



PSO-NRS: an online group feature selection algorithm based on PSO multi-objective optimization

Shunpan Liang¹ · Ze Liu¹ · Dianlong You¹ · Weiwei Pan¹ · Junjie Zhao¹ · Yefan Cao²

Accepted: 17 October 2022 / Published online: 10 November 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Online streaming feature selection plays an important role in dealing with multi-dimensional data problems. Many online streaming feature selection algorithms have been combined with evolutionary algorithms (EA) and play an important role, however, most of them use single-objective optimization which has some limitations. Meanwhile, they ignore the interaction between features. The combination of features with each other may generate higher relevance. Therefore, this paper proposes a new online group feature selection algorithm PSO-NRS by fusing particle swarm optimization (PSO) algorithm and neighborhood rough set theory (NRS). PSO-NRS is able to select the set of features that are highly correlated with labels by combining features randomly. Using NRS for online feature selection does not require any domain knowledge, which makes PSO-NRS generalize better and can handle different types of data. PSO-NRS applies two layers of filtering for online feature selection. In the first filtering layer, two objective functions are designed and multi-objective optimization by particle swarm is used to select the set of features with the highest relevance. In the second filtering layer, a search strategy is defined using a rough set-based evaluation method to complete the final feature selection. The interactions between features are considered and redundant features are removed during the two filtering layers. Finally, PSO-NRS is experimented on 14 different types of datasets and compared with six state-of-the-art online feature selection algorithms to strongly validate the effectiveness and generalization of this algorithm.

Keywords Particle swarm optimization · Neighborhood rough set · Multi-objective optimization · Feature interaction · Streaming feature

1 Introduction

Feature selection as an effective tool for processing high-dimensional data has been used in various aspects of research and has shown good performance [1–3]. An online feature selection method in the form of “streaming” features is becoming popular to make feature selection more relevant. In real life, features are generated all the time and appear one after the other with time. For example, *Weibo* hot topics are updated every minute, and when a hot topic appears, it may contain a new keyword, which is the key

feature to identify the hot topic [4]. The trend of streaming feature popularity makes some feature selection methods that require a complete feature space no longer applicable, because it is impossible to wait for all the features to arrive before processing them uniformly.

To handle data with “streaming” features, researchers have conducted a lot of research and proposed many online feature selection methods, such as Online Streaming Feature Selection (OSFS) [5], a new online streaming feature selection method based on Gap relation (OFS-A3M) [6], Alpha-Investing [7], etc. OSFS [5] is the first form of streaming features and designed a novel framework to handle streaming features. However, the parameter alpha must be specified in advance before feature selection, which has an impact on the algorithm’s independence test. Processing features individually undoubtedly increase the algorithm’s running time, and this disadvantage becomes more apparent as more features are chosen. For example, Alpha-Investing [7] is a framework for processing large data sets where the runtime does not increase exponentially

Ze Liu, Dianlong You, Weiwei Pan, Junjie Zhao and Yefan Cao contributed equally to this work.

✉ Shunpan Liang
liangshunpan@ysu.edu.cn

Extended author information available on the last page of the article.

regardless of the number of features, but processing the features only once reduces the time overhead, but increases the redundancy among features, making the algorithm less accurate. At the same time, rough sets are a powerful tool for handling streaming features. To solve the drawback that rough sets cannot handle real-valued data, Hu et al. proposed a neighborhood rough set approach that allows the algorithm to support both continuous and discrete data [8]. For example, OFS-A3M [6] processes streaming features using NRS and proposes an adaptive neighborhood relationship that allows the algorithm to select the right number of features for different instances in the data set, reducing the influence of parameters on the algorithm. However, the above method does not consider the interaction between features, when two or more features are combined will show higher correlation with the labels. Therefore, some important features will be missing if feature interactions are not considered.

Due to the characteristics of group inspiration, evolutionary algorithms have been introduced to solve the feature selection, but most of them only consider a single objective and ignore multiple objectives. DAEA [9] is an EA algorithm based on repeated analysis and improves the basic EA framework in three aspects to obtain good classification and generalization results. To solve the problems encountered by the above methods, this paper uses the PSO evaluation technique to find the best quality feature combinations by a two-layer filtering method. PSO has good exploration ability in solving the supervised feature selection problem. PSO is used in the first layer of filtering to choose the most relevant feature combinations for the arriving groups in the multi-objective optimization function [10] with the objective function. We will use the neighborhood rough set evaluation method for features in the second layer of filtering to judge the features chosen in the first layer of filtering, and we will get the final result after processing all of the groups. There have been many algorithms in papers that combine PSO with rough set(RS), such as MMOFS-3S [11], which combines PSO, MI, and RS, but RS is not the core. Some other algorithms apply to an objective equation that may not be related to NRS or have only one objective function. While our algorithm based on NRS design two kinds of objective functions for optimization, which makes the quality of selected features better.

The contributions made in this paper are as follows: 1) A two-layer filtering architecture is designed by fusing PSO and NRS, which considers the interaction between features and removes redundant features to obtain the best set of features. 2) Two NRS-based objective functions are designed so that the algorithm is optimized according to multiple objectives during the iterative process, finally obtaining the set of features with the highest relevance to the label space. 3) The PSO-NRS algorithm was evaluated

on 14 datasets with 6 online streaming feature selection algorithms from 3 classifiers to verify the effectiveness and generalization of the algorithm. Meanwhile, the stability of the algorithms as well as the significance differences are verified using Spider web graph and Critical Difference (CD) pictures. 4) To verify the effectiveness of the combination of PSO and NRS, we designed ablation experiments and demonstrated the effectiveness by analyzing the experimental results.

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 introduces theoretical knowledge and symbolic definitions. Section 4 presents the algorithm. Section 5 shows experimental results and Section 6 concludes the paper.

2 Related work

In recent years, online streaming feature selection [12–14] has become popular as an important branch of feature selection, assuming that features come one by one with the flow of time and processing the incoming features in real time [15, 16]. Since there is no complete feature space, it becomes extremely difficult to select features with high efficiency.

- **Feature selection using neighborhood rough sets**

To resolve continuous data types, NRS are proposed to address the issue that classical rough sets [17–19] are inconvenient for dealing with data sets with numerical type attributes. the classical rough set must discretize the data but the discretization process alters the data's original attribute properties. The NRS [20, 21] can handle both continuous and discrete data. OSFS-ET [22] is a novel early terminated online streaming feature selection framework, which could terminate the streaming feature selection early before the end of streaming features and guarantee a competing performance with the currently selected features. However, the traditional rough set cannot handle real-valued features leading to relatively low usability of the algorithm. A new online streaming feature selection method based on adaptive density neighborhood relation (OFS-Density) [23] proposes a new adaptive neighborhood relation based on density information from surrounding instances that does not require any prior domain knowledge. However, the disadvantage of the feature processing dominance in the online selection process causes the algorithm to occasionally fail to select the best combination of features. An online multi-label streaming feature selection framework (OM-NRS) [24] customizes a criterion to select important features and designs a pairwise correlation constraint between

features to filter out the redundant features. Although the algorithm is capable of handling multi-label [25–27] data, however, feature interactions are not considered.

In conclusion, although using NRS has the advantage of not requiring domain knowledge, most algorithms do not consider the interactions between features. Therefore, this paper uses feature groups to explore the interactions between features and improve the efficiency of the algorithm.

• **Feature selection through Particle Swarm Optimization**

The PSO algorithm is inspired by birds foraging and finds the optimal particles through continuous particle swarm iteration. Using PSO to select features has a positive effect [10, 28, 29]. The algorithm works by first initializing N particles, each of which consists of a multidimensional position and velocity vectors, and then the particles update themselves by pursuing two extremes, one of which is the optimal solution reached by the particle itself, called $p_{(best)}$, and the other is the optimal solution found by the whole population, called $p_{(gest)}$, and the pbest and gbest are updated through continuous iterations until the maximum number of iterations is reached or the optimization function has converged.

A PSO based feature selection method using mutual information (PSO-FS-MI) [30] treats feature selection as a minimization problem and combines PSO with mutual information (MI) [4, 31, 32] using the idea of wrappers for feature selection. Although the wrapper-based approach is time-consuming, satisfactory optimal feature sets can be produced by good optimization techniques. A Multi-objective framework based, Multi-label learning, Online Feature Selection algorithm (MMOFS-2S) [11] divides feature selection into three steps and processes streaming features as a group in line with the idea of online feature selection. Combining PSO, MI, and rough set for feature selection via a three-level filtering framework demonstrates the algorithm’s effectiveness. The complexity of the optimization function, on the other hand, becomes a disadvantage of the algorithm. PEFS [33] models feature selection as a bi-objective optimization problem with feature relevance and redundancy, based on a filtering strategy and using a heterogeneous approach for integrated learning. The method is not able to process streaming features and needs to have the full feature space before performing feature selection. In conclusion, the PSO algorithm for feature selection is a reasonable solution; however, the computational consumption caused by multiple iterations becomes a major disadvantage of the algorithm, and how to design the optimization function reasonably becomes the key.

3 Theory preparation

NRS support both continuous and discrete data sets, which well increase the scalability of the algorithm. In this section, we review some basic concepts and notations of NRS. The symbols and definitions of this section are given in Table 1.

Definition 1 Given DS , $\Delta(x, y)$ satisfies the following properties [8]:

- (1). $\Delta(x, y) \geq 0$; $\Delta(x, y) = 0$; if and only if $x = y$
- (2). $\Delta(x, y) = \Delta(y, x)$
- (3). $\Delta(x, z) \leq \Delta(x, y) + \Delta(y, z)$

Definition 2 Given DS and A , the neighborhood x_i of the instance on A can be expressed as [8]:

$$\theta_A(x_i) = \{x_j \parallel x_j \in U, \Delta(x_i, x_j) \leq \theta\} \tag{1}$$

The size of the threshold θ determines the number of neighbors of the instance x_i . The value of θ is taken as 0.35 of the distance from the farthest instance of x_i , which will eventually lead to an adaptive neighborhood relation R .

Definition 3 Given that DS , $\theta_A(x_i)$, X_1, X_2, \dots, X_N are sets of instances divided according to whether the values of decision attributes are the same, the upper and lower

Table 1 Symbol definition

Symbols	Definition
U	Non-empty set of instances
C	Conditional attribute set
D	Decision attributes (labels)
$DS = \langle U, C, D \rangle$	Decision system
A	Subset of properties of C
CF	Candidate feature set
SF	The first layer of filtering selects the features
$\Delta(x, y)$	Distance between instance x and instance y
P	Position matrix
V	Velocity matrix
S	Selected features
NS	Unselected features
$\underline{N}_A D$	D with regard to the lower approximation of A
$\overline{N}_A D$	D with regard to the upper approximation of A
θ	Threshold
$\theta_A(x_i)$	The neighborhood of instance x on A
$\gamma_A(D)$	The dependence of A on D
$\sigma_A(D, f)$	The importance of f to D
R	Neighborhood relationship
NDS	Non-dominated sorting
CW	Crowding distance

approximation of D with respect to A is defined as follows [8]:

$$\underline{N}_A D = \bigcup_{i=1}^N \underline{N}_A X_i \tag{2}$$

$$\overline{N}_A D = \bigcup_{i=1}^N \overline{N}_A X_i \tag{3}$$

Among them

$$\underline{N}_A X = \{x_i \mid \theta_A(x_i) \subseteq X, x_i \in U\} \tag{4}$$

$$\overline{N}_A X = \{x_i \mid \theta_A(x_i) \cap X \neq \emptyset, x_i \in U\} \tag{5}$$

$\underline{N}_A D$ is the positive region of the decision, denoted by $POS_A(D)$.

Definition 4 In DS , the dependence of A on D is represented by $\gamma_A(D)$ in the following equation [8]:

$$\gamma_A(D) = \frac{\|pos(A)\|}{\|U\|} \tag{6}$$

We can observe that the range of $\gamma_A(D)$ is $[0,1]$. If $\gamma_A(D)$ is equal to 1, it means that the feature subset A depends entirely on D .

Definition 5 In DS , given A and D , the importance of feature f for D is defined as follows [8]:

$$\sigma_A(D, f) = \gamma_A(D) - \gamma_{A \setminus \{f\}}(D) \tag{7}$$

4 Proposal method

This paper presents a multi-objective optimization algorithm for online group feature selection, which combines a PSO algorithm and NRS to enable the algorithm to select a more effective combination of features. During the algorithm’s processing, the arrival of features in groups allows to

consider the interaction between features. To select a better combination of features, we divide the algorithm into two parts, which we call two-layer filtering. Intra-group feature selection uses a multi-objective PSO algorithm to find SF by filtering features in groups. The NRS online feature selection evaluates features based on the neighborhood rough set evaluation criterion, and redundancy updates are processed for features that meet the redundancy requirements. The feature selection algorithm based on two-layer filtering eventually generates a combination of features CF with high relevance and low redundancy to the labels, and the framework is shown in Fig. 1.

4.1 Problem definition

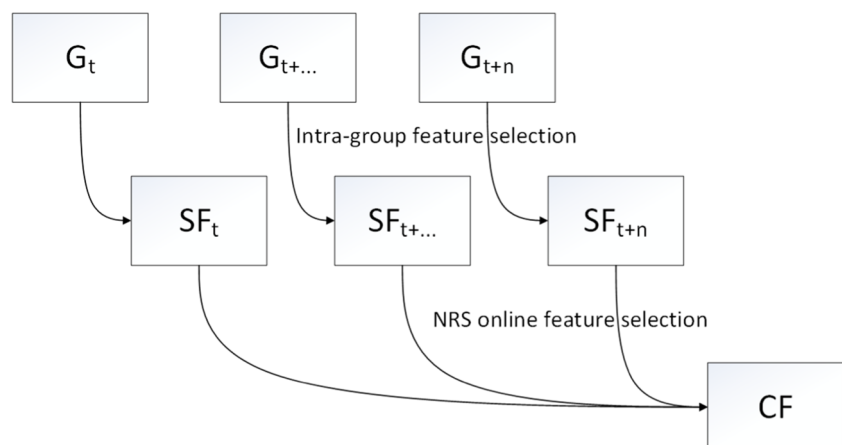
Given:

- A single-label data set, which can be represented using labels D and feature sets $C = \{f_1, f_2, \dots, f_n\}$.
- The features arrive at different periods and continuously flow into the group. Target:
- A subset of features M is selected from C . The number of M is much smaller than C .
- M can produce similar efficiency as the original C .

4.2 Intra-group feature selection

The PSO algorithm’s first filtering layer feature selection methods are classified as batch or online stream based on whether or not the entire feature space is available. We consider a group-based approach to stream feature selection to capitalize on the benefits of batch methods while taking the trend of streaming features into account. In the first layer of filtering, the features in the group G_t that arrive over time are randomly combined using the PSO algorithm, which eventually produces a combination of features that are highly correlated with the decision feature D through continuous iterations.

Fig. 1 Overview of PSO-NRS online feature selection



4.2.1 The particle

When initializing a particle swarm, a matrix can be used to represent the swarm and provide a clear view of the state of each particle. Each row in the matrix represents a particle, also known as a solution, which has a position and a velocity vector. The ultimate goal of the *PSO* algorithm is to discover the best solution. In (8), N particles are initialized and the dimensionality of the vectors in the matrix is the same as the number of features in each group. Every element p_j in the P of each particle every element is initially set to $[0,1]$ and the element value is expressed as the probability of selecting that feature. When the probability value is greater than 0.6 it means that this feature is selected, and the opposite is true. The V is initialized in the range $[-0.5,0.5]$ and is mainly used for the particle update operation.

$$\text{position : } \begin{pmatrix} p_{11} & \dots & p_{1W} \\ \vdots & \ddots & \vdots \\ p_{N1} & \dots & p_{NW} \end{pmatrix} \text{ velocity: } \begin{pmatrix} v_{11} & \dots & v_{1W} \\ \vdots & \ddots & \vdots \\ v_{N1} & \dots & v_{NW} \end{pmatrix} \tag{8}$$

Finding the optimal solution through continuous iterations of the particle swarm is the core of the *PSO* algorithm. In (9) and (10), the process of updating the position and velocity of particle j from the i th to the $i + 1$ th iteration is described. In each iteration, the particle moves from its position to a new position to approach the optimal position. The position and velocity matrix of the particle can be updated as:

$$\begin{aligned} vel_j^{i+1} = & w \times vel_j^i + c_1 \times r_1 \times (p_{(best)_j}^i - pos_j^i) \\ & + c_2 \times r_2 \times (g_{(best)_i}^i - pos_j^i) \end{aligned} \tag{9}$$

$$pos_j^{i+1} = vel_j^{i+1} + pos_j^i \tag{10}$$

where c_1 and c_2 are two constant values, $p_{(best)}$ is the personal best position of the particle and $g_{(best)}$ is the best position reached by the particle swarm, and r_1 and r_2 are random values in the range $[0, 1]$ to increase the randomness of the search. w is the Inertia weights, which represents the particle’s trust in the previous state of its own motion.

When w is large, the algorithm is good at global exploration; when w is small, the algorithm is good at local exploration. As a result, this algorithm employs a decreasing weight strategy, allowing it to find higher-quality solutions. The update process of w can be expressed as:

$$w = (w_{init} - w_{end}) \times (Iter_{init} - Iter) / Iter + w_{end} \tag{11}$$

The formula sets w as the current weight, w_{init} as the initialized weight at the beginning, w_{end} as the size of the weight at the final end, $Iter_{init}$ as the maximum number of iterations set, and $Iter$ as the current number of iterations.

4.2.2 Optimization functions

In general, the multi-objective EC-based (Evolutionary Computation) feature selection algorithms mainly use archiving and diversity enhancement techniques of Pareto dominance mechanism [34]. The *PSO* algorithm has the advantage of simplicity compared to other optimization algorithms, but the number of particles and multiple iterations makes the computational loss of the algorithm significantly higher, and the optimization function needs to be carefully designed to reduce the impact of computational effort. The NRS was chosen as one of the methods for calculating the dependency between features and labels because it requires no domain knowledge and no parameters to be set. The algorithm generates the objective function by taking into account the dependencies of S and NS in the group on the labels. In this case, using NRS to determine the dependence of labels on multiple variables significantly reduces the computational overhead. We can find the dependency of features by using the (6).

Algorithm 1 details the calculation of feature-to-label dependencies. The Euclidean distance between each instance and the other instances is calculated first, and the appropriate number of neighbors for each instance is returned using the corresponding neighborhood relation. To complete the dependency calculation, the Definition 3 is used to find the lower approximation of the corresponding feature subset and divide it by the number of instances.

In the initialized particle swarm probability matrix, we label the elements in each particle as S and NS by the magnitude of the probability. Dependency is obtained for each particle’s S and NS on the decision features, labeled $r_S(D), r_{NS}(D)$.

$$Obj_1 = r_S(D) - r_{NS}(D) \tag{12}$$

In (12), we want to make the S combinations extremely correlated with the decision features, while the NS combinations are less correlated, so it is not difficult to conclude that the larger the value of Obj_1 , the better.

If only Obj_1 is used as the optimization function, we find that the number of features in S increases with the number of iterations. To restrain the above trend and find the optimal combination of features more accurately, a second objective equation is proposed:

$$Obj_2 = r_{NS}(D) / r_S(D) \tag{13}$$

$r_S(D)$ and $r_{NS}(D)$ are the same as in Obj_1 , so the smaller the value of Obj_2 , the better. Optimization by multi-objective equation will make the feature combination after iteration produces good results.

Require: The target feature set F , the neighborhood relation R and the instance set U

Ensure: Dependence of F on labels D_F

- 1: $U_{(pos)}$: The number of positive samples in U , initialized to 0;
- 2: $\|U\|$: Number of instances in U ;
- 3: **for** x_i in U **do**
- 4: Calculate the distance between x_i and other instances by Euclidean distance;
- 5: Selecting neighbors of x_i based on R ;
- 6: $U_{(pos)}(x_i)$ represents the number of samples with the same attributes of (x_i) ;
- 7: $U_{(pos)} = U_{(pos)} + U_{(pos)}(x_i)$;
- 8: **end for**
- 9: $D_F = U_{(pos)} / \|U\|$;
- 10: **return** D_F ;

Algorithm 1 Calculate dependency.

4.2.3 Pareto optimal

Pareto optimality refers to an ideal state of resource allocation, indicating that there is no dominance between each other. The objective values of two solutions are constrained to each other in multi-objective optimization, making it difficult to choose the best solution. As a result, *NDS* [35] and *CW* [36] are applied to the algorithm for it to find the best solution. *NDS* takes the objective values of all particles as input and ranks the particles using a certain strategy. The final result is returned as a set of frontiers, with each frontier containing a collection of particles. At each frontier, no particle has an advantage over other particles. The upper frontier outnumbers the lower frontier. In the algorithm, each update to the position matrix produces a different target value, and we rank the particles non-dominantly and obtain frontier 1 as the set of candidate solutions. To select the most suitable from the many non-dominated solutions, we use the crowding distance as a measure and select the one with the lowest density between particles as the optimal solution generated by this iteration.

4.2.4 The specific process of the first phase

We assume that features arrive as streams and the general online streaming feature selection algorithm processes the arriving features in real time. The arriving features are summarized into groups until the number of features satisfies the group size or there are no more features.

The concept of combining the *PSO* algorithm in the Algorithm 2. Initialize N particles and each particle is given a P and V with dimensions equal to the group's size.

Following the determination of the target value for each particle, the N particles are subjected to *NDS* and *CW* calculations, and the best particle is chosen for comparison with $g_{(best)}$. During each iteration, if a more suitable solution appears, $g_{(best)}$ and $p_{(best)}$ need to be updated until the maximum number of iterations or the optimization function has converged. In the end, we get a set of features that are more combinable and the random combination of features breaks the drawbacks of traditional streaming features.

Require: The position matrix P and velocity matrix V of the particle swarm

Ensure: The first layer of filtering selects the features SF

- 1: I_{ters} : Number of iterations of the particle;
- 2: Op : Optimal particles in each iteration;
- 3: Initialize the P and V ;
- 4: **for** I_{ter} in I_{ters} **do**
- 5: The features are divided into S and NS based on the values of the initialized P ;
- 6: Use the *Algorithm 1* to calculate $\gamma_S(D)$ and $\gamma_{NS}(D)$;
- 7: Find the target value according to the (12) and (13);
- 8: Op is obtained using *NDS* and *CW*;
- 9: **if** Op is superior to $g_{(best)}$ **then**
- 10: $g_{(best)} = Op$;
- 11: Update V and P using (9) and (10);
- 12: **end if**
- 13: **if** The updated particles appear locally optimal(Lo) **then**
- 14: $p_{(best)} = Lo$;
- 15: **end if**
- 16: **end for**
- 17: $SF = g_{(best)}$;
- 18: **return** SF ;

Algorithm 2 Intra-group feature selection.

4.3 NRS online feature selection

The feature set SF has a high dependency on the label, but there is no redundant update operation among the features in SF , which can easily make some very poor features added to CF . As a result, we will introduce the feature evaluation method based on the rough set, which improves the efficiency the selected features.

4.3.1 The maximum dependency

In Algorithm 3, to select the optimal feature set, the features in SF need to be judged one by one. If the feature f_i is added to CF making the overall dependency increase, f_i is

added to CF , and if it does not increase or even decrease, f_i needs to be eliminated. If the dependency remains the same as before, it means that f_i has a redundant relationship with the features in CF and must be processed for redundancy updates. This evaluation method is theoretically the most effective in the rough set, but the slow computation speed and lack of sufficient experimental samples make producing equivalent result classes in the high-dimensional space difficult.

Require: The candidate feature set CF and the features filtered out in the first stage SF

Ensure: The candidate feature set CF

```

1:  $D_F$ : Feature dependency calculated using Algorithm 1;
2:  $\sigma(f)$ : Importance of features  $f$  calculated using (7);
3: for  $f_i$  in  $FS$  do
4:    $CF = CF \cup f_i$ ;
5:   if  $D_F(CF)$  increases then
6:     continue;
7:   end if
8:   if  $D_F(CF) = D_F(CF - f_i)$  then
9:     for  $f_j$  in  $CF$  do
10:      if  $\sigma(f_j) = 0$  then
11:         $CF = CF - f_j$ ;
12:      end if
13:    end for
14:   else
15:      $CF = CF - f_j$ ;
16:   end if
17: end for
18: return  $CF$ ;

```

Algorithm 3 NRS online feature selection.

4.3.2 Redundant updates

In a set of feature sets, there are often redundant features that do not serve any purpose and, on the contrary, sometimes reduce the efficiency of the algorithm. For instance, if one of the two features f_i and f_j is useful, the other will have effect. If the redundant features are not removed, the final feature selection process becomes lengthy and inefficient. When the newly arrived feature reduces the overall dependency to the previous dependency, it is added to the CF and the important test is run on each feature in the CF . If the significance of the feature is equal to 0, the feature is redundant.

4.4 General overview of algorithm

The proposed algorithm considers features arriving continuously over time and saved into a group. Groups arrive in an

online fashion. Intra-group feature selection and NRS online feature filtering are used to filter the features in each group. $PSO - NRS$ employs the PSO technique as the foundation for feature selection, combining the features in the group at random to find the most relevant feature set SF . In the second stage, the features in SF are filtered again using the feature evaluation method of NRS. The overall $PSO - NRS$ algorithm can be referred to Fig. 2.

The Intra-group feature selection:

- N particles are initialized at the beginning, each particle is represented by a multidimensional position and velocity vector and the dimension of the vector is equal to the group size. The position vector is initialized to some random values of probability for each dimensional element, and the probability values are used to obtain S and NS . The target value of each particle is derived from the target equation as a criterion for evaluating the merit of the particle.
- P and V need to be updated according to the individual optimal $p_{(best)}$ and global optimal $g_{(best)}$, and the updating process can be seen in Section 4.2.1.
- When the maximum number of iterations is satisfied or the function has converged to the state indicating that we have selected the optimal particle $g_{(best)}$ in the group. the is obtained in the $g_{(best)}$ for the second stage of filtering.

The NRS online feature filtering:

- This process involves adding features f_i to the CF in the SF until there are no more features in the SF . The feature f_i needs to be processed in real time, after a maximum dependency and redundancy update strategy to decide whether the feature is needed or not, as described in Section 4.3.
- By filtering each set of features at two levels, we end up with a set of features that are highly relevant to the label and have low redundancy.

4.5 Time complexity

From the principle, we can know that the time complexity is mainly concentrated in the calculation of the dependency on the label and the iteration of PSO. For the arrival of a set of features, the time complexity of the computation of the target equation in the first layer of filtering is $O(MNI^2 \log I)$, where M is the number of target equations, N is the number of particles, and I is the number of instances. The time complexity of Algorithm 1 is $O(N^2 \log N)$. The calculation of the non-dominated ranking as well as the congestion distance can be expressed as $O(N^2 \log N)$, so the total time complexity of the first layer of filtering is $O(MNI^2 \log I + MN^2 \log N)$, which

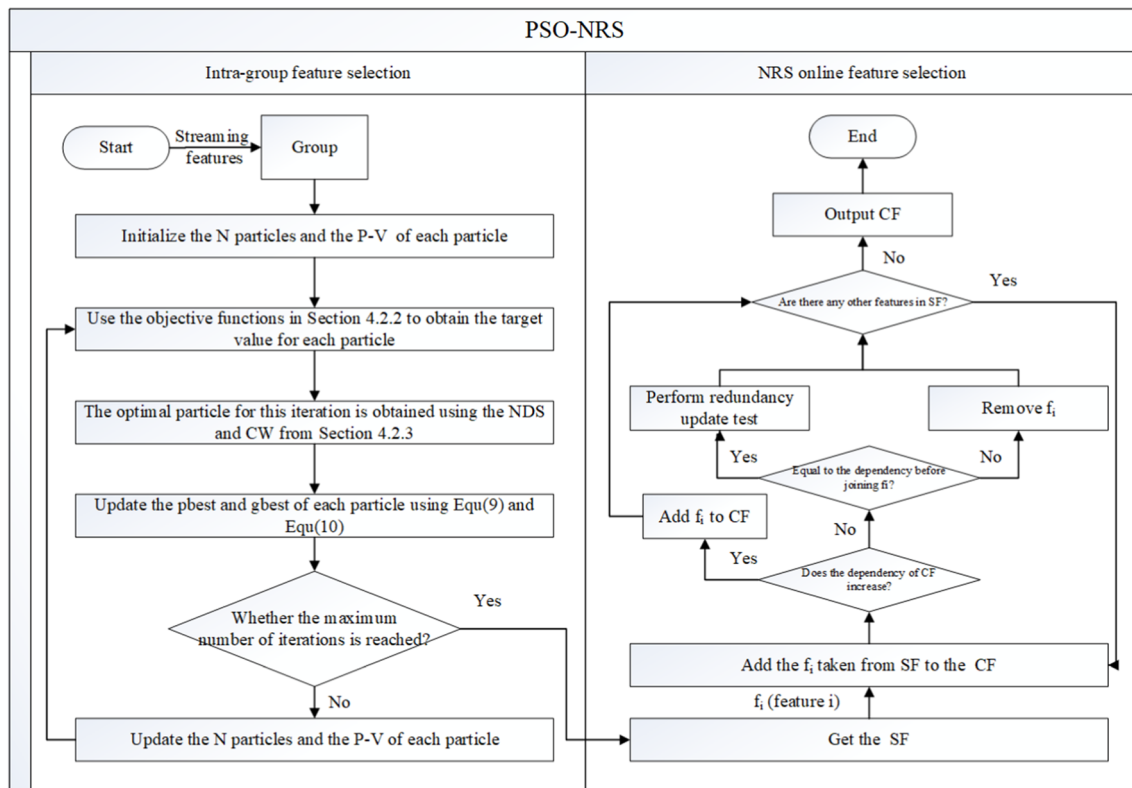


Fig. 2 Overview of online feature selection PSO-NRS

can be reduced to $O(MNI^2 \log I)$. The second layer of filtering is primarily for feature evaluation and redundancy update, so in the worst case a redundancy update operation for each feature may be required, with time complexity of $O(SCI \log I)$, where S is the total number of features and C is the candidate feature set of total number of features. The total time complexity is $O(MNI^2 \log I + SCI \log I)$, which can be summarized as $O(MNI^2 \log I)$.

5 Experiment

5.1 Data sets

In the experiment, we use three classifiers to evaluate the selected feature set in MATLAB R2018b. The whole algorithm is written by python language. All experimental results are generated in a computer with Intel(R) i7-8750H CPU 8G memory.

To validate the algorithm, 14 benchmark data sets from different domains were found for validation (overview of the data sets) The selected data sets have instances between 62 and 2600. The number of features ranged from 500 to 12582 and the range of categories was between 2 and 40. It

can be seen that the selected data sets cover a wide range of types from small to large samples and from low to high-dimensional. It is possible to verify the effectiveness of the algorithm. The details of each data set, such as the number of instances, the number of features, and the number of categories, are listed in Table 2.

Table 2 Details of the data sets

Data set	Instance	Feature	Class
ALL-AML	72	7129	2
ALL-AML-3	72	7129	3
ALL-AML-4	72	7129	4
LYMPHOMA	62	4026	3
MLL	72	12582	3
ORL	400	1024	40
ORLRAWS10P	100	10304	10
SRBCT	83	2308	4
WARPPIE10P	210	2420	10
YALE	165	1024	15
LUNG	203	3312	5
COIL20	1440	1024	20
ISOLET	1560	617	26
MADLON	2600	500	2

Table 3 Prediction accuracy using KNN classifier

Data sets	OSFS	Fast-OSFS	SAOLA	Alpha-Investing	OFS-A3M	OFS-Density	PSO-NRS	SD
ALL-AML	86.09	93.14	49.04	84.57	85.90	84.57	97.23	3.39
ALL-AML-3	81.61	85.90	48.09	45.24	77.24	81.52	92.95	7.83
ALL-AML-4	68.76	77.33	41.05	56.57	82.86	77.43	90.10	7.32
LYMPHOMA	36.53	51.05	93.74	23	75.11	23	88.63	8.19
MLL	76.1	78.86	39.52	66.29	79.05	86	95.81	3.43
ORL	23	40.5	6.25	65.75	80.5	72.75	88.75	2.85
ORLRAWS10P	41	61	43	79	91	58	94	2
SRBCT	79.63	84.26	96.32	79.49	89.12	84.19	97.65	4.71
WARPPIE10P	67.62	80	31.43	87.62	94.29	21.43	94.76	3.81
YALE	13.33	15.76	13.33	30.91	46.06	27.27	58.18	6.47
LUNG	74.37	81.28	92.61	86.18	88.13	74.91	95.06	1.6
COIL20	64.44	81.94	32.71	99.79	98.68	11.39	99.24	0.77
ISOLET	35.58	37.56	71.6	85.19	78.27	74.81	91.09	1.59
MADLON	55.69	55.96	55.65	65.42	69.54	49.92	77.75	11.99

5.2 Experiment to verify the effectiveness of the algorithm

5.2.1 Analysis of experimental data

We present the experimental results for the best performing algorithms in the table with bold and italicized. Tables 3, 4 and 5 lists the accuracy of each algorithm on different classifiers. There are three classifiers used in the experiments K-nearest neighbor (*KNN*), random forest (*RF*) and discriminant analysis (*DA*). Since the initial *P* and *V* of the algorithm are randomly generated, different experimental results are generated at the end. To get rid of the randomness of the algorithm in the data set, we perform a fivefold cross-validation on the data set and use the average of the results

as the final experimental results. There are various ideas for the implementation of classifiers. For example, For example, *KNN* compares the unlabeled features with the labeled features and finally selects *K* labels with the most occurrences. The random forest classifier embodies the idea of integration by having many decision trees, each of which is a classifier. It is the diversity of classifier algorithms that makes the final output different.

We compare with several state-of-the-art algorithms, such as SAOLA [12], OFS-A3M [6], OFS-Density [23], etc. As can be seen in Table 3, the algorithm performs quite well in the *KNN* classifier, with the accuracy better than the other algorithms in 12 of the 14 data sets and above 90% in 10 of the data sets. In the *RF* classifier, our algorithm achieves the optimum in ten data sets, while

Table 4 Prediction accuracy using RF classifier

Data sets	OSFS	Fast-OSFS	SAOLA	Alpha-Investing	OFS-A3M	OFS-Density	PSO-NRS	SD
ALL-AML	85.98	90.28	46.19	77.71	87.43	87.33	93.05	4.22
ALL-AML-3	85.9	84.38	48.16	61.71	85.81	91.52	87.33	9.45
ALL-AML-4	68.76	73.01	38.1	61.9	80.19	78.76	80.48	12.54
LYMPHOMA	50.11	60.47	83.52	44.84	77.21	44.84	78.21	8.19
MLL	83.74	85.9	36.67	81.62	86	91.62	92.86	3.43
ORL	21.76	44.5	6.25	71.75	83.75	78.25	88.25	4.51
ORLRAWS10P	49	49	43	77	91	59	91	7.35
SRBCT	80.74	85.44	98.82	97.57	89.12	82.94	96.4	2.94
WARPPIE10P	65.71	77.14	31.43	89.05	91.43	18.58	92.38	5.08
YALE	10.3	16.97	12.12	51.52	60.61	32.73	65.45	10.94
LUNG	81.28	87.68	93.11	88.13	90.65	84.27	90.61	3.37
COIL20	70.63	85.42	33.75	99.65	98.33	9.93	99.167	0.35
ISOLET	43.01	48.85	81.99	89.62	81.41	78.59	90.83	1.63
MADLON	59.15	61.38	51	68.69	73.62	50.46	78.46	9.58

Table 5 Prediction accuracy using DA classifier

Data sets	OSFS	Fast-OSFS	SAOLA	Alpha-Investing	OFS-A3M	OFS-Density	PSO-NRS	SD
ALL-AML	83.24	91.71	59.05	80.29	84.48	85.81	86.1	7.14
ALL-AML-3	84.57	81.62	61.81	61.81	83.05	88.76	81.62	5.87
ALL-AML-4	73.39	73.39	52.29	57.71	81.52	78.57	81.62	5.87
LYMPHOMA	50.11	58.47	91.74	47	75.11	47	81.32	9.58
MLL	80.38	88.76	39.52	74.44	80.57	87.43	84.57	9.32
ORL	16.5	35	2.75	68.75	83.25	69	91	3.1
ORLRAWS10P	15	49	17	62	77	22	84	8.6
SRBCT	67.43	87.79	97.57	79.34	90.29	74.63	91.47	7.21
WARPPIE10P	46.19	66.67	22.38	95.71	88.8	10.48	84.76	9.36
YALE	10.3	12.73	8.48	29.7	50.3	11.52	52.12	7.99
LUNG	83.27	86.7	93.11	90.62	91.61	83.28	96.04	2.55
COIL20	51.74	59.51	18.19	96.53	82.64	9.51	83.75	2.62
ISOLET	41.67	44.49	79.17	91.41	78.46	75.43	91.47	1.03
MADLON	59.73	59.54	55.42	59.54	60.58	47.69	59.62	1.56

differing little from the optimum in the remaining data sets. Six data sets achieved the best results in the *DA*. Although slightly inferior to the previous two classifiers, the algorithm's efficiency is relatively stable and it does not perform well on one data set but performs poorly on another. By comparing different algorithms, it can be found that there is a big difference in the performance of the algorithms in different domain data sets. In our experiments, we found that the algorithm selected only one feature in a certain data set resulting in a low accuracy rate. There are various reasons for this, it may be due to the data set or the features flowing over too well that other important features cannot be selected. PSO-NRS uses an online group feature selection method and combines features within groups randomly to discover a high-quality feature set. Meanwhile, the application of NRS can adapt to various types of datasets to enhance the generalization ability of the algorithm. Therefore, PSO-NRS can effectively overcome the above-mentioned drawbacks.

To measure the dispersion of the accuracy obtained from the five-fold cross-validation, the standard deviation (SD) was calculated for the five experimental results. Although it is the same data set, the data distribution has been changed significantly by continuously transforming the training and testing sets. From the SD results in Tables 3–5, they are basically scattered in the range of 0–12, with most of them in the middle of the range, indicating that the experimental results are not too scattered, which also illustrates the necessity and effectiveness of the five-fold cross-validation.

To prove the effectiveness of the algorithm, we compared PSO-NRS with the latest NRSIPSO algorithm [37], which incorporates PSO and NRS. Due to the unavailability of code, the experimental results of this article are directly

quoted in this paper. The ten-fold validation used by NRSIPSO to get the final results, and the five-fold cross-validation used by PSO-NRS, it has been stated that the results of ten-fold cross-validation and five-fold cross-validation are not very different. The results are shown in Table 6. It can be seen that there is not much difference between the two algorithms in terms of accuracy. In the ISOLET dataset, 91 features were selected in the PSO-NRS algorithm, while 152 features were selected in the NRSIPSO, indicating that our algorithm is more efficient in this dataset. In the PROState dataset, PSO-NRS selects 19 features and NRSIPSO selects 4 features, which is slightly inferior to the PSO-NRS algorithm in this dataset. But in general, the PSO-NRS algorithm does not differ much from the recent algorithms in terms of efficiency.

5.2.2 Verify the stability of the algorithm

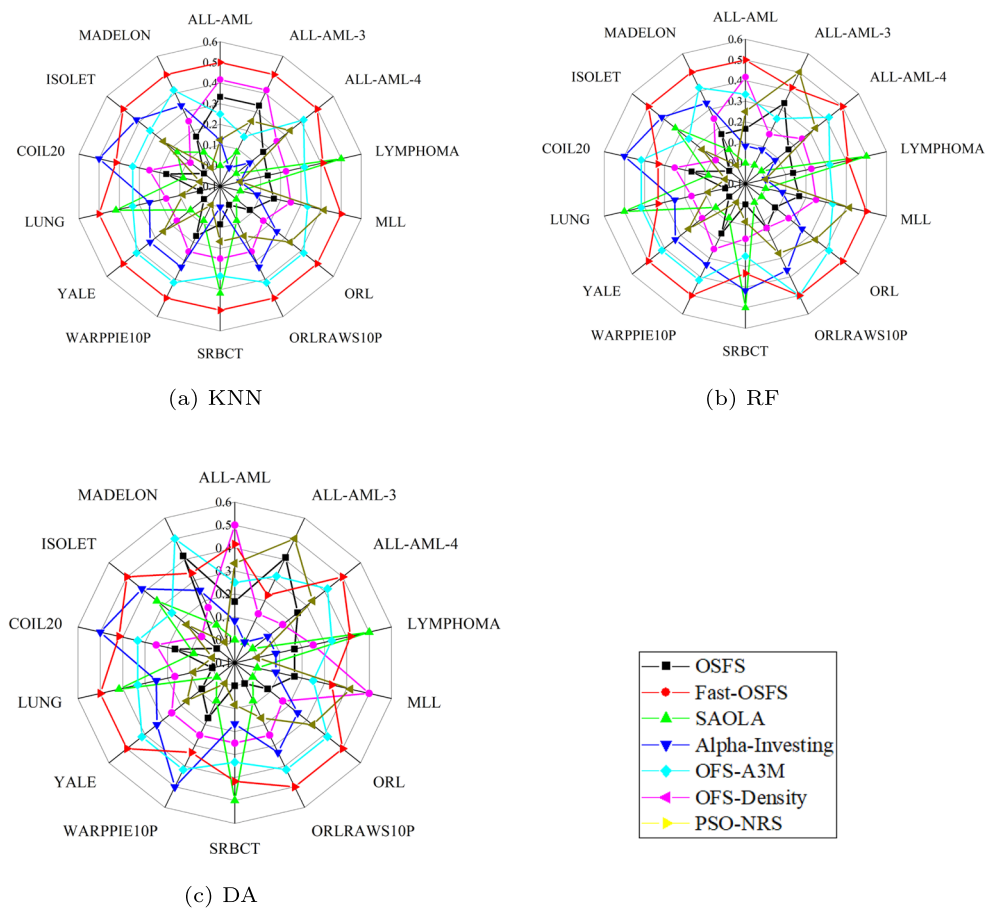
To present the experimental results more clearly, we ranked the algorithms according to their efficiency on each data set and normalized the ranked results (the normalized range is [0, 0.5]).

It can be observed from the Fig. 3 that our algorithm is always at the edge of the graph (the algorithm in red in the figure), which indicates that the algorithm is not only stable but also very efficient. Although the graphs enclosed in the *DA* classification are not very rounded, the efficiency

Table 6 Results of comparison PSO-NRS with NRSIPSO

Data sets	NRSIPSO	PSO-NRS
ISOLET	89.87	91.09
PROState	89.63	87.24

Fig. 3 Examine the stability between different online feature selection algorithms



of the algorithm is very close to the optimal value in other data sets where the optimal efficiency is not obtained, and the algorithm is consistently in the top three in the ranking. On the contrary, the graphs enclosed by the lines of other algorithms can be represented by irregular graphs, which can get high efficiency in some data sets but show inferiority in other data sets, reflecting an unstable effect.

5.2.3 A significantly different test between algorithms

We use the Friedman test and Bonferroni-Dunn test for statistical analysis of all online feature selection algorithms to compare the performance between algorithms [38].

The Friedman test can be defined as:

$$F_F = \frac{(N - 1)\chi_F^2}{N(k - 1) - \chi_F^2} \tag{14}$$

Equation (14) follows an F distribution, among them, $\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right]$, $R_j = \frac{1}{N} \sum_{i=1}^N r_i^j$, r_i^j denotes the ranking of the i -th algorithm on the j -th data set. We can obtain the degrees of freedom as $K - 1$ and $(K - 1)(N - 1)$. When F_F is higher than $F_{\{K-1\}\{(K-1)(N-1)\}}$, it means that the null hypothesis is rejected. From Table 7, we

can see that the null hypothesis is rejected for all evaluation metrics at α equal to 0.05, which indicates a significant difference between the algorithms. For the follow-up test, we chose the Bonferroni-Dunn test and determined the critical difference (CD). CD can be denoted as:

$$CD = q_\alpha \sqrt{\frac{k(k + 1)}{6N}} \tag{15}$$

When $K = 7$ and $N = 14$, CD was 2.153 at α equal to 0.05 because $q_\alpha = 2.638$.

To visualize the significant differences between the algorithms, we plotted the CD graph using the computed CD and F_F data. The algorithm on the far right is defined as the best algorithm in Fig. 4 which has a numerical axis

Table 7 Friedman test for classifiers

Classifier	F_F	$F_{(6,78)}$
KNN	12.995	–
RF	10.175	2.22
DA	6.697	–

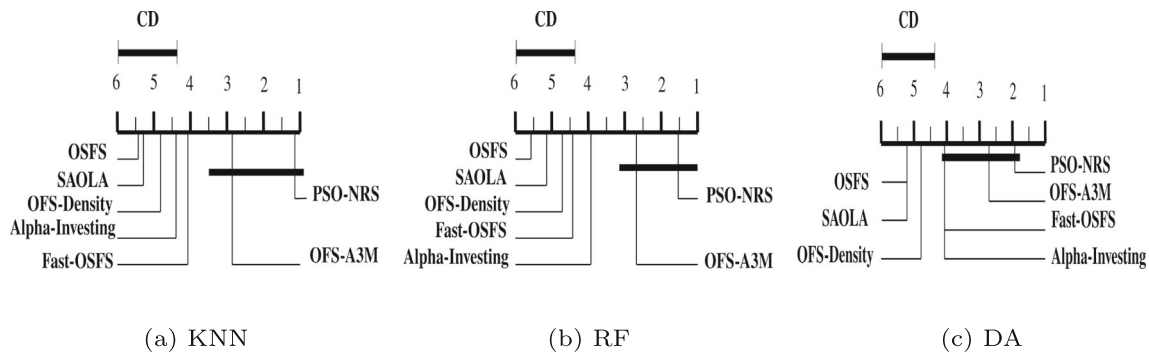


Fig. 4 Statistical analysis of different algorithms using CD diagram

from largest to smallest. In the *KNN* and *RF* classifiers, PSO-NRS and OFS-A3M are connected by a black line on the same side and at the right end of the other algorithms, indicating that there is no significant difference between the two algorithms and the effect is similar, whereas the other algorithms have a significant difference, indicating that there is a significant difference in the effect between the algorithms. In the *DA* classifier, there are four algorithms at the right end of the number axis, but the PSO-NRS algorithm is still at the rightmost end of the number axis, although there is no significant difference between the four algorithms, but our algorithm is still optimal.

5.3 Influence of parameters on the PSO-NRS

The use of the PSO algorithm leads to an increase in the number of algorithm parameters, and selecting the most beneficial parameters for the algorithm has a great improvement in the performance of the algorithm. To save space, we averaged the accuracies obtained in the three classifiers to complete the comparison of the experiments. Parameters such as group size, number of particles generated, and number of iterations were explored in four data sets.

5.3.1 The effect of group size on the PSO-NRS

The group size represents the number of features that need to be processed for the first filtering, and a higher number of features indicate a stronger combinability between features. Table 8 shows that the optimum is obtained on two datasets with a group size of 200, and one data set with a group size of 50 and 100, respectively. Figure 5(a) shows an analysis of the number of features selected by the algorithm and the running time, demonstrating that the running time decreases with increasing group size. The number of features selected tends to average out without much deviation. Therefore, to take into account the running efficiency of the algorithm and

the real application scenario of the streaming features, 100 is finally chosen as the group size.

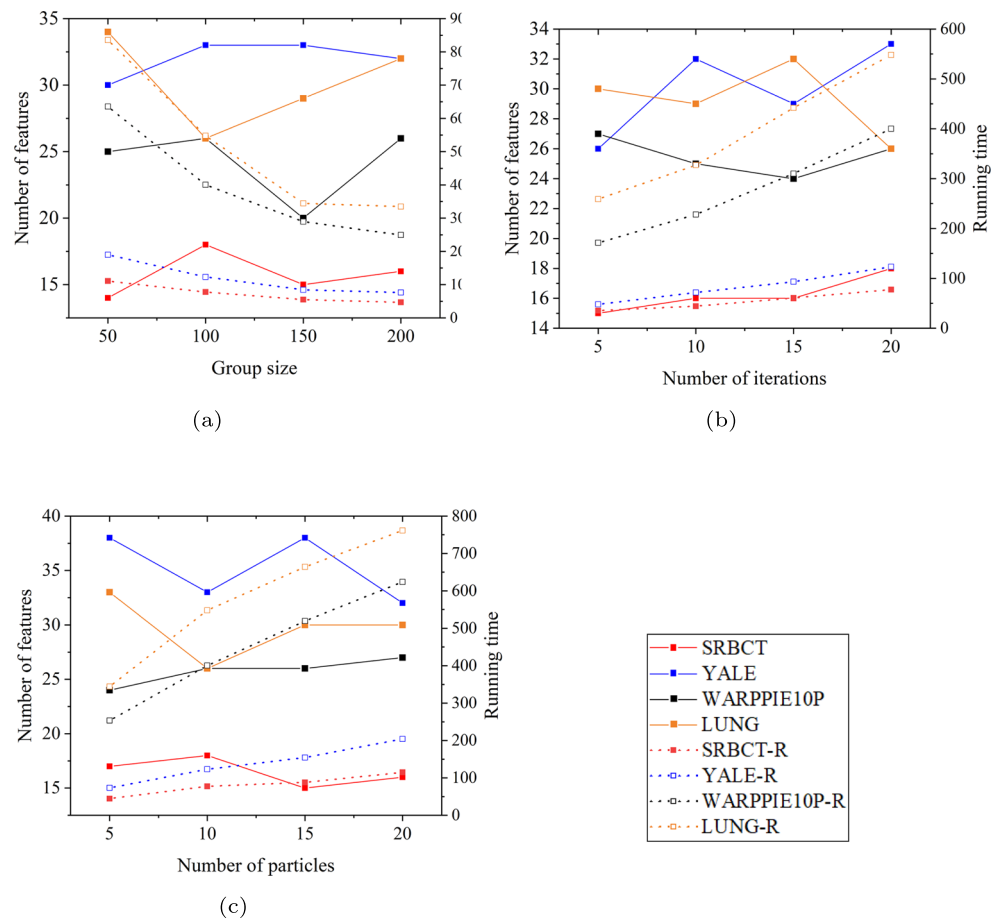
5.3.2 The effect of the number of particles on the PSO-NRS

Each particle represents a solution in the *PSO* algorithm, using the symbol *NOP* to denote the number of particles. Table 8 shows that the algorithm achieves the best accuracy in all three data sets when *NOP* is 15, and achieves the best in one data set when *NOP* is 10, implying that the number of particles can be considered between 10 and 15. In Fig. 5(b) it can be seen that the running time is increasing with the increase of particles, and in terms of the number of features selected, *NOP* equals 10 is relatively less in four

Table 8 Influence of parameters on the PSO-NRS

Size of group				
Data sets	50	100	150	200
SRBCT	97.06	95.17	91.86	94.36
YALE	59.8	58.59	61.21	61.62
WARPPIE10P	90	90.63	91.43	93.17
LUNG	94.72	95.17	93.91	90.46
Number of particals				
	5	10	15	20
SRBCT	90.74	95.17	96.69	92.75
YALE	59.6	58.59	62.02	60.81
WARPPIE10P	90.48	90.63	93.65	91.27
LUNG	93.52	95.17	93.89	91.59
Number of iterations				
	5	10	15	20
SRBCT	94.41	88.7	92.57	95.17
YALE	58.59	58.59	57.58	58.59
WARPPIE10P	92.38	93.05	90.48	90.63
LUNG	91.93	92.61	91.33	95.17

Fig. 5 Examine the stability between different online feature selection algorithms



data sets, so from various considerations, we finally choose *NOP* equals to 10 as the final result.

5.3.3 Effect of the number of iterations on the PSO-NRS

In the PSO algorithm, the idea that each particle approaches the optimal value through continuous iterations ensures that the algorithm can achieve a good result. Table 8 shows that the algorithm has three data sets that achieve the best results after 20 iterations. The running time of

the algorithm significantly increases with the number of iterations. as shown in Fig. 5(c). To make the accuracy of the algorithm increase, we choose to trade the running time of the algorithm for the steady state.

5.4 Analysis of PSO-NRS for ablation experiments and running time

In this section, we design ablation experiments to verify the effectiveness of the algorithm to combine PSO and NRS.

Table 9 Ablation experiments

Classifier	LUNG		SRBCT		WARPPIE10P	
	PSO-NRS	NRS	PSO-NRS	NRS	PSO-NRS	NRS
KNN	95.06	97.06	97.65	96.25	94.76	97.62
RF	90.61	94.07	96.4	87.72	92.38	90.48
DA	96.04	95.59	91.47	88.97	84.76	74.76
Classifier	MLL		LYMPHOMA		ISOLET	
	PSO-NRS	NRS	PSO-NRS	NRS	PSO-NRS	NRS
KNN	95.81	86	88.63	91.74	91.09	91.28
RF	92.86	87.33	78.21	70.95	90.83	91.99
DA	84.57	76	81.32	64.63	91.47	92.31

Table 10 Number of features selected by the algorithm in the 6 data sets

	LUNG	SRBCT	WARPPIE10P	MLL	LYMPHOMA	ISOLET
NRS	41	17	25	21	36	203
PSO-NRS	24	18	25	22	31	92

The first layer of filtering is removed from the original algorithm and only the second layer of filtering (NRS) is used for online streaming feature selection. The NRS is applied to 6 different types of datasets and the accuracy of the algorithm is analyzed in 3 different classifiers. Table 9 shows the accuracy of NRS in the KNN, RF, and DA classifiers.

From Table 9, it can be seen that PSO-NRS has higher accuracy than NRS in all three classifiers in the SRBCT and MLL datasets. In both WARPPIE10P and LYMPHOMA datasets, PSO-NRS has higher accuracy than NRS in two classifiers. In the LUNG and ISOLET datasets, NRS has better accuracy than PSO-NRS in the classifiers. To better verify the pros and cons of the algorithm, we compare the number of feature sets obtained by the NRS and PSO-NRS. As can be seen from Table 10, in the LUNG and ISOLET datasets, the number of features selected by NRS is significantly higher than PSO-NRS, but the accuracy are not significantly different. In the remaining datasets the number of features selected by the two algorithms is comparable, however, the accuracy of PSO-NRS is significantly better than that of NRS. Therefore, this experiment demonstrates the effectiveness of combining PSO with NRS.

We compare PSO-NRS with the OFS-A3M algorithm in terms of running time, which also uses NRS for online feature selection. Both algorithms are written using the python language. To reduce the length of the paper, we choose six datasets to compare in terms of running time. The specific experimental results are shown in Table 11. From Table 11, it is clear that OFS-A3M is significantly better than PSO-NRS in running time, which is the inevitable result of the reference to PSO in the algorithm, and the constant iteration makes the computational consumption of the algorithm significantly higher, which becomes a drawback of the algorithm.

6 Conclusion

In this paper, we propose a novel online feature selection algorithm called PSO-NRS. Use online group feature selection to increase the available features and explore the relationship between features to address the lack of available information for online stream feature selection that causes the selected feature set to be less efficient. A double-objective optimization function is designed to find the set of features that are highly correlated with the labels. The use of NRS does not require domain knowledge enhances the generalization ability of the algorithm. A feature search strategy is intended to select a set of features that are highly relevant to the labels and have low redundancy. In the experiments, we show that PSO-NRS is very competitive when evaluating the algorithm's accuracy among several classifiers and comparing it to some state-of-the-art algorithms; the algorithm's stability is examined and PSO-NRS is very stable state when compared to other algorithms; statistical tests are performed on the algorithm, and there is a significant difference between PSO-NRS and other algorithms, demonstrating the algorithm's effectiveness.

There are many scenarios in the field of feature selection that correspond to various possible situations encountered in the study, such as multi-label, semi-supervised, and unsupervised feature selection. In the following research, the above directions can be explored in depth to achieve the purpose of solving practical problems. PSO requires several iterations to find the optimal value therefore, which increase a high computational overhead. In future studies of PSO, the negative impact of iteration can be reduced by finding suitable strategies.

Table 11 Comparison with OFS-A3M algorithm in terms of runtime (Unit: second)

	ORL	ORLRAWS10P	SRBCT	YALE	LUNG	ISOLET
PSO-NRS	1199.87	471.65	77.59	123.19	543.15	8192.44
OFS-A3M	133.37	124.52	27.46	30.8	253.82	1678

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant No.51975505 and HeBei Natural Science Foundation under Grant No.G2021203010 & No.F2021203038. Meanwhile, it was supported by Key Laboratory of Robotics and Intelligent Equipment of Guangdong Regular Institutions of Higher Education Grant No.2017KSYS009.

References

- Agrawal P, Abutarboush HF, Ganesh T, Mohamed AW (2021) Metaheuristic algorithms on feature selection: a survey of one decade of research (2009-2019). *IEEE Access* 9:26766–26791. <https://doi.org/10.1109/ACCESS.2021.3056407>
- Bommert A, Welchowski T, Schmid M, Rahnenführer J (2022) Benchmark of filter methods for feature selection in high-dimensional gene expression survival data. *Brief Bioinform* 23(1):354. <https://doi.org/10.1093/bib/bbab354>
- Omuya EO, Okeyo GO, Kimwele MW (2021) Feature selection for classification using principal component analysis and information gain. *Expert Syst Appl* 174:114765. <https://doi.org/10.1016/j.eswa.2021.114765>
- Rahmaninia M, Moradi P (2017) Ofsfmi: online stream feature selection method based on mutual information. *Appl Soft Comput* 1568494617305161. <https://doi.org/10.1016/j.asoc.2017.08.034>
- Wu X, Yu K, Ding W, Wang H, Zhu X (2013) Online feature selection with streaming features. *IEEE Trans Pattern Anal Mach Intell* 35(5):1178–1192. <https://doi.org/10.1109/TPAMI.2012.197>
- Peng Z, Hu X, Li P, Wu X (2018) Online streaming feature selection using adapted neighborhood rough set. *Inf Sci* 481. <https://doi.org/10.1016/j.ins.2018.12.074>
- Aharoni E, Rosset S (2015) Generalized alpha investing: definitions, optimality results, and application to public databases. *J R Stat Soc* 76(4):771–794. <https://doi.org/10.1111/rssb.12048>
- Qing-Hua HU, Da-Ren YU, Xie ZX (2008) Numerical attribute reduction based on neighborhood granulation and rough approximation. *J Softw* <https://doi.org/10.3724/SP.J.1001.2008.00640>
- A duplication analysis-based evolutionary algorithm for biojective feature selection. *IEEE Trans Evol Comput* (2021). <https://doi.org/10.1109/TEVC.2020.3016049>
- Song X-F, Zhang Y, Gong D-W, Gao X-Z (2021) A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2021.3061152>
- Paul D, Jain A, Saha S, Mathew J (2021) Multi-objective pso based online feature selection for multi-label classification. *Knowl-Based Syst* 222(1):106966. <https://doi.org/10.1016/j.knosys.2021.106966>
- Kui YU, Xindong WU, Ding W, Pei J (2017) Scalable and accurate online feature selection for big data. *ACM Trans Knowl Discov Data* 11(2):16–11639. <https://doi.org/10.1145/2976744>
- You D, Wu X, Shen L, Deng S, Chen Z, Ma C, Lian Q (2019) Online feature selection for streaming features using self-adaption sliding-window sampling. *IEEE Access* 1–1. <https://doi.org/10.1109/ACCESS.2019.2894121>
- Bensaid F, Alimi AM (2020) Online feature selection system for big data classification based on multi-objective automated negotiation. *Pattern Recognit* 110(1):107629. <https://doi.org/10.1016/j.patcog.2020.107629>
- Lin Y, Hu Q, Liu J, Li J, Wu X (2017) Streaming feature selection for multi-label learning based on fuzzy mutual information. *IEEE Trans Fuzzy Syst* PP(99):1–1. <https://doi.org/10.1109/TFUZZ.2017.2735947>
- (2018) Online multi-label group feature selection. *Knowl-Based Syst* 143:42–57. <https://doi.org/10.1016/j.knosys.2017.12.008>
- Li Y, Lin Y, Liu J, Weng W, Shi Z, Wu S (2018) Feature selection for multi-label learning based on kernelized fuzzy rough sets. *Neurocomputing* 318:271–286. <https://doi.org/10.1016/j.neucom.2018.08.065>
- Bania RK, Halder A (2021) R-hefs: rough set based heterogeneous ensemble feature selection method for medical data classification. *Artif Intell Med* 114:102049. <https://doi.org/10.1016/j.artmed.2021.102049>
- Mohtashami M, Eftekhari M (2018) Using a novel merit for feature selection based on rough set theory. In: 2018 6th Iranian joint congress on fuzzy and intelligent systems (CFIS). <https://doi.org/10.1109/CFIS.2018.8336632>
- Sun L, Zhang J, Ding W, Xu J (2022) Mixed measure-based feature selection using the fisher score and neighborhood rough sets. *Appl Intell* 1–25. <https://doi.org/10.1007/s10489-021-03142-3>
- Yang X, Chen H, Li T, Wan J, Sang B (2021) Neighborhood rough sets with distance metric learning for feature selection. *Knowl-Based Syst* 107076:224. <https://doi.org/10.1016/j.knosys.2021.107076>
- Zhou P, Li P, Zhao S, Zhang Y (2021) Online early terminated streaming feature selection based on rough set theory. *Appl Soft Comput* 113:107993. <https://doi.org/10.1016/j.asoc.2021.107993>
- Peng ZA, Xh A, Pl A, Xw B (2019) Ofs-density: a novel online streaming feature selection method - sciencedirect. *Pattern Recogn* 86:48–61. <https://doi.org/10.1016/j.patcog.2018.08.009>
- Liu J, Lin Y, Li Y, Weng W, Wu S (2018) Online multi-label streaming feature selection based on neighborhood rough set. *Pattern Recognit*. <https://doi.org/10.1016/j.patcog.2018.07.021>
- Dai L, Du G, Zhang J, Li C, Li S (2020) Joint multilabel classification and feature selection based on deep canonical correlation analysis. *Concurr Comput Pract Exp* 32(23). <https://doi.org/10.1002/cpe.5864>
- Sun L, Yin T, Ding W, Qian Y, Xu J (2020) Multilabel feature selection using ml-relieff and neighborhood mutual information for multilabel neighborhood decision systems. *Inf Sci* 537:401–424. <https://doi.org/10.1016/j.ins.2020.05.102>
- Fan Y, Liu J, Liu P, Du Y, Lan W, Wu S (2021) Manifold learning with structured subspace for multi-label feature selection. *Pattern Recogn* 120:108169. <https://doi.org/10.1016/j.patcog.2021.108169>
- Song X-F, Zhang Y, Guo Y-N, Sun X-Y, Wang Y-L (2020) Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. *IEEE Trans Evol Comput* 24(5):882–895. <https://doi.org/10.1109/TEVC.2020.2968743>
- Zhang Y, Li HG, Wang Q, Peng C (2019) A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection. *Appl Intell*. <https://doi.org/10.1007/s10489-019-01420-9>
- Baruah HS, Thakur J, Sarmah S, Hoque N (2020) A feature selection method using pso-mi. In: 2020 International conference on computational performance evaluation (comPE), pp 280–284. <https://doi.org/10.1109/ComPE49325.2020.9200034>
- Pedrycz W, Miao D, Li F (2017) Granular multi-label feature selection based on mutual information. *Pattern Recognition the Journal of the Pattern Recognition Society*. <https://doi.org/10.1016/j.patcog.2017.02.025>
- Hatami M, Mehrmohammadi P, Moradi P (2020) A multi-label feature selection based on mutual information and ant colony optimization. In: 2020 28th Iranian conference on electrical engineering (ICEE). <https://doi.org/10.1109/ICEE50131.2020.9260852>

33. Ah A, Mbd B, Np C (2021) A pareto-based ensemble of feature selection algorithms. *Expert Syst Appl.* <https://doi.org/10.1016/j.eswa.2021.115130>
34. Han F, Chen W-T, Ling Q-H, Han H (2021) Multi-objective particle swarm optimization with adaptive strategies for feature selection. *Swarm Evol Comput* 62:100847. <https://doi.org/10.1016/j.swevo.2021.100847>
35. Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput* 2(3):221–248. <https://doi.org/10.1162/evco.1994.2.3.221>
36. Yue C, Suganthan PN, Liang J, Qu B, Yu K, Zhu Y, Yan L (2021) Differential evolution using improved crowding distance for multimodal multiobjective optimization. *Swarm Evol Comput* 62:100849. <https://doi.org/10.1016/j.swevo.2021.100849>
37. Feng J, Gong Z (2022) A novel feature selection method with neighborhood rough set and improved particle swarm optimization. *IEEE Access* 10:33301–33312. <https://doi.org/10.1109/ACCESS.2022.3162074>
38. Demiar J, Schuurmans D (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7(1):1–30. <https://doi.org/10.1007/s10846-005-9016-2>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Shunpan Liang received the Ph.D. degree in mechanical and electronic engineering from Yanshan University, Qinhuangdao, Hebei, China, in 2013.

His main research interests include recommendation systems and machine learning.

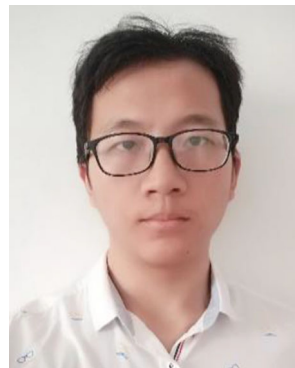


Ze Liu is currently pursuing the master's degree with the School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, China.

His current research interests include streaming feature selection and causal discovery.

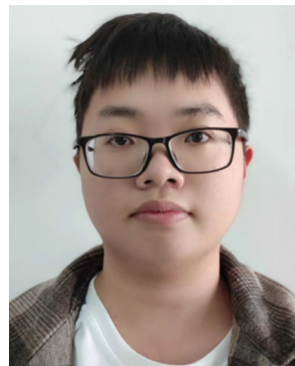


Dianlong You is currently an Associate Professor, received the Ph.D. degree in computer application technology from Yanshan University, Qinhuangdao, HeBei, China, in 2014. From 2017-8 to 2018-8, he was a visiting scholar with the School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, USA. His current research interests include machine learning, streaming feature selection and causal discovery. He has over 20 publications including journals of IEEE TNNLS, IEEE TKDE, IEEE TKDD, INS, and KBS, etc. Dr. You is a member of IEEE.



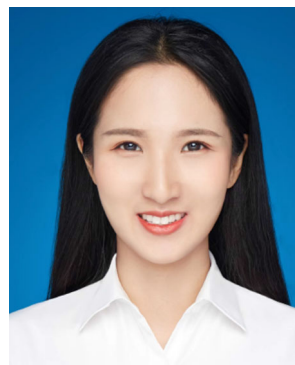
Weiwei Pan is currently pursuing the master's degree with the School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, China.

His current research interests include multi-label data stream classification and incremental learning.



Junjie Zhao is currently pursuing the master's degree with the School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, China.


His current research interests include graph neural network and multi-behavior recommend.



Yefan Cao is currently pursuing the master's degree with the School of Software, Beihang University, Beijing, China.

Her current research interests include computer vision and deep learning.

Affiliations

Shunpan Liang¹  · Ze Liu¹ · Dianlong You¹ · Weiwei Pan¹ · Junjie Zhao¹ · Yefan Cao²

Ze Liu
liuze@stumail.yzu.edu.cn

Dianlong You
youdianlong@sina.com

Weiwei Pan
panweiwei@stumail.yzu.edu.cn

Junjie Zhao
zhaojunjie1314@stumail.yzu.edu.cn

Yefan Cao
caoyefan2019@buaa.edu.cn

¹ School of Information Science and Engineering, Yanshan University, Qinhuangdao, 066004, Hebei, China

² School of Software, Beihang University, Beijing, 100191, Beijing, China