Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

Research Article

A Granular-Ball Based Online Streaming Feature Selection Algorithm

Jiangdi Yang[1], Shaokun Jia[2], Chenglin Zhang[2], Jie Yang[2*], Fan Zhao[2]

[1] School of Computer Science and Technology, Guizhou University, Guiyang, China
[2] Key Laboratory of Big Data Intelligent Computing, Chongqing University of Posts and Telecommunications, Chongqing, China
E-mail: 530966074@qq.com

Received: 2 July 2025; **Revised:** 22 August 2025; **Accepted:** 29 August 2025

Abstract: Online Streaming Feature Selection (OSFS) is a feature selection method to identify relevant features in real-time from high-dimensional, continuously generated data streams. Traditional methods require the pre-setting of static parameters and exhibit insufficient robustness. Therefore, these methods may fail to adapt effectively to evolving feature sets and are prone to noise interference. To overcome these limitations, this paper proposes an online streaming feature selection algorithm based on Granular Ball-Online Streaming Feature Selection (GB-OSFS). GB-OSFS adopts a granular ball approach in OSFS, shifting the focus from individual sample points to granular balls for calculating feature dependencies. Furthermore, we select granular balls with a purity of 1 as the evaluation metric for feature importance. As a result, this approach addresses the issues of traditional streaming feature selection methods being prone to noise interference and the necessity of pre-setting various hyperparameters. To validate the efficacy of GB-OSFS, a series of comprehensive empirical tests were conducted across ten datasets. The findings indicate that GB-OSFS surpasses six other OSFS algorithms in terms of prediction accuracy and robustness.

***Keywords*:** Online Streaming Feature Selection (OSFS), feature selection, robustness, granular ball

MSC: 68W27, 68T09, 03E72

1. Introduction

Feature selection [1–3] is a fundamental problem in the field of machine learning. In the era of big data, the explosion of information from sources like the Internet of Things (IoT), social media, and financial transactions has led to the continuous generation of massive, high-dimensional streaming data. This data is not only large in volume but often characterized by its non-stationary and heterogeneous nature, posing significant challenges for real-time analysis and decision-making [4]. Consequently, feature selection algorithms based on streaming data have become a research hotspot. These algorithms, known as Online Streaming Feature Selection (OSFS), aim to dynamically select relevant features as they arrive, in real-time, without prior knowledge of the entire feature space.

An effective OSFS algorithm must adapt to various datasets. However, many OSFS algorithms require presetting different hyperparameters to achieve optimal performance when handling different datasets. Appropriate hyperparameters can enhance the effectiveness of feature selection, while inappropriate ones can adversely affect the results. For example, in α-investing [5], two parameters that significantly influence the results require to be set before performing streaming

DOI: https://doi.org/10.37256/cm.6620257675

feature selection. Datasets vary widely in practice, making it evident that setting different parameters for each dataset is difficult and time-consuming. Moreover, existing OSFS algorithms demonstrate limited robustness due to their reliance on individual data points for feature relevance computation. This point-wise estimation approach makes them highly sensitive to noise variations, where even minor disturbances can substantially compromise their selection performance.

To address the above issues, we drew inspiration from Neighborhood Rough Sets (NRS) and granular balls. NRS is an advancement developed from Probabilistic Rough Sets (PRS). The feature selection algorithm based on rough set theory relies on attribute reduction [6–8]. Lin [9] conducted in-depth research on rough sets and neighborhood systems, exploring the relationships between neighborhoods, rough sets, and fuzzy sets. Yao [10] also analyzed the granular structure by combining rough sets and neighborhood systems. Subsequently, Hu et al. [11] and colleagues expanded upon Lin's neighborhood framework [12], introducing the NRS approach. This method employs a neighborhood-based relationship rather than an equivalence-based one to partition datasets, allowing NRS to handle continuous data without preprocessing. Nonetheless, within the NRS framework, the choice of neighborhoods and the impact of noise interference continue to be a challenge. Subsequently, Xia [13, 14] proposed the concept of Granular Ball Computing (GBC), which aggregates multiple data samples into a single granular ball. By treating the granular ball as the research subject, this approach effectively addresses the issue of noise interference that arises when individual data samples are used as the research subject. Currently, GBC has witnessed continual progress in both its methodologies and practical applications. Recently, Cheng et al. [15] developed a manifold clustering algorithm based on GBC for handling ultra-scalable data, which generates high-quality anchors that closely align with the underlying data distribution. To better handle clusters of varying shapes, Xie et al. [16] proposed an adaptive density clustering method that incorporates multi-granularity granular-ball fusion. Pan et al. [17] introduced an adaptive radius determination mechanism that dynamically adjusts based on granularity, and further explored the application of the resulting granular balls in classification tasks. Sajid et al. [18] proposed the GE-GB-RVFL model, a hybrid framework that integrates granular computing with graph embedding techniques. Additionally, Gao et al. [19] presented a multi-scale outlier detection method based on fuzzy rough sets, aimed at effectively identifying diverse types of outliers.

Building upon these insights, this paper introduces granular ball computing into OSFS, designing a novel OSFS method called Granular Ball-Online Streaming Feature Selection (GB-OSFS). The novelty and significance of our work are multi-fold:

• GB-OSFS represents data using granular balls, significantly reducing the data size and improving selection performance. At the same time, since a granular ball contains multiple data points, it exhibits excellent noise resistance.

• Furthermore, since GB-OSFS sets the granular ball purity to 1, the sample labels within a granule ball of purity 1 are consistent, thereby enhancing classification performance. Moreover, there is no need to set different purity parameters for each dataset, effectively eliminating the hyperparameter dependency that plagues many existing methods (e.g., α-investing), and it demonstrates strong performance across various datasets.

The novelty and significance of our work are multi-fold. Unlike traditional OSFS methods that operate on individual data points and are susceptible to noise and parameter sensitivity, our proposed GB-OSFS algorithm introduces a fundamental paradigm shift by leveraging GBC as its core computational unit. This is, to the best of our knowledge, the first effort to integrate GBC into the OSFS framework. This integration offers two paramount advantages: 1) Enhanced Robustness: By evaluating feature relevance based on granular balls rather than single points, our method inherently dampens the adverse effects of noise and outliers. 2) Reduced Hyperparameter Dependency: The use of stable, high-purity granular balls eliminates the need for tedious pre-configuration of multiple problem-specific thresholds, a common limitation in existing algorithms like α-investing. Consequently, GB-OSFS presents a more generic, robust, and efficient solution for feature selection in dynamic streaming environments, paving the way for more reliable real-time analytics.

The rest of the paper is structured as follows: Section 2 discusses related work. Section 3 provides a detailed introduction to the OSFS method with the introduction of granular ball computing. Section 4 conducts relevant experiments and analyzes the experimental results. Finally, Section 5 concludes the paper.

2. Related work

In this section, we introduce relevant background knowledge, including feature selection methods, OSFS algorithms, granular ball computing, and pertinent definitions. The adopted symbols of this article are summarizd and explained in Table 1.

Table 1. Symbol annotations

Symbol	Annotations
U	The set of objects to be discussed within a range is called the domain.
B	B is relation of equivalence on U.
C	Conditional feature set.
D	Decision feature set.
h	Mapping function.
Q	Granular ball Center.
R_G	Granular ball radius.
S	Selected feature set.
F	A feature set on U.
R	Feature set relevance.
P	Granular ball purity.
$Mdeps$	Granular ball coverage with feature purity.
H	Remove one or more items from S.
$ball.num$	Number of samples covered by a specific granular ball.

2.1 *Three methods of feature selection*

Feature selection methods can be divided into three categories: filter methods, wrapper methods, and embedded methods [20, 21].

Filter methods score features based on evaluation criteria and then rank them in descending order according to their assigned scores. It selects features independently by evaluating the correlation or statistical properties between the features and the target variable, without relying on a specific machine learning model. Filter methods are typically performed before model training and are characterized by high computational efficiency and strong generalizability [22].

Wrapper methods [23, 24] are feature selection technique that evaluates the quality of feature subsets by utilizing the performance of machine learning models, using classifiers to evaluate these subsets. Since a classifier needs to be trained for each feature subset, many of them perform very poorly. Consequently, previous research on these methods has often focused on optimizing this step.

Embedded methods [25] during the model training process, it automatically selects the features that contribute the most to the model through regularization or specific algorithms. These methods inherently assume static feature-label relationships, rendering them incapable of handling dynamically evolving features. Furthermore, their batch-processing paradigm fails to meet the real-time decision-making requirements of contemporary data streams. Consequently, the research on OSFS becomes indispensable.

2.2 *OSFS methods*

The advantage of OSFS over traditional feature selection methods is that the former can handle high-dimensional data, while traditional methods require significant time costs to process high-dimensional data, which is often impractical in real-world scenarios. Therefore, OSFS has become a focus in recent years [26, 27].

OSFS algorithms not only consider the relationship between features and labels, but also focus on the relationships among features. Consequently, this method eliminates redundant features by leveraging inter-feature relationships, thereby enhancing prediction accuracy.

In recent years, researchers have been striving to optimize OSFS. For example, Zhou et al. proposed the α-investing method, which requires pre-set hyperparameters [5]. The effectiveness of the algorithm largely depend on the chosen parameter settings.

Although the training speed of α-investing is decent, its low prediction accuracy is a significant issue. Therefore, Wu et al. proposed two OSFS algorithms, OSFS and Fast OSFS [1], which effectively improve prediction accuracy. The algorithm consists of two key steps:

1) Relevance analysis: exclude irrelevant features.

2) Redundancy analysis: eliminates redundant features.

OSFS selects features based on conditional uncertainty, which requires a large number of training instances. When the dataset is small, OSFS may produce unreliable results.

Although OSFS and Fast-OSFS have better prediction accuracy compared to other OSFS methods, it exhibits high parameter sensitivity and is susceptible to noise interference. Yu et al. presented a Scalable and Accurate OnLine Approach for feature selection (SAOLA) [4], which is primarily used to address feature selection problems in large-scale data streams. The core idea of the algorithm is to adaptively select the features most relevant to the task while suppressing the impact of redundant features, thereby improving the model's prediction accuracy. SAOLA is a feature selection method for online learning, but it faces challenges when dealing with feature redundancy, which may lead to a larger feature set and lower prediction accuracy.

Lou et al. presented a Rough Hypercuboid based Distributed Online Feature Selection (RHDOFS) method [28], witch is an optimization filtering method used for high-dimensional data processing, aimed at efficient estimation and prediction in high-dimensional and complex data environments. It combines regularization techniques and optimal filtering ideas, and is widely applied in fields such as signal processing, data fusion, and weather forecasting. Although RHDOFS effectively mitigates issues like the curse of dimensionality through regularization techniques when dealing with high-dimensional systems, it still has limitations, including high computational complexity, sensitivity to regularization parameters, reliance on prior knowledge, and strict assumptions regarding the noise model. OSFS-Vague [29] is a vague set [30] learning algorithm primarily used to handle data that contains uncertainty and fuzziness. It is developed based on the traditional idea of optimal subspaces. The OSFS-Vague algorithm combines feature selection with fuzzy logic processing methods, effectively handling the fuzziness and imprecision present in the data, thus improving the accuracy of classification or regression tasks. However, by using fuzzy sets to process the uncertainty in the data, while this approach enhances the algorithm's flexibility, it may also interpret noise as fuzzy information when the data contains noise, which could affect the accuracy of feature selection. Especially when the data quality is poor, OSFS-Vague may retain some irrelevant noisy features, leading to a decline in model performance.

2.3 *The definition of the OSFS problem*

Assuming OSFS = (C, D, h, t) represents an OSFS framework, where C is the set of conditional features and D is the set of decision features. $C = [x_1, x_2, \ldots, x_n]^T \in R^{n\times d}$ consists of n samples from d-dimensional feature space $F = [f_1, f_2, \ldots, f_d]^T \in R^d$. $D = [y_1, y_2, \ldots, y_n]^T \in R^{n\times L}$ consists of n samples from class label space (decision feature space) $L = \{l_1, l_2, \ldots, l_m\}$, where l_i represents the value of the class label. Given C and D, at time point t, we obtain the feature f_i from $C \cup D$ and do not know the specific size of the feature space d in advance. The problem of OSFS is to select a subset from the features that have already arrived, such that the mapping function h: $x_i \to L$ $(x_i \in C)$ at time point t performs sufficiently well.

2.4 *Granular ball computing*

Building on the theoretical framework of traditional granular ball computing and Chen's insights in Science [31] on the large-scale priority of human cognition, Wang [32] proposed several approaches to granular cognitive computing.

Building upon this foundation, Xia et al. [13] introduced an innovative, efficient, and resilient granular ball computing technique. The central concept of this method involves utilizing "granular balls" to comprehensively or partially encompass the sample space. A granular ball is represented by Q and R, and their definitions are as follows:

$$Q = \frac{1}{v}\sum_{i=1}^{v} x_i, \tag{1}$$

$$R = \max_{i=1}^{v} |x_i - Q|, \tag{2}$$

where x_i delegates the samples in GB and v is the number of objects in GB. Q and R represent the center and radius of the granular ball, respectively. This approach enables multigranularity learning features, such as scalability and the ability to handle multiple scales, and provides an accurate depiction of the sample space.

Given a dataset $U_1 = \{x_i,\ i = 1,\ \ldots,\ m\}$, where x_i and m degelate the samples and the number of samples in U_1, and $GB_j(j = 1,\ 2,\ ...,\ b)$ is a granular ball generated on U_1 and b represents the number of granular balls created on U_1. The generation of these granular balls in granular ball computing is mainly assessed based on the following criteria.

1) Regarding the coverage degree, when all other factors remain consistent, a higher degree of coverage indicates less loss of sample information and a more accurate representation. If we represent the quantity of samples in the jth granular ball as $|GB|$, the coverage rate can be represented as: $\left(\sum_{j=1}^{b}\left(\left|GB_j\right|\right)/m\right)$.

2) To achieve optimal outcomes across various problems, the quality of each granular ball GB_j must exceed a specified threshold . This condition is linked to the maximum roughness of the granular ball, guaranteeing that even the smallest granular ball maintains the necessary level of precision to effectively represent the problem.

The granular ball generation algorithm follows the human tendency to first grasp the general outline of things. Therefore, at the initial stage, the entire dataset is considered as a single large ball, and the quality of this ball is the lowest at this point, rendering it unable to capture any data distribution characteristics. "Purity" serves as a metric to assess the quality of a granular ball [13], representing the proportion of the most prevalent label within it. This setting of purity = 1 is highly effective for clean, well-separated datasets as it prioritizes purity above all else. This practice ironically introduces a rigid, pre-set hyperparameter that the paradigm claims to avoid. This fixed threshold becomes a significant limitation for noisy, complex, real-world datasets, where enforcing perfect purity can lead to overfitting, instability, or excessive granules. Subsequently, the number of unique label categories m within the granular ball is tallied, and the granular ball is divided into m sub-granular balls. The purity of each sub-granular ball is then computed, a pivotal step in determining whether further splitting is necessary. As the balls continue to split, their purity steadily improves, leading to a clearer decision boundary. Once they meet a specific condition, the splitting stops. The final granular balls are referred to as stable granular balls. As shown in Figure 1 below, it is a schematic diagram of granular ball splitting.

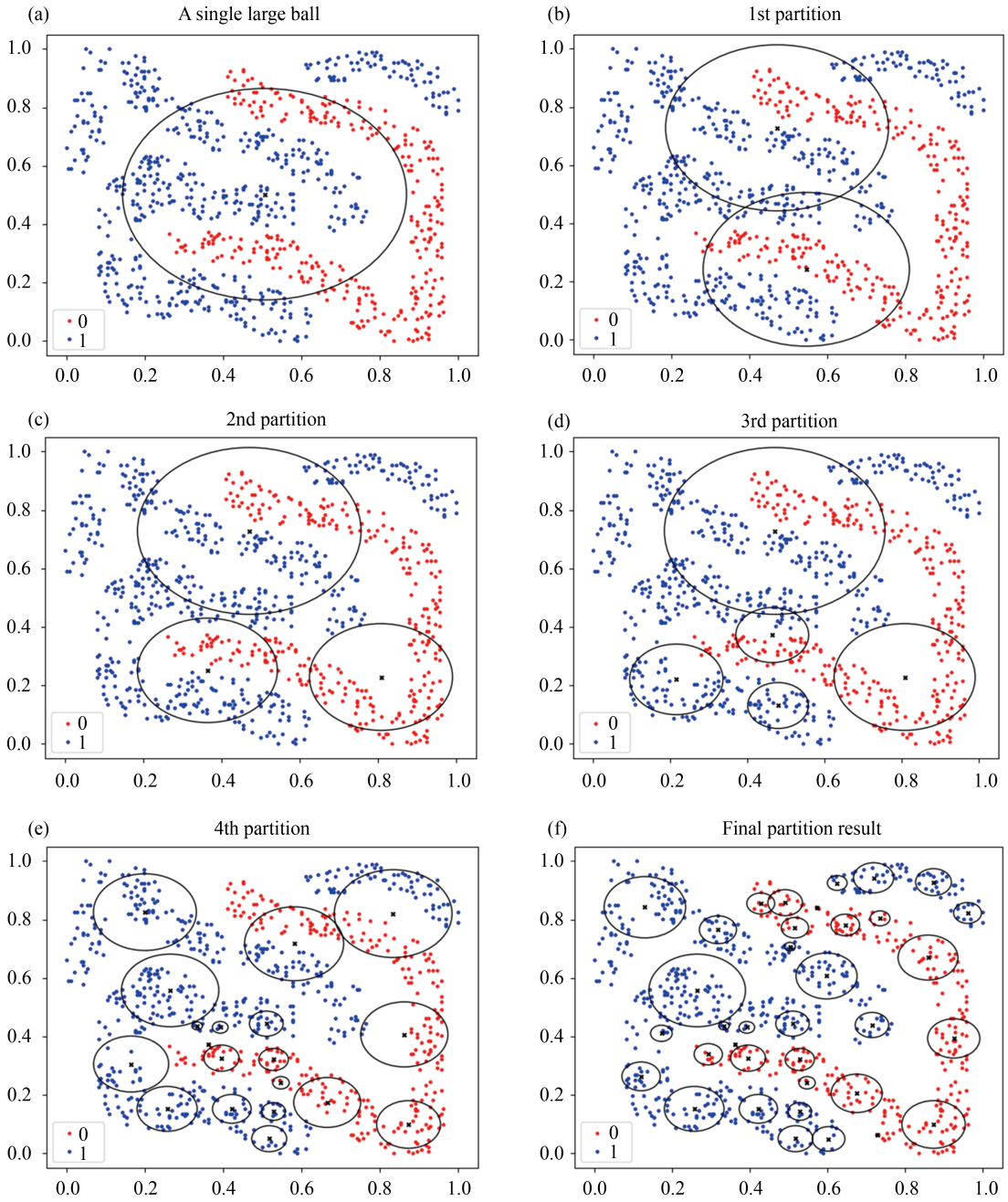

Figure 1. Process of granular ball generation on the dataset

2.5 *Basic related definitions*

According to their feature values for attribute B, objects are grouped into equivalence classes in the traditional rough set model [33], denoted by $[x]_B$. $\{[x_i]_B | x_i \in U\}$ representing a small subspace within a larger space, where $U = \{x_1, x_2, ..., x_n\}$ is a non-empty finite set of objects called the universe. Rough set theory is often applied to data mining and classification tasks. For subset X, the following are the definitions of lower and upper approximation:

$$\underline{B}X = \{[x_i]_B | [x_i]_B \subseteq X,\ x_i \in U\}, \tag{3}$$

$$\overline{B}X = \{[x_i]_B | [x_i]_B \cap X \neq \emptyset,\ x_i \in U\}. \tag{4}$$

The lower approximation signifies the largest union of granules within X, whereas the upper approximation represents the smallest union of granules in X. Consequently, the objects in U can be categorized into three distinct parts: the positive region, the boundary region, and the negative region.

$$POS_B(X) = \underline{B}X, \tag{5}$$

$$BND_B(X) = \overline{B}X - \underline{B}X, \tag{6}$$

$$NEG_B(X) = U - \overline{B}X. \tag{7}$$

The neighborhood rough set uses neighborhood relations to replace the traditional rough set based on equivalence relations.

Definition 1 (Dependency) [34] Assuming a feature subset $B \subseteq C$, the degree of dependence of B relative to D is defined as the proportion of consistent objects, as follows:

$$\gamma_B(D) = \frac{CARD(POS_B(D))}{CARD(U)}. \tag{8}$$

Feature selection aims to select a subset B from the feature set C such that the degree of dependence from B to D is maximized.

Definition 2 (Significance) [34] Given the conditional feature set $B(B \subseteq C)$ and the decision feature set D, the importance of feature f ($f \in B$) relative to B is defined as:

$$\delta_B(D,\ f) = \gamma_B(D) - \gamma_{B\setminus\{f\}}(D). \tag{9}$$

Definition 3 (Maximal-dependency) [34] Assuming C represents a feature set containing m conditional features, the goal of feature selection is to select a subset $S \subseteq C$ containing d $(d < m)$ features from C such that this subset has the maximum degree of dependence with respect to the decision set D and the minimum number of features.

$$\max\{D\} \cap \min\{d\}\, s.t. D = \gamma_B(D). \tag{10}$$

$D_\gamma = \gamma_B(D)$ represents the degree of dependence between the feature subset obtained according to Formula (9) and the target class label D .

Definition 4 (Maximal-relevance) [34] Maximum relevance refers to the process of maximizing the degree of dependency while searching for features, ensuring that the average dependency degree of each feature f relative to the target class D is maximized.

$$\min\{R\}\, s.t. R = \frac{1}{|S|}\sum_{f_i \in S} \gamma_{f_i}(D). \tag{11}$$

Here, S is the currently selected feature set.

However, the maximal relevance evaluation criterion can select features that have a high degree of dependence on the decision class, but it cannot eliminate redundancy within the selected feature subset.

Definition 5 (Maximal-significance) [34] Given a conditional feature set S and a decision feature set D, for feature $f \in S$, the importance of f relative to S is defined as:

$$\delta_S(D,\ f) = \gamma_S(D) - \gamma_{S\setminus\{f\}}(D). \tag{12}$$

By utilizing the importance of each feature relative to the feature set, we can measure the significance of each feature within the selected candidate subset. The maximal importance condition can be used to select mutually exclusive features as follows:

$$\max\{S\}\, s.t. S = \frac{1}{|S|}\sum_{f_i \in S} \delta_S(D). \tag{13}$$

In the context of OSFS, computing all potential groups of possible features using Formula (10) to maximize the dependency of the selected features is impractical. Therefore, when the feature stream arrives, we initially employ the "Maximal-relevance" criterion to filter out relevant features and eliminate those that are irrelevant. If a newly incoming feature enhances the dependency of the already selected subset, it is incorporated into the candidate set based on the "Maximal-dependency" criterion. Conversely, if the overall dependency remains constant upon adding the new feature, we first integrate this new feature with the currently selected subset of candidate features, and subsequently apply the "Maximal-significance" criterion to eliminate any insignificant features.

Since GB-OSFS uses granular balls to process data, the method of calculating dependency in GB will be different. In the next section, we will improve the original framework model of feature selection and integrate rough set and granular ball computing into the calculation of feature relevance and redundancy.

In the next section, we will improve the original framework model of feature selection and integrate rough set and granular ball computing into the calculation of feature relevance and redundancy.

3. OSFS methods based on GBC

3.1 *GB-OSFS*

In this section, we propose a OSFS algorithm based on granular ball computing, called GB-OSFS.

In this paper, we proposed a new framework for streaming feature selection algorithms based on granular ball computing [35] and NRS theory. When a new feature emerges, we use granular ball theory to split the data into a small number of granular balls. Then we calculate its correlation using Definition 6 and Formula (14) to divide the features into relevant and irrelevant features. And relevant features need to be analyzed using Definition 4 and Definition 5. Finally, we eliminate redundant features to obtain the optimal feature subset. The specific procedure is illustrated in Figure 2.

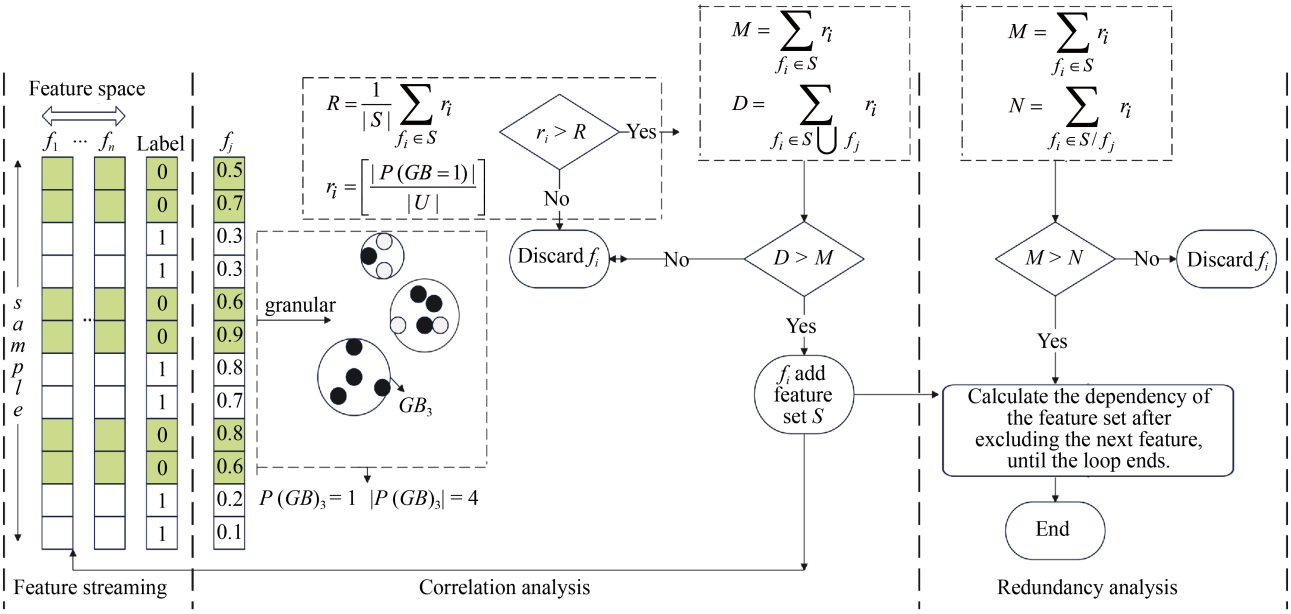

Figure 2. GB-OSFS algorithm framework

3.2 *Correlation analysis*

Definition 6 (Correlation in GB-OSFSB) Given a dataset $U = \{x_i,\ i = 1,\ \ldots,\ n\}$. Respectively, $GB_j(j = 1,\ 2,\ ...,\ k)$ is a granular ball generated on feature set B. $P(GB_j)$ represents the purity of the j-th particle sphere. The number of samples covered by particle spheres with a purity of 1 is approximated as B. Therefore, the dependency of B on D is as follows:

$$\gamma_B(D) = \frac{\left|P(GB_j = 1)\right|}{|U|}. \tag{14}$$

In the Equation, $\left|P(GB_j = 1)\right|$ and $|U|$ represent the number of samples contained in particle spheres with a purity of 1, and the total number of samples. To select the highly correlation features from the feature streaming, we calculate g for each features and compare with R. If g is greater than R, we consider the feature is highly correlation.

Algorithm 1 Correlation analysis

1. **Input:** X (Sample value), Y (Label value)
2. **Output:** Current selected feature set
3. Let *mdeps* = n, *num* = 0, *posnum* = m
4. A new feature f arrives
5. Generate the granular ball list *GB* under the feature f
6. **for each** in *GB*
7. **if** *purity* = 1
8. *num* += *ball.num*
9. **end if**
10. **end for**
11. **if** *num* > *n*
12. *Add f to the set S* $\longrightarrow s'$
13. **end if**

14. Generate the granular ball list GB_s under the feature s'
15. **for each** in GB_s
16. **if** $purity = 1$
17. $num \mathrel{+}= ball.num$
18. **end if**
19. **end for**
20. **if** $num > m$
21. $s.append\ (f)$
22. $mdeps = num/u,\ posnum = num, S = s'$
23. **end if**
24. **return** S

There is a feature streaming as $F = \{f_1,\ f_2,\ ...,\ f_n\}$ and the optimal feature set S, $S \subseteq F$. At time t, the arriving feature is f_t. If $g_t > R$, $f_t\ (1 \leq t \leq n)$ is high correlation feature. R is defined as follow:

$$R = \frac{1}{|S|}\gamma_S(D). \tag{15}$$

Then, add f to the selected feature subset and use Equation (10) to calculate the dependency degree of the selected features on D at this point, compared to when f was not included.

Algorithm 1 illustrates the method of calculating dependency using granular ball computing.

3.3 *Redundancy analysis*

Redundancy analysis is the calculation of the degree of similarity between different features. If two features describe concepts that are very similar, one of them should be removed.

Definition 7 (Redundancy analysis) Redundant features are eliminated using a first-in, first-out approach. In other words, when performing redundancy calculations, the most recently added feature in the selected feature set is excluded first. If the dependency degree at this point is less than that before exclusion, only the second feature is excluded, and the dependency degree is recalculated. This process continues in a similar manner thereafter. Assume a feature set $S = \{f_1,\ f_2,\ f_3,\ ...,\ f_n\}$, and the current feature set is $H = \{f_2,\ f_3,\ f_4,\ ...,\ f_n\}$. Then, calculate the dependency of the feature subsets S and H on the decision set D using Equation (15). If the dependency of S is less than that of H, we remove the feature f_1.

A more specific example illustrates the framework proposed in this paper. We assume a selected feature set $S = \{f_1,\ f_2,\ f_3,\ ...,\ f_n\}$. At time t, we will obtain a feature f_t. Firstly, we use the feature data of f_t to generate granular balls and we calculate the number of samples covered by the granular balls with a purity of 1. If this number of samples is greater than mdeps, we will include f_t into S. Then, we generate granular balls based on the feature data in S and calculate the number of samples covered by the granular balls with a purity of 1. If this value is greater than posnum, f_t will not be removed from S. The above steps constitute Algorithm 1. Then, set $H = \{f_2,\ f_3\ ,...,\ f_n,\ f_t\}$. Subsequently, use Algorithm 1 to calculate the sample coverage size for both S and H separately, if the former is larger, then f_1 is a redundant feature. Conversely, set $H = \{f_1,\ f_3,\ ...,\ f_n,\ f_t\}$. Then, repeat this calculation until all the features in S have been traversed, this step is part of Algorithm 2. Finally, after all the features have been processed through these two algorithms, the optimal feature subset will be selected.

The GB-OSFS algorithm can calculate the correlation between features and labels, allowing it to obtain features useful for classifying datasets. At the same time, the algorithm also assesses the correlation among different features to eliminate redundant features, thereby further improving prediction accuracy. The GB-OSFS algorithm is capable of selecting the best features that are most beneficial for classifying the dataset.

Algorithm 2 Redundancy analysis

1. **Input:** S (*Selected feature set*)

2. **Output:** Feature set after removing redundancies.
3. **for** i in $|S|$
4. According to Definition 7, the features in S are removed sequentially. $S \longrightarrow H$
5. **do** algorithmic 1 (6-10) on S and H
6. **if** $S.num < H.num$
7. $S = H$
8. $mdeps = S.num/|S|$, $posnum = S.num$
9. **end if**
10. **end for**
11. **return** S

4. Experiments

In the experiments, we mainly solve the following Research Questions (RQs):
RQ 1. Does GB-OSFS outperform other algorithms in the presence of noise?
RQ 2. Are there any factors that are irrelevant to the experimental results?
RQ 3. What are the influences of hyper-parameter p on the GB-OSFS?

4.1 *General settings*

Datasets. In this paper, we have selected elelven benchmark datasets from various sources. Datasets have different types and varying sizes. The main goal is to test whether there is a significant performance gap among the algorithms on different types of datasets. Table 2 lists the different datasets used in the experiment. To simulate a noisy environment, we intentionally corrupt a certain percentage of samples by altering their original labels to incorrect ones.

Table 2. Details of selected datasets

Dataset	Instance	Feature	Class
Gas sensor array under flow modulation	53	432	4
Magic	19,020	10	2
QSAR	1,055	41	2
Raisin dataset	900	7	2
Leukemia	72	7,129	2
DLBCL	77	5,469	2
Iono	351	34	2
Diabates	768	8	2
Colon	62	2,000	2
Lymphoma	62	4,026	3

Baselines. The algorithms were primarily compared based on prediction accuracy, which refers to the classification tendencies of classifiers trained using different algorithms and then compared with the actual results of testing predictions. The comparative experiments in this paper adopted all the methods listed in Table 3.

Table 3. Descriptions of all the involved models

OSFS [1]	A traditional feature selection method that compares the scores of different features to get the best selection.
Fast-OSFS [1]	This method is an advanced version of OSFS, which optimizes the selection features and can speed up most of the data training process.
SAOLA [4]	It is a method used for feature selection and sparse modeling. It achieves efficient feature selection and dimensionality reduction by iteratively selecting the most relevant features and performing sparse modeling using orthogonal least squares.
α-investing [5]	Compared with other algorithms, this algorithm requires to pre-set additional hyperparameters.
RHDOFS [28]	By introducing robust statistical methods, the impact of noise and outliers on feature selection is reduced.
OSFS-Vague [29]	A novel algorithm based on vague set theory.
GB-OSFS	A novel algorithms based on granular ball computing is proposed in this paper.

Implementation Details. We conduct statistical tests using the Friedman test based on prediction accuracy results [34]. The algorithm performs better the lower the Rank value. The lower the rank value, the higher the precision ranking, indicating better performance. We used the prediction results of three classifiers, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Gradient Boosting Regression Tree (GBRT), as the evaluation metrics for prediction accuracy. The experiments are performed on a computer with 16 GB RAM, 2.4 GHz, and Intel (R) i5-9300H processor.

4.2 *Comparison of the effectiveness of the GB-OSFS with other algorithms (RQ. 1)*

In this section, we compared the number of selected features, prediction accuracy, noise accuracy (for convenience of description, this paper refers to the prediction accuracy under noise as noise accuracy) of several other algorithm. Table 4 reports the number of features selected by each algorithm. GB-OSFS selects an acceptable number of features. Therefore, the following analysis emphasizes accuracy.

Table 4. The number of selected features

Dataset	GB-OSFS	OSFS	α-investing	SAOLA	Fast-OSFS	PHDOFS	OSFS-Vague
Magic	9	5	9	2	5	3	2
QSAR	9	6	33	4	6	8	3
Leukemia	6	3	12	3	2	4	3
Conlom	8	7	20	4	8	7	2
Raisin dataset	7	2	7	1	2	2	2
Iono	7	5	21	2	5	3	2
Gas sensor array under flow modulation	4	5	6	9	8	4	5
DLBCL	8	8	21	3	8	6	2
Diabates	8	4	8	3	4	2	1
Lymphoma	2	5	61	45	9	2	7
Average	6.8	5	17.7	7.6	5.7	4.1	2.9

4.2.1 *The prediction accuracy analysis for OSFS in the presence of noise*

In this section, we will compare the Noise prediction accuracy in these algorithms.

Tables 5 to 8 show the accuracy of various algorithms under different noise rates. It can be observed that GB-OSFS performs excellently in the presence of noise. And as the noise rate increases, the win/loss/Tie ratio of GB-OSFS gradually increases. This indicates that compared to other algorithms, GB-OSFS exhibits strong noise resistance. More specifically, Tables 5 demonstrates that GB-OSFS demonstrates a win/loss/Tie ratio of 88/51/41 against comparative algorithms at a

10% noise level. Notably, GB-OSFS achieves a statistically significant win/loss/Tie ratio of 18 : 6 : 6 against SAOLA. Furthermore, GB-OSFS achieves a Rank value of 2.58. Under 20% noise rate, GB-OSFSB demonstrates a win/loss/Tie ratio of 92/51/37 against comparative algorithms. GB-OSFS achieves win/draw/loss ratio of 16/8/6 against OSFS-Vague. When the noise rate reaches 30%, from Table 7, we can see that the win/loss/Tie ratio at this point is 110/28/42. Where GB-OSFS achieves win/draw/loss ratio of 19/5/6 against OSFS. And at this point, the Rank value of GB-OSFS is only 2.62, which is quite low compared to RHDOFS. When the noise rate reaches 40%, these differences are even more significant. The Rank value of GB-OSFS is not only as low as 2.06, but its overall win/loss/Tie ratio is also impressive at 131/12/37. Among them, the win/loss/Tie ratio of GB-OSFS against OSFS is a 23/1/6 and the win/loss/Tie ratio of FAST-OSFS against OSFS is a 22/1/7.

Table 5. Prediction accuracy of gb-osfs vs OSFS algorithms (10% noise rate)

dataset	Type	GB-OSFS (our method)	OSFS	α-investing	SAOLA	Fast-OSFS	RHDOFS	OSFS-Vague
QSAR	KNN	0.781	0.773	0.82	0.742	0.782	0.775	0.465
	SVM	0.663	0.663	0.663	0.663	0.663	0.663	0.663
	GBRT	0.842	0.83	0.865	0.79	0.845	0.842	0.679
Gas sensor array under flow modulation	KNN	0.873	0.619	0.844	0.641	0.727	0.863	0.866
	SVM	0.291	0.281	0.292	0.281	0.284	0.282	0.282
	GBRT	0.837	0.644	0.821	0.652	0.748	0.856	0.835
DLBCL	KNN	0.803	0.886	0.867	0.881	0.886	0.785	0.857
	SVM	0.752	0.752	0.752	0.752	0.752	0.752	0.752
	GBRT	0.801	0.869	0.809	0.87	0.866	0.834	0.859
Leukmia	KNN	0.84	0.911	0.788	0.898	0.915	0.923	0.929
	SVM	0.652	0.652	0.652	0.652	0.652	0.652	0.652
	GBRT	0.846	0.909	0.81	0.904	0.875	0.821	0.905
Conlon	KNN	0.676	0.758	0.692	0.69	0.777	0.702	0.732
	SVM	0.646	0.646	0.646	0.646	0.646	0.646	0.646
	GBRT	0.684	0.764	0.696	0.692	0.771	0.753	0.743
Lymphoma	KNN	0.916	0.902	0.897	0.902	0.904	0.923	0.912
	SVM	0.68	0.68	0.68	0.68	0.68	0.68	0.68
	GBRT	0.854	0.832	0.763	0.835	0.783	0.884	0.883
Raisin dataset	KNN	0.821	0.796	0.817	0.804	0.789	0.756	0.786
	SVM	0.793	0.586	0.792	0.502	0.669	0.673	0.698
	GBRT	0.848	0.843	0.846	0.835	0.845	0.813	0.839
Magic	KNN	0.807	0.771	0.808	0.749	0.771	0.696	0.745
	SVM	0.779	0.777	0.778	0.769	0.777	0.673	0.708
	GBRT	0.873	0.838	0.872	0.82	0.838	0.742	0.811
Iomo	KNN	0.864	0.883	0.893	0.763	0.883	0.835	0.869
	SVM	0.641	0.681	0.699	0.625	0.684	0.667	0.641
	GBRT	0.886	0.916	0.924	0.802	0.921	0.865	0.890
Diabates	KNN	0.699	0.692	0.681	0.671	0.690	0.558	0.354
	SVM	0.651	0.651	0.651	0.651	0.651	0.651	0.651
	GBRT	0.762	0.748	0.758	0.735	0.747	0.663	0.657
Static	Rank	2.58	5.02	4.03	3.5	4.45	4.05	4.37
	Win/Loss/Tie	88/51/41	15/9/6	13/11/6	18/6/6	13/11/6	15/8/7	14/6/10

Table 6. Prediction accuracy of gb-osfs vs OSFS algorithms (20% noise rate)

dataset	Type	GB-OSFS (our method)	OSFS	α-investing	SAOLA	Fast-OSFS	RHDOFS	OSFS-Vague
QSAR	KNN	0.786	0.764	0.816	0.7	0.78	0.765	0.516
	SVM	0.663	0.663	0.663	0.663	0.663	0.663	0.663
	GBRT	0.833	0.816	0.857	0.774	0.837	0.836	0.683
Gas sensor array under flow modulation	KNN	0.807	0.593	0.861	0.547	0.656	0.825	0.832
	SVM	0.292	0.279	0.292	0.284	0.286	0.286	0.284
	GBRT	0.809	0.588	0.822	0.606	0.645	0.749	0.808
DLBCL	KNN	0.734	0.777	0.797	0.818	0.806	0.745	0.812
	SVM	0.752	0.752	0.752	0.752	0.752	0.752	0.752
	GBRT	0.757	0.808	0.779	0.805	0.817	0.806	0.842
Leukmia	KNN	0.771	0.781	0.7	0.843	0.768	0.816	0.813
	SVM	0.652	0.652	0.652	0.652	0.652	0.652	0.652
	GBRT	0.787	0.814	0.7	0.833	0.818	0.783	0.858
Conlon	KNN	0.653	0.702	0.558	0.658	0.727	0.632	0.648
	SVM	0.646	0.646	0.646	0.646	0.646	0.646	0.646
	GBRT	0.641	0.712	0.559	0.605	0.715	0.625	0.661
Lymphoma	KNN	0.881	0.876	0.863	0.88	0.883	0.897	0.879
	SVM	0.68	0.68	0.68	0.68	0.68	0.68	0.68
	GBRT	0.826	0.822	0.756	0.805	0.765	0.832	0.863
Raisin dataset	KNN	0.821	0.802	0.818	0.804	0.8	0.785	0.791
	SVM	0.793	0.52	0.792	0.502	0.552	0.663	0.699
	GBRT	0.848	0.838	0.846	0.835	0.839	0.803	0.839
Magic	KNN	0.807	0.77	0.807	0.748	0.77	0.683	0.741
	SVM	0.779	0.777	0.778	0.769	0.777	0.689	0.721
	GBRT	0.873	0.836	0.872	0.82	0.836	0.736	0.811
Iomo	KNN	0.839	0.858	0.89	0.733	0.885	0.805	0.83
	SVM	0.648	0.674	0.678	0.62	0.68	0.667	0.642
	GBRT	0.858	0.89	0.924	0.764	0.913	0.933	0.868
Diabates	KNN	0.699	0.693	0.689	0.581	0.69	0.547	0.413
	SVM	0.651	0.651	0.651	0.651	0.651	0.651	0.651
	GBRT	0.762	0.746	0.753	0.729	0.746	0.667	0.653
Static	Rank	3.3	4.17	3.47	4.78	3.43	4.52	4.33
	Win/Loss/Tie	92/51/37	15/9/6	13/10/7	19/5/6	14/10/6	15/9/6	16/8/6

Table 7. Prediction accuracy of gb-osfs vs OSFS algorithms (30% noise rate)

dataset	Type	GB-OSFS (our method)	OSFS	α-investing	SAOLA	Fast-OSFS	RHDOFS	OSFS-Vague
QSAR	KNN	0.812	0.748	0.805	0.649	0.765	0.745	0.577
	SVM	0.663	0.663	0.663	0.663	0.663	0.663	0.663
	GBRT	0.86	0.796	0.845	0.75	0.821	0.802	0.719
Gas sensor array under flow modulation	KNN	0.765	0.591	0.796	0.528	0.586	0.773	0.822
	SVM	0.289	0.289	0.292	0.286	0.289	0.286	0.296
	GBRT	0.754	0.578	0.733	0.524	0.613	0.735	0.784
DLBCL	KNN	0.726	0.705	0.643	0.734	0.726	0.713	0.727
	SVM	0.752	0.752	0.752	0.752	0.752	0.752	0.752
	GBRT	0.744	0.721	0.656	0.74	0.772	0.732	0.743
Leukmia	KNN	0.717	0.689	0.655	0.709	0.695	0.702	0.692
	SVM	0.652	0.652	0.652	0.652	0.652	0.652	0.652
	GBRT	0.724	0.747	0.659	0.765	0.72	0.722	0.727
Conlon	KNN	0.658	0.681	0.538	0.648	0.678	0.622	0.606
	SVM	0.646	0.646	0.646	0.646	0.646	0.646	0.646
	GBRT	0.648	0.661	0.539	0.626	0.672	0.632	0.632
Lymphoma	KNN	0.847	0.843	0.845	0.850	0.856	0.842	0.84
	SVM	0.68	0.68	0.68	0.68	0.68	0.68	0.68
	GBRT	0.811	0.805	0.766	0.783	0.755	0.813	0.836
Raisin dataset	KNN	0.821	0.792	0.805	0.801	0.799	0.784	0.789
	SVM	0.793	0.531	0.764	0.5	0.596	0.653	0.722
	GBRT	0.848	0.84	0.844	0.836	0.84	0.805	0.835
Magic	KNN	0.807	0.769	0.807	0.749	0.769	0.663	0.707
	SVM	0.779	0.776	0.778	0.769	0.776	0.685	0.68
	GBRT	0.873	0.836	0.872	0.82	0.836	0.723	0.784
Iomo	KNN	0.839	0.818	0.842	0.721	0.834	0.778	0.81
	SVM	0.641	0.643	0.671	0.677	0.645	0.665	0.641
	GBRT	0.863	0.853	0.876	0.752	0.863	0.863	0.845
Diabates	KNN	0.699	0.675	0.678	0.576	0.689	0.532	0.419
	SVM	0.651	0.651	0.651	0.651	0.651	0.651	0.651
	GBRT	0.762	0.742	0.751	0.722	0.743	0.654	0.648
Static	Rank	2.62	4.23	3.62	4.55	3.62	4.77	4.60
	Win/Loss/Tie	110/28/42	19/5/6	18/5/7	19/5/6	17/4/9	20/3/7	17/6/7

Table 8. Prediction accuracy of gb-osfs vs OSFS algorithms (40% noise rate)

dataset	Type	GB-OSFS (our method)	OSFS	α-investing	SAOLA	Fast-OSFS	RHDOFS	OSFS-Vague
QSAR	KNN	0.817	0.623	0.703	0.575	0.669	0.705	0.632
	SVM	0.663	0.663	0.663	0.663	0.663	0.663	0.663
	GBRT	0.865	0.711	0.759	0.711	0.743	0.786	0.727
Gas sensor array under flow modulation	KNN	0.71	0.513	0.802	0.484	0.635	0.732	0.772
	SVM	0.292	0.291	0.292	0.282	0.286	0.286	0.289
	GBRT	0.741	0.523	0.772	0.482	0.641	0.705	0.738
DLBCL	KNN	0.71	0.648	0.628	0.682	0.692	0.692	0.694
	SVM	0.752	0.752	0.752	0.752	0.752	0.752	0.752
	GBRT	0.74	0.682	0.648	0.71	0.735	0.711	0.72
Leukmia	KNN	0.65	0.58	0.586	0.609	0.625	0.632	0.645
	SVM	0.652	0.652	0.652	0.652	0.652	0.652	0.652
	GBRT	0.629	0.594	0.583	0.646	0.689	0.623	0.682
Conlon	KNN	0.632	0.602	0.515	0.609	0.607	0.604	0.622
	SVM	0.646	0.646	0.646	0.646	0.646	0.646	0.646
	GBRT	0.606	0.618	0.507	0.603	0.622	0.583	0.643
Lymphoma	KNN	0.785	0.765	0.772	0.753	0.756	0.76	0.783
	SVM	0.68	0.68	0.68	0.68	0.68	0.68	0.68
	GBRT	0.768	0.735	0.735	0.705	0.712	0.753	0.762
Raisin dataset	KNN	0.821	0.765	0.806	0.756	0.775	0.752	0.796
	SVM	0.793	0.502	0.758	0.491	0.586	0.632	0.719
	GBRT	0.848	0.824	0.844	0.807	0.819	0.792	0.839
Magic	KNN	0.807	0.756	0.806	0.742	0.756	0.656	0.693
	SVM	0.779	0.771	0.777	0.767	0.771	0.672	0.705
	GBRT	0.873	0.826	0.862	0.815	0.826	0.711	0.77
Iomo	KNN	0.863	0.788	0.796	0.705	0.806	0.78	0.828
	SVM	0.642	0.641	0.653	0.663	0.641	0.666	0.641
	GBRT	0.877	0.809	0.841	0.732	0.829	0.872	0.85
Diabates	KNN	0.68	0.594	0.667	0.529	0.598	0.533	0.583
	SVM	0.651	0.651	0.651	0.651	0.651	0.651	0.651
	GBRT	0.757	0.699	0.732	0.7	0.711	0.625	0.677
Static	Rank	2.05	4.85	3.63	5.32	4.02	4.53	3.60
	Win/Loss/Tie	131/12/37	23/1/6	22/1/7	121/3/6	22/2/6	22/2/6	21/3/6

Figure 3 shows the ranking of prediction accuracy for different algorithms under various noise rates. We manually introduced noise proportions of 5%, 10%, 15%, 20%, 25%, 30%, 35%, and 40% on each dataset. The procedure entailed the following steps: (1) Randomly selecting samples from each dataset and altering their labels; (2) Conducting 10-fold cross-validation on the datasets with varying noise ratios to evaluate the robustness of the classifiers based on test accuracy.As can be seen from Figure 3, under 10% noise, GB-OSFS ranks first in prediction accuracy on two datasets and second on four datasets as well as under 20% noise. But under 40% noise, GB-OSFS ranks first in prediction accuracy on eight datasets and second on two datasets. And it can be seen that as the noise rate increases, more algorithms, including GB-OSFS, rank higher in noise accuracy. Figure 4 illustrate the differences in prediction accuracy among different algorithms at different noise rates. Figure 4a and Figure 4b indicate that at lower noise rates, the noise resistance of

GB-OSFS is not significantly different from that of Fast-osfs, α-investing, and OSFS-Vague. However, Figure 4c and Figure 4d show that under high noise conditions, GB-OSFS far outperforms the other six algorithms.

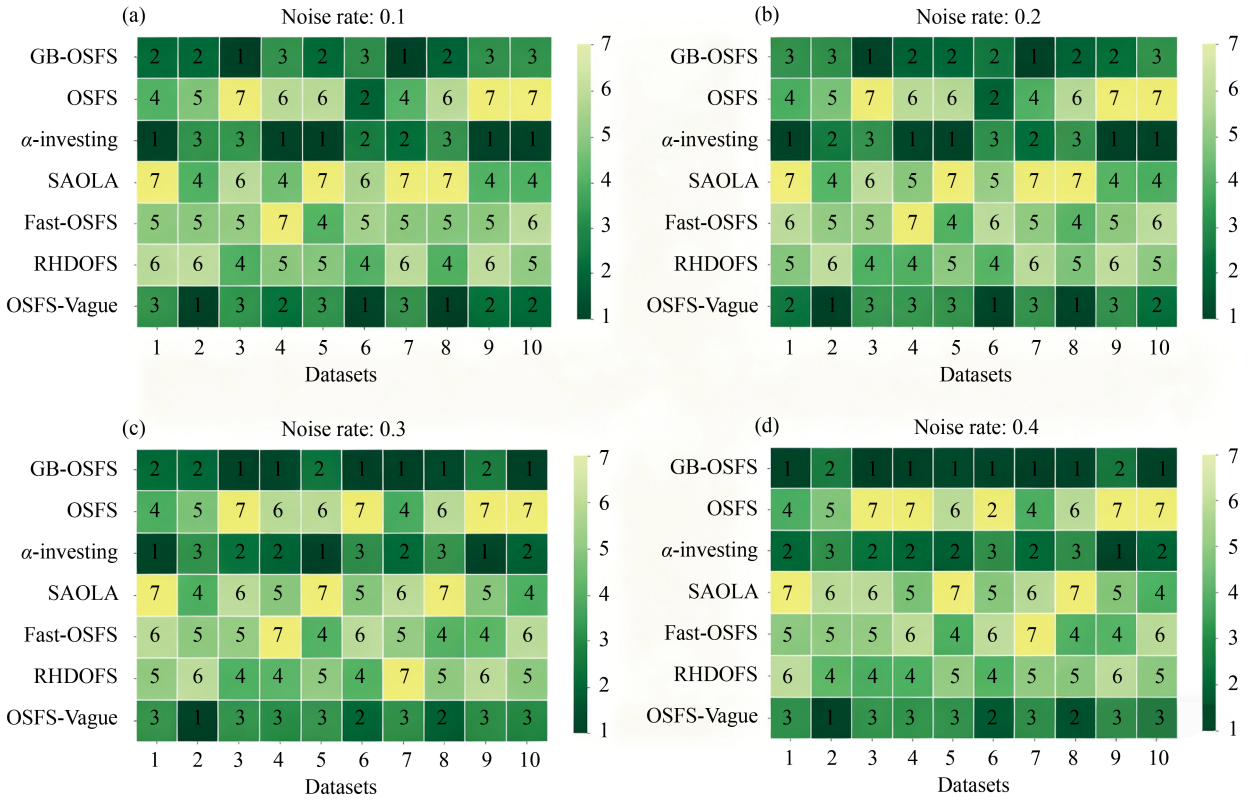

Figure 3. The rank order of accurary under different noise rates

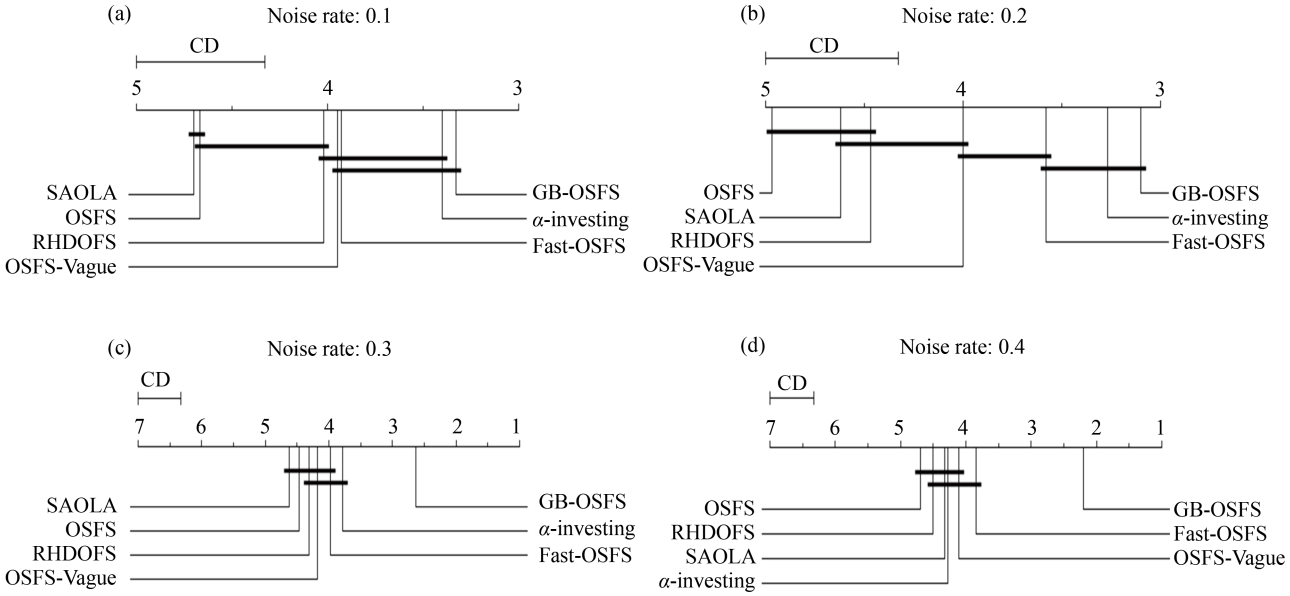

Figure 4. The rank order of accurary under different noise rates

4.2.2 *Prediction accuracy for OSFS*

To evaluate the comparative advantages of our proposed feature method, we conducted comprehensive prediction accuracy experiments across multiple feature selection algorithms. The evaluation framework employed three distinct classifiers: KNN, SVM, and GBRT. To ensure fair and consistent comparisons, we maintained identical hyperparameter configurations for all classifiers throughout the evaluation process, regardless of the feature selection algorithm being tested.

Table 9. Prediction accuracy of GB-OSFS vs OSFS algorithms

dataset	Type	GB-OSFS (our method)	OSFS	α-investing	SAOLA	Fast-OSFS	RHDOFS	OSFS-Vague
Lymphoma	KNN	1.000	0.985	0.968	0.985	1.000	0.983	0.968
	SVM	0.680	0.680	0.680	0.941	0.680	0.680	0.680
	GBRT	0.969	0.905	0.742	0.937	0.854	0.983	0.968
Gas sensor array under flow modulation	KNN	0.950	0.505	0.877	0.691	0.827	0.913	0.930
	SVM	0.292	0.276	0.292	0.276	0.276	0.276	0.276
	GBRT	0.862	0.522	0.862	0.739	0.794	0.895	0.841
Leumia	KNN	1.000	0.959	0.958	0.945	0.987	1.000	0.919
	SVM	0.652	0.652	0.652	0.652	0.652	0.652	0.652
	GBRT	0.932	0.946	0.904	0.918	0.904	0.903	0.891
DLBCL	KNN	0.974	0.898	0.948	0.987	0.962	0.843	0.872
	SVM	0.752	0.752	0.752	0.752	0.752	0.752	0.752
	GBRT	0.933	0.884	0.855	0.842	0.870	0.935	0.935
Conlon	KNN	0.837	0.819	0.626	0.662	0.805	0.806	0.837
	SVM	0.646	0.646	0.646	0.760	0.646	0.646	0.646
	GBRT	0.805	0.756	0.692	0.756	0.773	0.835	0.822
Magic	KNN	0.807	0.769	0.808	0.748	0.769	0.681	0.656
	SVM	0.779	0.778	0.779	0.760	0.778	0.696	0.685
	GBRT	0.873	0.837	0.872	0.820	0.837	0.748	0.735
QSAR	KNN	0.776	0.775	0.824	0.782	0.775	0.792	0.411
	SVM	0.663	0.663	0.663	0.773	0.663	0.663	0.663
	GBRT	0.821	0.846	0.864	0.828	0.845	0.846	0.662
Raisin dataset	KNN	0.821	0.795	0.821	0.806	0.786	0.775	0.783
	SVM	0.793	0.578	0.793	0.505	0.663	0.676	0.700
	GBRT	0.848	0.855	0.845	0.835	0.847	0.825	0.838
Iono	KNN	0.895	0.883	0.883	0.774	0.883	0.849	0.854
	SVM	0.692	0.700	0.732	0.641	0.700	0.667	0.641
	GBRT	0.923	0.931	0.923	0.820	0.934	0.898	0.900
Diabates	KNN	0.699	0.687	0.699	0.691	0.687	0.562	0.354
	SVM	0.651	0.651	0.651	0.651	0.651	0.651	0.651
	GBRT	0.762	0.747	0.759	0.750	0.746	0.671	0.657

Table 9 shows the prediction accuracy of different algorithms on ten datasets. GB-OSFS has a significantly higher average accuracy than other algorithms and has a high win/loss/Tie ratio of 21/2/7 compared with OSFS-Vague algorithm, but this is not significant when compared with α-investing algorithm. The win/loss/Tie ratio of GB-OSFS against the

other algorithms falls between these two algorithms. Although GB-OSFS does not have the highest prediction accuracy among all datasets, its overall win/loss/Tie ratio is 110/25/45, as seen from Table 9.

4.3 *Ablation experiment (RQ. 2)*

In this section, we separately removed the granular ball module and the redundant feature filtering module to conduct experiments on the performance of feature selection. In the figure below, "unfilter" denotes the variant of the GB-OSFS algorithm where only the feature filtering module is removed, "filter" corresponds to the experimental results of the complete GB-OSFS method, while "OSFS" serves as the baseline method, representing the performance of the traditional OSFS algorithm without the granular ball computing technique.

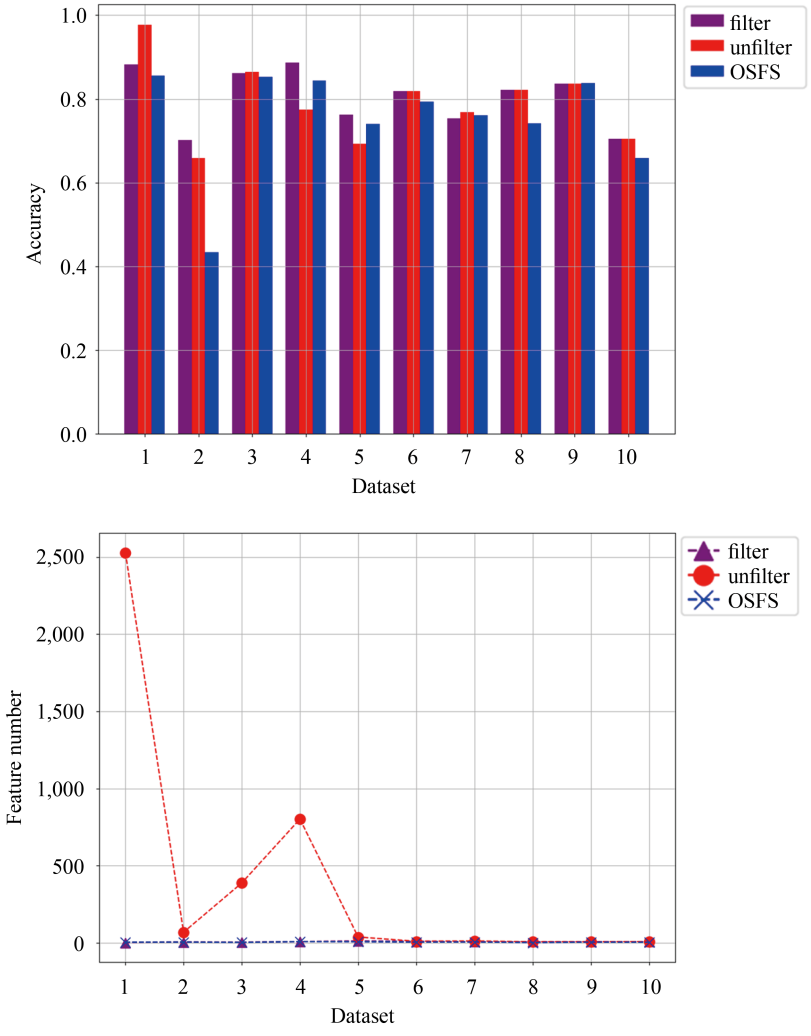

Figure 5. Accuracy on the dataset and the number of selected features

From Figure 5, it can be observed that although the accuracy does not decrease significantly when redundancy filtering is not performed, the number of selected features increases substantially. This increase in the number of features can increase the difficulty of handling the problem. Additionally, when granular balls are not introduced for feature selection, the accuracy decreases.

4.4 *Parameter sensitivity experiments of GB-OSFS (RQ. 3)*

In this section, we will investigate how the parameter P affects GB-OSFS. The p-value ranges from 0.6 to 1. According to Figure 6, it can be observed that the precision is optimal when $P = 1$. This is because higher purity of the particles leads to better quality. However, the decrease in precision is not significant when P is 0.9 on same datasets, indicating good particle quality at these values. On the other hand, when P is 0.7 and 0.6, there is a noticeable change in precision. This is due to the low quality of the particles at these values, which severely impacts the accuracy when using such low-quality particles for feature selection. Figure 6 shows that the performance is best when $P = 1$.

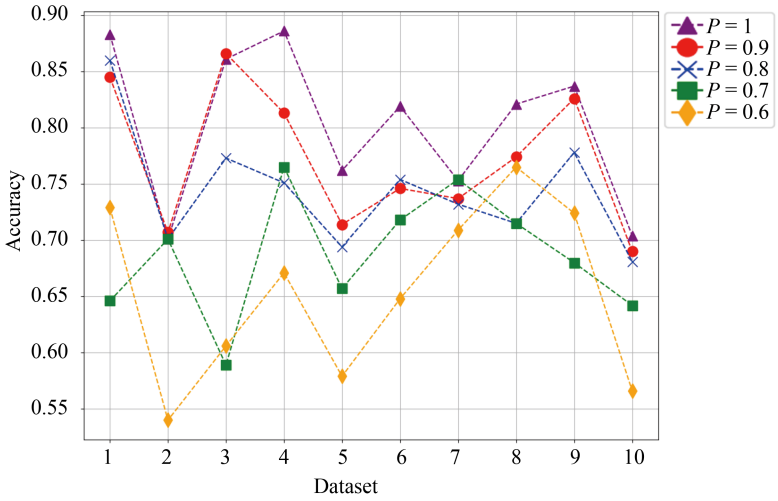

Figure 6. The prediction average accuracy under different P values

4.5 *Summary of experiments*

In this section, the advantages and disadvantages of each algorithm are comprehensively explained through the test of accuracy and number of feature selection.

Although α-investing has the second highest prediction accuracy, and also selects a small number of features, but its accuracy is not better than GB-OSFS.

OSFS is a classic algorithm for streaming feature selection, but it is at a disadvantage in terms of accuracy, with its only advantage being in the number of features selected.

Fast-OSFS has made a series of optimizations to OSFS, and although it slightly outperforms OSFS, its accuracy still does not reach the level of GB-OSFS.

α-investing is distinctly different from several other streaming feature selection algorithms in that it requires setting appropriate parameters based on the characteristics of the dataset. However, data is complex and variable, making it impossible to set suitable parameters for every dataset. Moreover, its accuracy is not better than that of GB-OSFS, and it has a larger number of selected features.

GB-OSFS outperforms other algorithms in terms of prediction accuracy, selects a moderate number of features, and its accuracy under noise is far superior to that of other algorithms. In summary, the GB-OSFS algorithm is superior to other algorithms in a comprehensive situation because it surpasses other algorithms in terms of prediction accuracy for the majority of cases, and notably, its accuracy under noisy conditions is far higher than that of other algorithms. Additionally, the number of features it selects is moderate. However, the GB-OSFS algorithm still has certain shortcomings; it is at a disadvantage in terms of the number of features it selects. This will be the direction of our future research.

5. Conclusion

In this paper, we have proposed GB-OSFS, a novel OSFS algorithm grounded in GBC. The key innovation of our work lies in the conceptual shift from point-wise to granule-wise processing. By representing the data universe with adaptive granular balls and utilizing purity-stable granules for feature evaluation, GB-OSFS effectively addresses two critical limitations of prevailing methods: sensitivity to noise and the reliance on pre-defined hyperparameters. Our comprehensive experimental results demonstrate that GB-OSFS consistently outperforms state-of-the-art algorithms in terms of prediction accuracy, particularly in noisy environments, while maintaining a competitive number of selected features. Despite its compelling advantages, GB-OSFS is not without limitations, which also open avenues for fruitful future research:

1) The present work assumes a relatively stable feature space. An essential extension is to investigate mechanisms for detecting and adapting to concept drift within the granular-ball framework, ensuring the model's relevance in non-stationary streams.

2) The splitting process, while efficient, could be further optimized for ultra-high-speed data streams. Future work will explore incremental granular-ball updating strategies and parallelized implementations for distributed computing environments.

3) Exploring the integration of GB-OSFS as a preprocessing layer within deep learning architectures for streaming data could lead to end-to-end learning models that are both efficient and interpretable.

In application field, GB-OSFS can be applied in economics field, for real-time feature selection in high-frequency trading data or macroeconomic indicator streams. In engineering field, in sensor networks for IoT devices, where streaming features are generated continuously and require efficient, robust selection. In data science filed, for automated feature selection in large-scale data pipelines, enhancing model interpretability and reducing computational costs. These applications highlight the practicality and broad relevance of our method beyond theoretical contexts.

Acknowledgement

This work was supported by the National Science Foundation of China (Grant number 62466063).

Disclosure

The authors have nothing to disclose beyond the information provided in the Acknowledgement and Conflict of interest sections.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Wu X, Yu K, Ding W, Wang H, Zhu X. Online feature selection with streaming features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013; 35(5): 1178-1192. Available from: https://doi.org/10.1109/TPAMI.2012.197.

[2] Chandrashekar G, Sahin F. A survey on feature selection methods. *Computers & Electrical Engineering*. 2014; 40(1): 16-28. Available from: https://doi.org/10.1016/j.compeleceng.2013.11.024.

[3] Kira K, Rendell LA. A practical approach to feature selection. In: *ML92: Proceedings of The Ninth International Workshop on Machine Learning*. Aberdeen: Scotland; 1992. p.249-256.

[4] Yu K, Wu X, Ding W, Pei J. Scalable and accurate online feature selection for big data. *ACM Transactions on Knowledge Discovery from Data*. 2016; 11(2): 1-39. Available from: https://doi.org/10.1145/2976744.

[5] Zhou J, Stine RA, Foster DP, Ungar LH. Streamwise feature selection. *Journal of Machine Learning Research*. 2006; 7(1): 1861-1885.

[6] Li W, Zhou H, Xu W, Wang XZ. Interval dominance-based feature selection for interval-valued ordered data. *IEEE Transactions on Neural Networks and Learning Systems*. 2023; 34(10): 6898-6912. Available from: https://doi.org/10.1109/TNNLS.2022.3184120.

[7] Zhang P, Li T, Yuan Z, Luo C, Liu K, Yang X. Heterogeneous feature selection based on neighborhood combination entropy. *IEEE Transactions on Neural Networks and Learning Systems*. 2024; 35(3): 3514-3527. Available from: https://doi.org/10.1109/TNNLS.2022.3193929.

[8] Luo C, Wang S, Li T, Chen H, Lv J, Yi Z. Large-scale meta-heuristic feature selection based on BPSO assisted rough hypercuboid approach. *IEEE Transactions on Neural Networks and Learning Systems*. 2023; 34(12): 10889-10903. Available from: https://doi.org/10.1109/TNNLS.2022.3171614.

[9] Lin T. Neighborhood systems-a qualitative theory for fuzzy and rough sets. *Advances in Machine Intelligence and Soft Computing*. 1997; 4: 132-155.

[10] Yao Y. Rough sets, neighborhood systems and granular computing. In: *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering*. Canada: Edmonton; 1999. p.1553-1558.

[11] Hu Q, Yu D, Xie Z. Numerical attribute reduction based on neighborhood granulation and rough approximation. *Journal of Software*. 2008; 19(3): 640-649. Available from: https://doi.org/10.3724/SP.J.1001.2008.00640.

[12] Lin T. Granular computing on binary relations. In: *Proceedings of the Third International Conference on Rough Sets and Current Trends in Computing*. USA: Malvern; 2002. p.296-299.

[13] Xia S, Liu Y, Ding X, Wang G, Yu H, Luo Y. Granular ball computing classifiers for efficient, scalable and robust learning. *Information Sciences*. 2019; 483: 136-152. Available from: https://doi.org/10.1016/j.ins.2019.01.010.

[14] Xia S, Bai X, Wang G, Cheng Y, Meng D, Gao X, et al. An efficient and accurate rough set for feature selection, classification, and knowledge representation. *IEEE Transactions on Knowledge and Data Engineering*. 2024; 35(4): 5319-5331. Available from: https://doi.org/10.1109/TKDE.2022.3220200.

[15] Cheng D, Liu S, Xia S, Wang G. Granular-ball computing-based manifold clustering algorithms for ultra-scalable data. *Expert Systems with Applications*. 2024; 247: 123313. Available from: https://doi.org/10.1016/j.eswa.2024.123313.

[16] Xie J, Jiang L, Xia SY, Xiang XX, Wang GY. An adaptive density clustering approach with multi-granularity fusion. *Information Fusion*. 2024; 106: 102273. Available from: https://doi.org/10.1016/j.inffus.2024.102273.

[17] Pan J, Lang G, Xiao Q, Yang T. A framework of granular-ball generation for classification via granularity tuning. *Applied Intelligence*. 2025; 55(1): 63. Available from: https://doi.org/10.1007/s10489-024-05904-1.

[18] Sajid M, Quadir A, Tanveer M, Initiative ADN. GB-RVFL: Fusion of randomized neural network and granular ball computing. *Pattern Recognition*. 2025; 159: 111142. Available from: https://doi.org/10.1016/j.patcog.2024.111142.

[19] Gao C, Tan X, Zhou J, Ding W, Pedrycz W. Fuzzy granule density-based outlier detection with multi-scale granular balls. *IEEE Transactions on Knowledge and Data Engineering*. 2025; 37(3): 1182-1191. Available from: https://doi.org/10.1109/TKDE.2024.3525003.

[20] Hall MA. Correlation-based feature selection of discrete and numeric class machine learning. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco, CA, United States: Morgan Kaufmann Publishers Inc.; p.359-366.

[21] Liu H, Motoda H. *Computational Methods of Feature Selection*. 1st ed. Chapman & Hall/CRC; 2007.

[22] Dash M, Liu H. Feature selection for classification. *Intelligent Data Analysis*. 1997; 1(1-4): 131-156. Available from: https://doi.org/10.1016/S1088-467X(97)00008-5.

[23] Sun L, Yin T, Ding W, Qian Y, Xu J. Feature selection with missing labels using multilabel fuzzy neighborhood rough sets and maximum relevance minimum redundancy. *IEEE Transactions on Fuzzy Systems*. 2022; 30(5): 1197-1211. Available from: https://doi.org/10.1109/TFUZZ.2021.3053844.

[24] Sang B, Chen H, Yang L, Li T, Xu W, Luo C. Feature selection for dynamic interval-valued ordered data based on fuzzy dominance neighborhood rough set. *Knowledge-Based Systems*. 2021; 227: 107223. Available from: https://doi.org/10.1016/j.knosys.2021.107223.

[25] Zou H, Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*. 2005; 67(2): 301-320. Available from: https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[26] Barddal JP, Enembreck F, Gomes HM, Bifet A, Pfahringer B. Boosting decision stumps for dynamic feature selection on data streams. *Information Systems*. 2019; 83: 13-29. Available from: https://doi.org/10.1016/j.is.2019.02.003.

[27] De Moraes MB, Gradvohl ALS. A comparative study of feature selection methods for binary text streams classification. *Evolving Systems*. 2021; 12(4): 997-1013. Available from: https://doi.org/10.1007/s12530-020-09357-y.

[28] Luo C, Wang S, Li T, Chen H, Lv J, Yi Z. RHDOFS: A distributed online algorithm towards scalable streaming feature selection. *IEEE Transactions on Parallel and Distributed Systems*. 2023; 34(6): 1830-1847. Available from: https://doi.org/10.1109/TPDS.2023.3265974.

[29] Yang J, Wang Z, Wang G, Liu Y, He Y, Wu D. OSFS-Vague: OSFS algorithm based on vague set. *CAAI Transactions on Intelligence Technology*. 2024; 9(6): 1451-1466. Available from: https://doi.org/10.1049/cit2.12327.

[30] Gau WL, Buehrer DJ. Vague sets. *IEEE Transactions on Systems, Man, and Cybernetics*. 1993; 23(2): 610-614. Available from: https://doi.org/10.1109/21.229476.

[31] Chen L. Topological structure in visual perception. *Science*. 1982; 218(4573): 699-700. Available from: https://doi.org/10.1126/science.7134969.

[32] Wang G. DGCC: Data-driven granular cognitive computing. *Granular Computing*. 2017; 2(4): 343-355. Available from: https://doi.org/10.1007/s41066-017-0048-3.

[33] Zhou P, Hu X, Li P, Wu X. OSFS using adapted neighborhood rough set. *Information Sciences*. 2019; 481: 258-279. Available from: https://doi.org/10.1016/j.ins.2018.12.074.

[34] Xia S, Wang C, Wang G, Gao X, Ding W, Yu J, et al. GBRS: A unified granular-ball learning model of pawlak rough set and neighborhood rough set. *IEEE Transactions on Neural Networks and Learning Systems*. 2025; 36(1): 1719-1733. Available from: https://doi.org/10.1109/TNNLS.2023.3325199.

[35] Demšar J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*. 2006; 7: 1-30.