

(1) *Uniqueness assurance: The defined adjustments ensure that the elements in adjacent scales a_i^j and a_i^{j+1} from coarse to fine are unique. This prevents data duplication or ambiguity, ensuring clarity and consistency of the data.*
(2) *Annotation operation: The introduced annotation operation h distinguishes elements of $f(o_x, a_i^j)$ and $f(o_x, a_i^{j+1})$ in a manner that allows them to be differentiated. This operation aids in distinguishing relevant information during data processing and analysis.*
(3) *Symbol ($\star$): In the text, the symbol $\star$ denotes the specific implementation of the annotation operation, illustrating how to differentiate elements of adjacent scales.*

Next, the concept of reverse selection can be defined using simple coordination judgment.

Definition 3.5. Given a MsDT as a quadruple $(O, \{a_i^j, i = 1, 2, \dots, q, j = 1, 2, \dots, t\} \bigcup \{d\}, \{f(o, a) \mid \in O, a \in A\}, \{g(o, d) \mid o \in O\})$, for $X \in O/a_i^j$, $Y \in O/d$, if $X \subseteq Y$, this indicates that objects with a value of $f(o_x, a_i^j)$ $(o_x \in X)$ on a certain attribute a_i^j are consistent with the decision.

According to Definition 3.5, a depth search algorithm that reflects mappings through coordination relations can be defined. Initially, the original multiscale table can be transformed using Definition 3.4, such that each attribute in this table can generate a SbT. Subsequently, within each SbT, a depth search is conducted. Using coordination relations, an assessment is made to determine whether nodes from adjacent scales are coordinated, influencing the decision to proceed with further exploration. This constitutes the precise fusion discussed in this work. Its operational principles are depicted in Algorithm 2.

The time complexity of this algorithm can be analyzed as follows:

1. Outer loop (for loop): The outer loop iterates from $i = 1$ to q, where q is a parameter from the input. Each iteration constructs an SbT based on the data handling defined in Definition 3.4. Therefore, the time complexity of the outer loop is $O(q)$.
2. First inner loop (for loop): For each i, there is an inner loop that iterates from $j = 1$ to t, where t is the scale numbers of attribute a_i. Here, SbT_i is constructed, resulting in a time complexity of $O(t)$.
3. Second inner loop (while loop): For each SBT (SbT_i), a depth-first traversal is performed. Assuming the SbT has n nodes, the time complexity to traverse each node and process its children is $O(n)$.

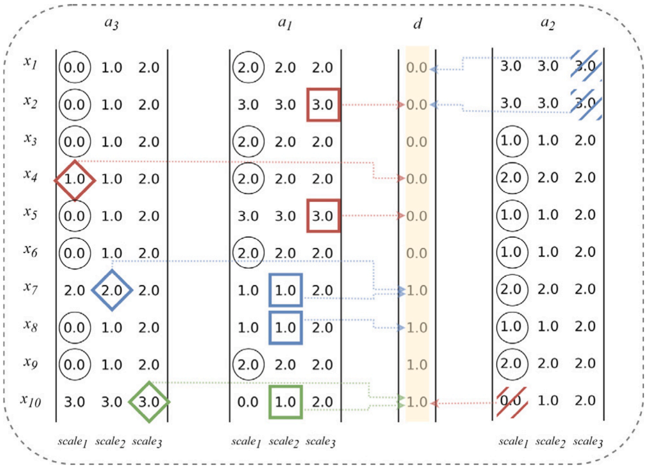
In summary, the overall time complexity of the algorithm is the product of the complexities of the outer and inner loops, and the traversal operations, $O(q \cdot t \cdot n)$. Here, q is the attribute numbers of data tables, t

Algorithm 2: Precise fusion algorithm.

```
Input  : $MsDT = (O, \{a_i^j, i = 1, 2, \ldots, q, j = 1, 2, \ldots, t\} \bigcup d, \{f(o, a) \mid o \in O, a \in A\}, \{g(o, d) \mid o \in O\})$.
Output : Tuple data or interval-value data.
begin
    for $i = 1 : q$ do
        // First, use Definition 3.4 to process the data with labels in preparation for generating the SbT.
        $tuplelist = NULL$
        for  $j = 1 : t$ do
            build $SbT_i$
        end
        // Perform a deep travel on the $SbT_i$.
        $queue = SbT_i.root$
        while queue exist do
            $node = queue.pop$
            $children = node.children$
            $queue.push(children)$
            for $k \in queue$ do
                if the sample corresponding to k is consistent with the attribute then
                    retrieve $k$ into tuplelist and remove $k$ from the queue
                end
            end
        end
        // Retrieve tuple data result.
        for $k \in tuplelist$ do
            retrieve the value corresponding to each object under this attribute
        end
        // Retrieve interval-valued data result.
        for k in tuplelist do
            retrieve the value from min to max in minist scale corresponding to each object under this attribute
        end
    end
end
return   : Tuple data result or interval-valued data result.
// The interval value is the maximum and minimum values corresponding to the tuple converted to the finest scale value.
```

is the scale numbers of each attribute, n is the total number of nodes in the SbT. Therefore, the algorithm's time complexity is $O(q \cdot t \cdot n)$, where the specific value of n depends on the structure of the SbT and how the data is processed.

Example 3.2. In Table 3, a tree structure is employed, utilizing depth search to find the most suitable fusion results for each attribute, starting from the coarsest scale. a_1 is initially selected: at the third scale, decision values are unique for instances with attribute value 3, thus halting the search. At the second scale, apart from instances where the search stopped, decision values for attribute value 1 are unique, also leading to the search stopping. Moving on to examine values at the first scale, decision values for attribute value 2 remain non-unique. Since the finest scale has been reached, values at this scale are chosen for all other instances. This completes the selection process for a_1. a_2 and a_3 follow the same process, with the selection process displayed in Fig. 4, and the results of the precise fusion are shown in Table 4.



	a_3			a_1			d	a_2		
	$scale_1$	$scale_2$	$scale_3$	$scale_1$	$scale_2$	$scale_3$		$scale_1$	$scale_2$	$scale_3$
x_1	0.0	1.0	2.0	2.0	2.0	2.0	0.0	3.0	3.0	3.0
x_2	0.0	1.0	2.0	3.0	3.0	3.0	0.0	3.0	3.0	3.0
x_3	0.0	1.0	2.0	2.0	2.0	2.0	0.0	1.0	1.0	2.0
x_4	1.0	1.0	2.0	2.0	2.0	2.0	0.0	2.0	2.0	2.0
x_5	0.0	1.0	2.0	3.0	3.0	3.0	0.0	1.0	1.0	2.0
x_6	0.0	1.0	2.0	2.0	2.0	2.0	0.0	1.0	1.0	2.0
x_7	2.0	2.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	2.0
x_8	0.0	1.0	2.0	1.0	1.0	2.0	1.0	1.0	1.0	2.0
x_9	0.0	1.0	2.0	2.0	2.0	2.0	1.0	2.0	2.0	2.0
x_{10}	3.0	3.0	3.0	0.0	1.0	2.0	1.0	0.0	1.0	2.0

Fig. 4. The selection chart.

Table 4
Results of Example 3.2.

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

4. Experimental analysis

To validate the efficiency of the proposed fusion method, comparative experiments were conducted on ten datasets obtained from the UCI database (https://archive.ics.uci.edu/ml/index.php). The detailed information on these datasets is provided in Table 5. The experimental procedures were carried out on a personal computer with specific hardware and software configurations, as described in Table 6. It is widely acknowledged that MsDT and Ms-MsDT cannot be directly obtained from any common databases. The simulation method can be borrowed from [41] to transform real datasets into Ms-MsDT and MsDT. In the comparison of multi-source fusion methods, using hierarchical clustering based on the number of decision attributes number, with each attribute set to have a maximum scale quantity of $scale_{max} = 2 * num(d)$ and a minimum scale quantity of $scale_{min} = \lfloor num(d)/2 \rfloor + 1$. Each attribute's range of cluster numbers is from $scale_{min}$ to $scale_{max}$, and the number of clusters for each attribute is set to $[scale_{min} + 1, scale_{max}]$. $num(d)$ can be used to define the number of sources. The multi-scale data for each attribute was randomly shuffled and unevenly distributed to different sources, Moreover, the initial attribute data was randomly inserted into one of the sources. With these two steps, the generation of Ms-MsDT was completed. For the comparison of precise fusion with optimal scale, we similarly used the method from [41] to generate multi-scale data and set the scale cluster number to one less than the number of distinct values of the attribute.

Due to the inability of existing methods to perform multi-source multi-scale fusion simultaneously, separate comparisons between multi-source fusion methods and multi-scale selection methods can be conducted and evaluated the effectiveness of our proposed approach.

Table 5
The detailed presentation of datasets.

No.	Datasets	Abbreviations	Objects	Attributes	Classes
1	Tae	Tae	151	5	3
2	Automobile	Automobile	159	18	4
3	Wine	Wine	178	13	3
4	Auto MPG	AM	398	7	3
5	Wine Quality-red	WQR	1599	11	6
6	Wine Quality-white	WQW	4898	11	7
7	Page Blocks Classification	PBC	5473	10	5
8	Shill Bidding Dataset	SBD	6321	9	2
9	Apartments for Rent Classified	ARC	10 000	6	3
10	HTRU2	HTRU2	17 898	8	2

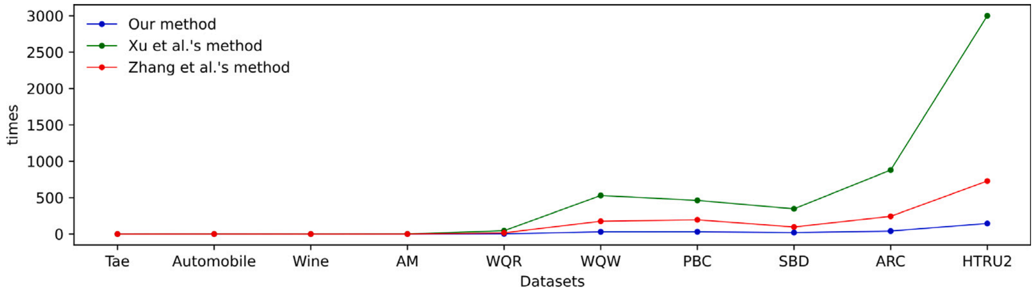


Fig. 5. Fusion time comparison chart.

Table 6
The running environment of experiments.

Name	Model	Parameter
CPU	Intel (R) Core (TM) i7 - 13700H	2.40 GHz
Platform	Python	3.10
System	Windows11	64 bit
Memory	DDR5	32 GB

4.1. The analysis of fusion method

To validate the effectiveness of our proposed fusion method in terms of time and accuracy, which is compared with two state-of-the-art fusion methods, method in [22] and method in [23].

(1) Zhang et al.'s method [22]: This new information fusion method is based on tolerance relations. It measures information sources and attributes by defining a new conditional entropy through tolerance relations and performs fusion on incomplete systems.

(2) Xu et al.'s method [23]: This information fusion method is based on fuzzy dominance relations. It measures information sources and attributes by defining a new fuzzy dominance conditional entropy and performs fusion on ordered information systems.

First, we evaluated the runtime of our method across ten datasets and compared it with Zhang et al.'s and Xu et al.'s methods. The results indicate that our method consistently achieved the fastest test runtime. Specifically, for the Tae, Automobile, Wine, and AM datasets, the runtime of our method was under 1 s. On the largest dataset, HTRU2, our method took 144.32 s. In contrast, Xu et al.'s method was the slowest, with a runtime exceeding 3000 s on the HTRU2 dataset. These results are detailed in Table 7 and Fig. 5.

Next, a detailed comparison of Approximation Precision (AP) and Approximation Quality (AQ) among the three fusion methods was conducted (see Definition 2.2 for specifics). AP and AQ are key indicators in rough set theory used to evaluate the importance of data, and they are crucial for assessing the accuracy and quality of data fusion results. In this study, the distance relation $R_\alpha = (x, y) \in O \times O \mid \frac{dis(x,y)}{\max_{z\in O} dis(x,z)} \leqslant \alpha$, defined by Xu et al. [23], was used as a baseline, where $dis(x, y)$ denotes the distance between x and y. It systematically adjusted the alpha parameter α within the range of 0.05 to 0.5 with a step size of 0.05, and performed a total of ten independent measurements. Through these experiments, we observed that lower alpha values are generally associated with higher performance of our proposed fusion method in terms of AP and AQ. Specifically, lower alpha values help enhance our method's precision and overall quality, highlighting our approach's superiority under different parameter settings.

In the experiments with the Tae dataset, our method did not surpass those of Zhang et al. and Xu et al. Specifically, although our algorithm showed some effectiveness, its performance did not exceed that of the two existing methods, which might be due to differences in dataset characteristics and algorithm adaptability. For the AM dataset, our method showed lower precision and quality when alpha values were below 0.4, while Zhang et al.'s method performed better in these metrics. However, our method still outperformed Xu et al.'s method in this scenario, indicating competitive performance under certain alpha settings. As alpha values increased, our method's precision and quality approached Zhang et al.'s level, demonstrating good adaptability and stability across different alpha settings. In the remaining seven datasets, our method achieved comparable precision and quality compared to Zhang et al.'s method at lower alpha values and performed better to Xu et al.'s method. This suggests that our method effectively utilizes lower alpha values for data fusion to match or exceed existing methods. Although minor performance differences occasionally appeared with increasing alpha values, these were not significant, and overall trends are detailed in Tables 8, 9 and Figs. 6, 7. Overall, our method offers comparable fusion results to other methods while significantly improving computational efficiency, indicating it not only matches existing technology but also brings substantial advancements in computational performance.

4.2. The analysis of scale selection

In addition to comparing with state-of-the-art fusion-related methods, we also conducted classifier comparisons. Initially, processing the dataset following the method in [41] transforming the original dataset into a multiscale decision information table, the scale details can be found in Table 10. our method is named OSS (processed using Algorithm 2),three derived datasets can be compared.

Table 7
Fusion time comparison.

Methods	Tae	Automobile	Wine	AM	WQR	WQW	PBC	SBD	ARC	HTRU2
Our method	**0.09**	**0.20**	**0.09**	**0.22**	**3.10**	**30.26**	**30.81**	**19.90**	**40.29**	**144.35**
Xu et al.’s method	0.41	1.22	0.77	1.80	44.52	529.54	462.48	345.86	879.14	–
Zhang et al.’s method	0.09	0.43	0.17	0.36	15.645	175.41	195.63	95.98	243.34	728.67

The ‘–’ symbol indicates a runtime exceeding 3000 s.

Table 8
Comparison of AP for our method, Xu et al.’s method and Zhang et al.’s method.

Datasets	Methods	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
Tae	Our method	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	**0.05**	0.00
	Xu et al.’s method	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.01	0.00
	Zhang et al.’s method	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.01	0.01
Automobile	Our method	**1.00**	**1.00**	**1.00**	**1.00**	**0.85**	**0.85**	**0.83**	0.62	0.39	0.20
	Xu et al.’s method	1.00	1.00	1.00	1.00	0.82	0.81	0.69	0.50	0.33	0.18
	Zhang et al.’s method	1.00	1.00	1.00	1.00	0.85	0.85	0.71	0.71	0.48	0.26
Wine	Our method	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.98**
	Xu et al.’s method	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88
	Zhang et al.’s method	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98
AM	Our method	0.43	0.43	0.43	0.43	0.43	0.43	0.43	**0.21**	**0.21**	**0.21**
	Xu et al.’s method	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.18	0.18	0.18
	Zhang et al.’s method	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.21	0.21	0.21
WQR	Our method	**0.81**	**0.81**	**0.81**	**0.81**	**0.81**	**0.81**	0.38	0.38	0.09	0.09
	Xu et al.’s method	0.69	0.69	0.69	0.69	0.69	0.69	0.23	0.23	0.05	0.05
	Zhang et al.’s method	0.81	0.81	0.81	0.81	0.81	0.81	0.39	0.39	0.10	0.10
WQW	Our method	**0.83**	**0.83**	**0.83**	**0.83**	**0.83**	**0.83**	0.35	0.35	**0.07**	**0.07**
	Xu et al.’s method	0.76	0.76	0.76	0.76	0.76	0.76	0.29	0.29	0.05	0.05
	Zhang et al.’s method	0.83	0.83	0.83	0.83	0.83	0.83	0.36	0.36	0.07	0.07
PBC	Our method	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.13**	**0.06**	**0.06**	**0.03**
	Xu et al.’s method	0.09	0.09	0.09	0.09	0.09	0.09	0.08	0.05	0.05	0.01
	Zhang et al.’s method	0.15	0.15	0.15	0.15	0.15	0.15	0.13	0.06	0.06	0.03
SBD	Our method	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.70**	**0.70**	**0.70**	**0.22**
	Xu et al.’s method	0.96	0.96	0.96	0.96	0.96	0.96	0.55	0.48	0.48	0.04
	Zhang et al.’s method	0.99	0.99	0.99	0.99	0.99	0.99	0.70	0.70	0.70	0.22
ARC	Our method	**0.08**	**0.08**	**0.08**	**0.08**	**0.08**	**0.08**	**0.08**	**0.08**	**0.01**	**0.01**
	Xu et al.’s method	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Zhang et al.’s method	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.01	0.01

Table 9
Comparison of AQ for our method, Xu et al.’s method and Zhang et al.’s method.

Datasets	Methods	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
Tae	Our method	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	**0.12**	0.01
	Xu et al.’s method	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.04	0.01
	Zhang et al.’s method	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.04	0.04
Automobile	Our method	**1.00**	**1.00**	**1.00**	**1.00**	**0.92**	**0.92**	**0.91**	0.77	0.58	0.36
	Xu et al.’s method	1.00	1.00	1.00	1.00	0.90	0.89	0.82	0.69	0.53	0.35
	Zhang et al.’s method	1.00	1.00	1.00	1.00	0.92	0.92	0.83	0.83	0.66	0.44
Wine	Our method	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**
	Xu et al.’s method	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94
	Zhang et al.’s method	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
AM	Our method	0.67	0.67	0.67	0.67	0.67	0.67	0.67	**0.43**	**0.43**	**0.43**
	Xu et al.’s method	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.39	0.39	0.39
	Zhang et al.’s method	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.43	0.43	0.43
WQR	Our method	**0.90**	**0.90**	**0.90**	**0.90**	**0.90**	**0.90**	0.59	0.59	0.23	0.23
	Xu et al.’s method	0.83	0.83	0.83	0.83	0.83	0.83	0.43	0.43	0.13	0.13
	Zhang et al.’s method	0.90	0.90	0.90	0.90	0.90	0.90	0.60	0.60	0.24	0.24
WQW	Our method	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	**0.91**	0.56	0.56	0.17	0.17
	Xu et al.’s method	0.87	0.87	0.87	0.87	0.87	0.87	0.50	0.50	0.13	0.13
	Zhang et al.’s method	0.91	0.91	0.91	0.91	0.91	0.91	0.57	0.57	0.18	0.18
PBC	Our method	**0.32**	**0.32**	**0.32**	**0.32**	**0.32**	**0.32**	**0.29**	**0.21**	**0.20**	**0.11**
	Xu et al.’s method	0.25	0.25	0.25	0.25	0.25	0.25	0.23	0.18	0.18	0.03
	Zhang et al.’s method	0.32	0.32	0.32	0.32	0.32	0.32	0.29	0.21	0.20	0.11
SBD	Our method	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.82**	**0.82**	**0.82**	**0.36**
	Xu et al.’s method	0.98	0.98	0.98	0.98	0.98	0.98	0.71	0.65	0.65	0.07
	Zhang et al.’s method	1.00	1.00	1.00	1.00	1.00	1.00	0.82	0.82	0.82	0.36
ARC	Our method	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.02**	**0.02**
	Xu et al.’s method	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00
	Zhang et al.’s method	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.02	0.02

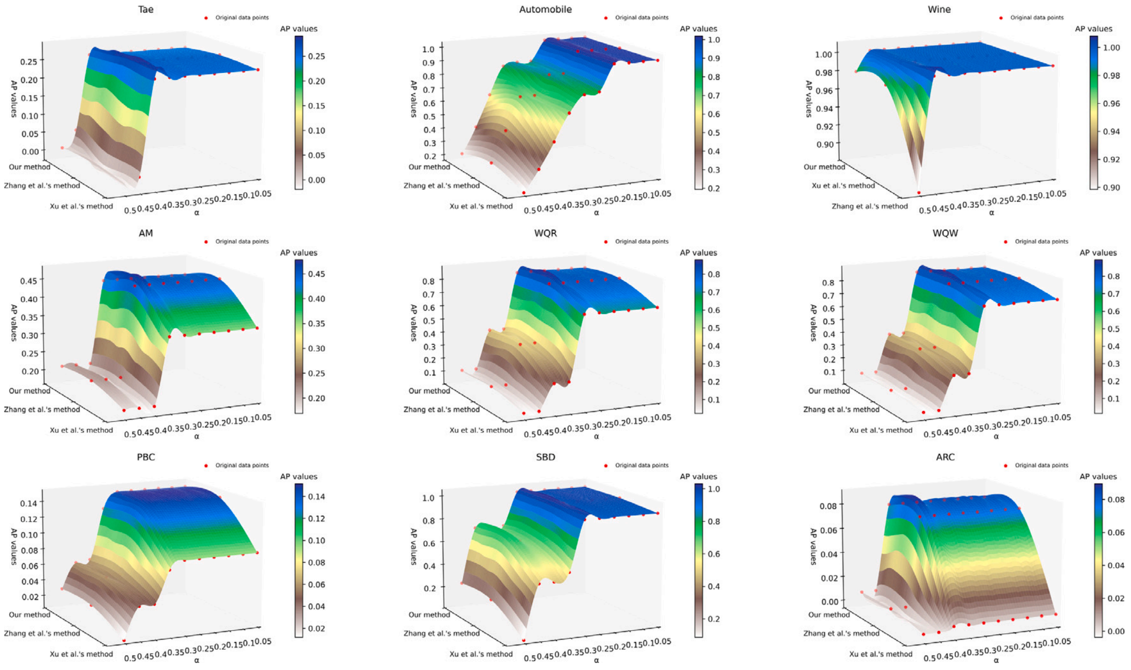


Fig. 6. Comparison of AP in nine datasets.

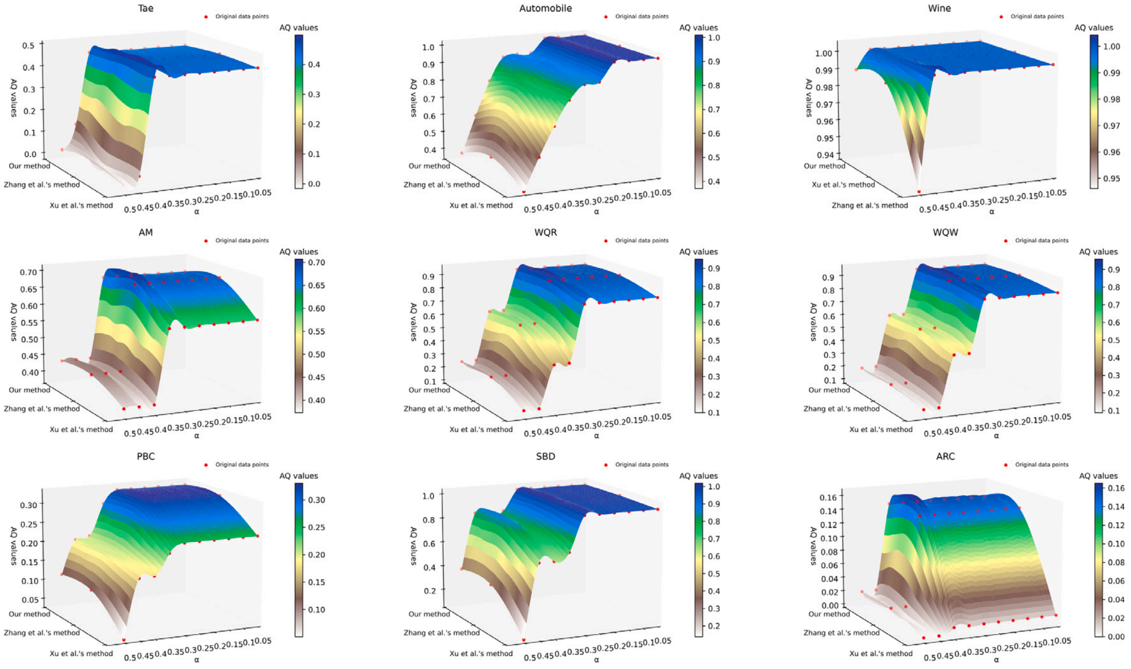


Fig. 7. Comparison of AQ in nine datasets.

(1) RAW (original dataset)
(2) CS (coarsest scale chosen from each attribute in the multiscale table)
(3) FS (finest scale chosen from each attribute in the multiscale table)

A ten-fold cross-validation method was employed to ensure the reliability of model evaluation, and the performance of three widely used classifiers, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP), was systematically compared. For the KNN classifier experiments, Euclidean distance was used as the metric, and the value of k was adjusted according to the characteristics of each dataset. The results indicated that the classification accuracy achieved with the OSS method was the same as that with the FS method across all datasets, demonstrating that our method can maintain classification performance comparable to the finest scale (FS) even with slightly lower data precision. It is noteworthy that on the ARC dataset, the accuracy of the OSS method was lower than that of the RAW dataset, suggesting a potential correlation between data processing methods and classification results. Additionally, on the other eight datasets, the OSS method consistently outperformed both the RAW and CS datasets, further validating its effectiveness. In the evaluation of the SVM classifier, the OSS method demonstrated comparable classification accuracy to the FS method, with both outperforming the RAW and CS datasets. This result indicates that our OSS method exhibits excellent

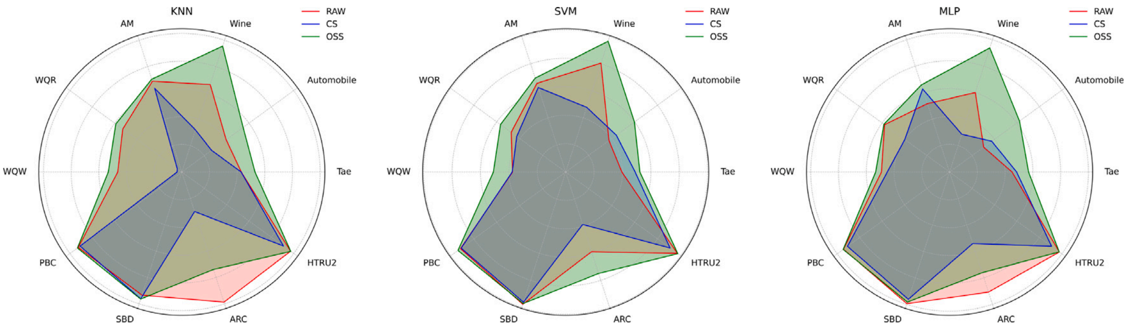


Fig. 8. The fusion comparison for RAW, CS, and OSS on three different classifiers.

Table 10
The number of scales set for different datasets.

Dataset	Scales
Tae	1 × 3 × 3 × 1 × 4
Automobile	5 × 1 × 1 × 5 × 5 × 5 × 4 × 5 × 4 × 6 × 5 × 7 × 2 × 4 × 4 × 5 × 5 × 5
Wine	4 × 4 × 6 × 5 × 6 × 4 × 4 × 4 × 5 × 5 × 5 × 3 × 4
AM	4 × 2 × 3 × 4 × 4 × 5 × 3
WQR	6 × 7 × 5 × 10 × 10 × 6 × 5 × 7 × 8 × 8 × 6
WQW	10 × 10 × 10 × 7 × 14 × 9 × 9 × 7 × 7 × 7 × 5
PBC	16 × 4 × 13 × 12 × 5 × 5 × 6 × 18 × 15 × 13
SBD	5 × 7 × 2 × 2 × 3 × 2 × 2 × 2 × 3
ARC	12 × 9 × 2 × 2 × 16 × 12
HTRU2	7 × 10 × 9 × 11 × 7 × 5 × 8 × 11

performance with the SVM classifier. In the MLP classifier experiments, we configured each hidden layer with 30 neurons and performed up to 500 iterations to ensure the model could fully adapt to all datasets. The experimental results showed that on the ARC dataset, the accuracy of the OSS method was slightly lower than that of the RAW dataset but comparable to the FS method, while on other datasets, the OSS method outperformed both RAW and CS datasets. Overall, among the three classifiers, the OSS method showed the most outstanding performance on the Wine dataset, demonstrating its significant advantage in classification performance.

The FS method consistently selects the finest scale of data, which provides the highest information quality and densest features, resulting in superior performance in classification tasks. However, the OSS method simplifies data storage by extracting from the raw data, making data processing more manageable. Although the extracted information from OSS is slightly less than that from FS, it is still superior to that from the CS method. This indicates that the OSS method achieves a good balance between information density and classification accuracy. Experimental results further validate the effectiveness of the OSS method. Despite the OSS method extracting less information than FS, it can still achieve the same classification accuracy in practical applications. This result shows that the OSS method, during scale selection optimization, effectively retains key information and maintains high classification performance. Therefore, the OSS method provides a more streamlined data processing solution compared to FS, demonstrating significant effectiveness and potential in practical applications. Detailed comparison results of various classifiers can be found in Tables 11–13, and Fig. 8 for a deeper understanding of the performance differences between methods.

In addition, we conducted hypothesis testing. For the OSS and FS methods, χ^2-tests are performed across three different classifiers: KNN, SVM, and MLP, to evaluate the effects of these methods. The χ^2-statistic represents the measure of deviation between observed and expected data within each classifier. The P-value indicates the results of the χ^2-tests for each classifier, measuring the significance of the statistical hypothesis test. A P-value greater than the significance level (typically 0.05) indicates failure to reject the null hypothesis. Here, all classifiers have P-values of 1.0, suggesting that neither classifier significantly affects the distribution of the data due to the feature selection methods. The degrees of freedom (F) column displays 8 for all classifiers, representing the degrees of freedom used in the χ^2 distribution to calculate the P-values. Furthermore, T-tests are conducted to comparing the OSS method with raw data (RAW) across the three different classifiers (KNN, SVM, MLP). The T-statistic measures the statistical difference in means between OSS and RAW data for each classifier. The P-value is an indicator of the hypothesis test's results; a P-value less than the significance level (typically 0.05) suggests rejecting the null hypothesis. For example, the SVM classifier has a P-value of 3.64×10^{-3} which is much less than 0.05, indicating a significant difference in means between OSS and RAW data when using the SVM classifier. In summary, our integrated classification accuracy shows a significant difference compared to the original classification accuracy. These results will be displayed in Tables 14, 15 .

5. Conclusions

Research on multi-source and multi-scale fusion is of great academic significance. Multi-source fusion involves gathering essential data from

Table 11
Comparison of different scale selection methods under the KNN classifier.

Method	Tae	Automobile	Wine	AM	WQR	WQW	PBC	SBD	ARC	HTRU2
RAW	43.04 ± 1.88	39.62 ± 1.09	66.41 ± 1.0	68.87 ± 0.58	52.78 ± 0.18	46.35 ± 0.04	92.55 ± 0.01	93.39 ± 0.01	98.69 ± 0.0	97.21 ± 0.0
CS	42.92 ± 1.09	26.62 ± 2.18	31.96 ± 1.37	63.49 ± 0.43	4.07 ± 0.03	3.41 ± 0.01	90.92 ± 0.01	96.17 ± 0.01	29.99 ± 0.01	90.84 ± 0.01
FS	52.92 ± 1.42	54.79 ± 1.65	95.49 ± 0.17	70.4 ± 0.2	58.98 ± 0.09	53.16 ± 0.03	93.28 ± 0.01	96.36 ± 0.01	73.96 ± 0.02	97.36 ± 0.0
OSS	**52.92 ± 1.42**	**54.79 ± 1.65**	**95.49 ± 0.17**	**70.4 ± 0.2**	**58.98 ± 0.09**	**53.16 ± 0.03**	**93.28 ± 0.01**	**96.36 ± 0.01**	73.96 ± 0.02	**97.36 ± 0.0**

Table 12
Comparison of different scale selection methods under the SVM classifier.

Method	Tae	Automobile	Wine	AM	WQR	WQW	PBC	SBD	ARC	HTRU2
RAW	39.58 ± 2.19	37.67 ± 1.68	80.46 ± 0.98	65.53 ± 0.48	47.42 ± 1.13	37.44 ± 0.63	91.58 ± 0.44	97.74 ± 0.0	58.9 ± 0.85	97.16 ± 0.02
CS	48.92 ± 1.58	44.08 ± 0.75	47.81 ± 0.87	62.47 ± 0.3	42.59 ± 0.12	37.67 ± 1.72	90.97 ± 0.01	96.2 ± 0.01	38.75 ± 1.16	90.86 ± 0.01
FS	52.25 ± 1.72	59.83 ± 1.14	96.63 ± 0.26	69.4 ± 0.22	57.04 ± 0.09	50.88 ± 0.05	93.77 ± 0.01	97.17 ± 0.01	75.05 ± 0.05	97.56 ± 0.0
OSS	**52.25 ± 1.72**	**59.83 ± 1.14**	**96.63 ± 0.26**	**69.4 ± 0.22**	**56.73 ± 0.1**	**50.92 ± 0.05**	93.75 ± 0.01	97.17 ± 0.01	74.98 ± 0.05	**97.56 ± 0.0**

Table 13
Comparison of different scale selection methods under the MLP classifier.

Method	Tae	Automobile	Wine	AM	WQR	WQW	PBC	SBD	ARC	HTRU2
RAW	45.04 ± 1.14	30.38 ± 2.15	60.07 ± 1.79	51.7 ± 4.16	57.85 ± 0.24	49.06 ± 0.11	94.39 ± 0.01	99.51 ± 0.0	90.76 ± 0.21	97.77 ± 0.0
CS	48.33 ± 1.32	37.58 ± 1.21	28.56 ± 1.18	62.72 ± 0.41	39.84 ± 0.14	44.86 ± 0.07	90.9 ± 0.01	96.17 ± 0.01	54.08 ± 0.02	90.84 ± 0.01
FS	56.92 ± 1.11	62.29 ± 0.76	93.76 ± 0.23	65.82 ± 0.31	58.41 ± 0.16	53.21 ± 0.05	94.67 ± 0.01	98.31 ± 0.0	76.11 ± 0.03	97.62 ± 0.0
OSS	**56.92 ± 1.11**	**62.29 ± 0.76**	**93.76 ± 0.23**	**65.82 ± 0.31**	**58.41 ± 0.16**	**53.21 ± 0.05**	**94.67 ± 0.01**	98.31 ± 0.0	76.11 ± 0.03	97.62 ± 0.0

Table 14
Perform a χ^2-test for OSS and FS.

Classifier	χ^2 statistic	P-value	F
KNN	4.05×10^{-29}	1.0	8
SVM	8.08×10^{-4}	1.0	8
MLP	4.00×10^{-29}	1.0	8

Table 15
Perform a T-test for OSS and RAW.

Classifier	T-statistic	P-value
KNN	−1.11	0.29
SVM	−3.89	3.64×10^{-3}
MLP	−1.67	0.12

multiple sensors, while multi-scale methods allow for selective processing of data at various levels of granularity, catering to specific needs. This paper proposes a supervised fusion framework based on Ms-MsDT, utilizing the information gain function from decision trees. Firstly, our proposed information gain function helps swiftly identify critical scales for data filtering. Subsequently, employing depth-first search with scale trees enables efficient reverse selection of data. Furthermore, we conduct comparative experiments across ten datasets against various state-of-the-art fusion methods and scale selection techniques. The effectiveness and efficiency of our proposed multi-source and multi-scale fusion approach are well-validated. Although we have integrated tree structures with multi-scale tables, the proposed fusion framework has several limitations. First, its current processing capabilities are primarily limited to categorical and numerical data, and it has not yet been extended to handle other data types. Second, although the fusion process demonstrates relative advantages in terms of time efficiency, the tree-based structure used may lead to significant space consumption when dealing with large-scale datasets. Additionally, the framework has not yet incorporated state-of-the-art deep learning techniques in the reverse data selection process, missing the opportunity to leverage these methods for enhancing fusion accuracy. Future work should explore cutting-edge areas like granular computing and deep learning, integrate with adaptive boosting in ensemble learning, and delve deeper into the fusion mechanisms of multi-source and multi-scale tables.

CRediT authorship contribution statement

Xiaoyan Zhang: Writing – review & editing, Validation, Methodology, Investigation, Funding acquisition, Conceptualization. **Jiajia Lin:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, 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.

Data availability

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

Acknowledgments

The authors would like to thank the Associate Editor and the reviewers for their insightful comments and suggestions.

This work was supported by the National Natural Science Foundation of China (Grant NO.12371465) and the Chongqing Natural Science Foundation (Grant NO. CSTB2023NSCO-MSX1063).

References

[1] Chongzhao Han, Hongyan Zhu, Zhansheng Duan, Multi-Source Information Fusion, Tsinghua University Press, Beijing, 2010.
[2] Quan Pan, Multi-Source Information Fusion Theory and Its Applications, Tsinghua University Press, Beijing, 2013.
[3] David L. Hall, Sonya A.H. McMullen, Mathematical Techniques in Multisensor Data Fusion (Artech House Information Warfare Library), Artech House, USA, 2004.
[4] Wei Wei, Jiye Liang, Information fusion in rough set theory : An overview, Inf. Fusion 48 (2019) 107–118.
[5] Pengfei Zhang, Tianrui Li, Guoqiang Wang, Chuan Luo, Hongmei Chen, Junbo Zhang, Dexian Wang, Zeng Yu, Multi-source information fusion based on rough set theory: A review, Inf. Fusion 68 (2021) 85–117.
[6] Doudou Guo, Weihua Xu, Weiping Ding, Yiyu Yao, Xizhao Wang, Witold Pedrycz, Yuhua Qian, Concept-cognitive learning survey: Mining and fusing knowledge from data, Inf. Fusion 109 (2024) 102426.
[7] Yuhua Qian, Jiye Liang, Chuangyin Dang, Knowledge structure, knowledge granulation and knowledge distance in a knowledge base, Internat. J. Approx. Reason. 50 (1) (2009) 174–188.
[8] Yuhua Qian, Jiye Liang, Chuangyin Dang, Incomplete multigranulation rough set, IEEE Trans. Syst. Man Cybern. A 40 (2) (2010) 420–431.
[9] Yuhua Qian, Jiye Liang, Yiyu Yao, Chuangyin Dang, MGRS: A multi-granulation rough set, Inform. Sci. 180 (6) (2010) 949–970.
[10] Yuhua Qian, Shunyong Li, Jiye Liang, Zhongzhi Shi, Feng Wang, Pessimistic rough set based decisions: A multigranulation fusion strategy, Inform. Sci. 264 (2014) 196–210.
[11] Yuhua Qian, Hu Zhang, Yanli Sang, Jiye Liang, Multigranulation decision-theoretic rough sets, Internat. J. Approx. Reason. 55 (1) (2014) 225–237.
[12] Xiaoyan Zhang, Xudong Huang, Weihua Xu, Matrix-based multi-granulation fusion approach for dynamic updating of knowledge in multi-source information, Knowl.-Based Syst. 262 (2023) 110257.
[13] Glenn Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.
[14] Arthur P. Dempster, Upper and lower probabilities induced by a multivalued mapping, in: Classic Works of the Dempster-Shafer Theory of Belief Functions, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 57–72.
[15] Philippe Smets, Analyzing the combination of conflicting belief functions, Inf. Fusion 8 (4) (2007) 387–412.
[16] Yong Deng, Zhenfu Zhu, Shan Zhong, Fuzzy information fusion based on evidence theory and its application in target recognition, Acta Aeronaut. Astron. Sin. 26 (2005) 754–758.
[17] Guang Yang, Shuofeng Yu, Shan Lu, George Smith, The comprehensive diagnostic method combining rough sets and evidence theory, Appl. Math. Nonlinear Sci. 6 (2) (2021) 171–180.
[18] YuTing Liu, Nikhil R. Pal, Amar R. Marathe, ChinTeng Lin, Weighted fuzzy Dempster–Shafer framework for multimodal information integration, IEEE Trans. Fuzzy Syst. 26 (1) (2017) 338–352.

[19] Yong Chen, Yongchuan Tang, Yan Lei, An improved data fusion method based on weighted belief entropy considering the negation of basic probability assignment, J. Math. Univ. Tokushima 2020 (2020) 1594967.

[20] Hongfei Wang, Xinyang Deng, Wen Jiang, Jie Geng, A new belief divergence measure for Dempster–Shafer theory based on belief and plausibility function and its application in multi-source data fusion, Eng. Appl. Artif. Intell. 97 (2021) 104030.

[21] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.

[22] Xiaoyan Zhang, Xiuwei Chen, Weihua Xu, Weiping 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.

[23] Weihua Xu, Yanzhou Pan, Xiuwei Chen, Weiping Ding, Yuhua Qian, A novel dynamic fusion approach using information entropy for interval-valued ordered datasets, IEEE Trans. Big Data 9 (3) (2023) 845–859.

[24] Ke Cai, Weihua Xu, An efficient multi-source information fusion approach for dynamic interval-valued data via fuzzy approximate conditional entropy, Int. J. Mach. Learn. Cybern. 15 (2024) 3619–3645.

[25] Xiuwei Chen, Maokang Luo, Incremental information fusion in the presence of object variations for incomplete interval-valued data based on information entropy, Inform. Sci. 667 (2024) 120479.

[26] Anhui Tan, Jinjin Li, Yaojin Lin, Guoping Lin, Matrix-based set approximations and reductions in covering decision information systems, Internat. J. Approx. Reason. 59 (2015) 68–80.

[27] Anhui Tan, Weizhi Wu, Yuzhi Tao, On the belief structures and reductions of multigranulation spaces with decisions, Internat. J. Approx. Reason. 88 (2017) 39–52.

[28] Anhui Tan, Weizhi Wu, Yuhua Qian, Jiye Liang, Jinkun Chen, Jinjin Li, Intuitionistic fuzzy rough set-based granular structures and attribute subset selection, IEEE Trans. Fuzzy Syst. 27 (3) (2019) 527–539.

[29] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353.

[30] Jerry R. Hobbs, Granularity, in: Readings in Qualitative Reasoning About Physical Systems, Morgan Kaufmann, 1990, pp. 542–545.

[31] YiYu Yao, Granular computing using neighborhood systems, in: Rajkumar Roy, Takeshi Furuhashi, Pravir K. Chawdhry (Eds.), Advances in Soft Computing, London, 1999, pp. 539–553.

[32] Wentao Li, Haoxiang Zhou, Weihua Xu, Xi-Zhao Wang, Witold Pedrycz, Interval dominance-based feature selection for interval-valued ordered data, IEEE Trans. Neural Netw. Learn. Syst. 34 (10) (2023) 6898–6912.

[33] Weihua Xu, Kehua Yuan, Wentao Li, Weiping Ding, An emerging fuzzy feature selection method using composite entropy-based uncertainty measure and data distribution, IEEE Trans. Emerg. Top. Comput. Intell. 7 (1) (2023) 76–88.

[34] Wei-Zhi Wu, Yee Leung, Optimal scale selection for multi-scale decision tables, Internat. J. Approx. Reason. 54 (8) (2013) 1107–1129.

[35] Chuan Luo, Tianrui Li, Yanyong Huang, Hamido Fujita, Updating three-way decisions in incomplete multi-scale information systems, Inform. Sci. 476 (2019) 274–289.

[36] Weihua Xu, Wentao Li, Granular computing approach to two-way learning based on formal concept analysis in fuzzy datasets, IEEE Trans. Cybern. 46 (2) (2016) 366–379.

[37] Feng Li, Baoqing Hu, Jun Wang, Stepwise optimal scale selection for multi-scale decision tables via attribute significance, Knowl.-Based Syst. 129 (2017) 4–16.

[38] Chen Hao, Jinhai Li, Min Fan, Wenqi Liu, Eric C.C. Tsang, Optimal scale selection in dynamic multi-scale decision tables based on sequential three-way decisions, Inform. Sci. 415–416 (2017) 213–232.

[39] Yanhong She, Zhuojun Zhao, Mengting Hu, Wenli Zheng, Xiaoli He, On selection of optimal cuts in complete multi-scale decision tables, Artif. Intell. Rev. 54 (8) (2021) 6125–6148.

[40] Han Bao, Wei-Zhi Wu, Jia-Wen Zheng, Tong-Jun Li, Entropy based optimal scale combination selection for generalized multi-scale information tables, Int. J. Mach. Learn. Cybern. 12 (5) (2021) 1427–1437.

[41] Xiaoyan Zhang, Yuyang Huang, Optimal scale selection and knowledge discovery in generalized multi-scale decision tables, Internat. J. Approx. Reason. 161 (2023) 108983.

[42] Sheng-yi Jiang, Lian-xi Wang, Efficient feature selection based on correlation measure between continuous and discrete features, Inform. Process. Lett. 116 (2) (2016) 203–215.

[43] Tran Binh, Bing Xue, Mengjie Zhang, A new representation in PSO for discretization-based feature selection, IEEE Trans. Cybern. 48 (6) (2018) 1733–1746.

[44] Qinghua Hu, Daren Yu, Jinfu Liu, Congxin Wu, Neighborhood rough set based heterogeneous feature subset selection, Inform. Sci. 178 (18) (2008) 3577–3594.

[45] Xiao Zhang, Changlin Mei, Degang Chen, Jinhai Li, Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy, Pattern Recognit. 56 (2016) 1–15.

[46] Pengfei Zhang, Tianrui Li, Zhong Yuan, Chuan Luo, Keyu Liu, Xiaoling Yang, Heterogeneous feature selection based on neighborhood combination entropy, IEEE Trans. Neural Netw. Learn. Syst. 35 (3) (2024) 3514–3527.

[47] Jianhua Dai, Wenxiang Chen, Liyun Xia, Feature selection based on neighborhood complementary entropy for heterogeneous data, Inform. Sci. 682 (2024) 121261.

[48] Johannes Fürnkranz, Decision tree, in: Encyclopedia of Machine Learning and Data Mining, Springer US, Boston, MA, 2016, pp. 1–5.