Expert Systems With Applications 204 (2022) 117520

Contents lists available at ScienceDirect

Expert Systems With Applications

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

General assembly framework for online streaming feature selection via Rough Set models

Peng Zhou [a,b,c], Yunyun Zhang [a,b,c], Peipei Li [d,e,*], Xindong Wu [d,e]

[a] Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education of Education of China, Hefei, China
[b] School of Computer Science and Technology, Anhui University, Hefei, China
[c] Information Materials and Intelligent Sensing Laboratory of Anhui Province, Hefei, China
[d] Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education, Ministry of Education of China, Hefei, China
[e] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China

ARTICLE INFO

Keywords:
Feature selection
Online feature selection
Streaming features
General assembly framework
Rough Set models

ABSTRACT

We may not know the entire feature space in advance for real-world applications, and features can exist in a stream mode, called streaming features. Online streaming feature selection aims to select optimal streaming features on the fly and can be summarized into three main components: irrelevant feature discarding, relevant feature selecting, and redundant feature removing. Therefore, the core issue of the streaming feature selection framework is the calculation of the relationship between features. This paper applies Rough Set models to discover the feature relationships for the most crucial advantages: they do not require any domain knowledge and can measure the selected features as integral. After the formal definitions of feature relevance, irrelevance, and redundancy from the Rough Set perspective, we analyze and abstract the feature relationship calculation from three levels: Rough Set model, positive region, and consistency calculation. Then we design a novel general assembly Rough Set based Streaming Feature Selection Framework, named RS-SFSF, which could assemble new algorithms for different problems step by step. Researchers in different areas can quickly build the algorithms they need based on our new framework. To demonstrate the effectiveness of RS-SFSF, we derived four new algorithms based on RS-SFSF by using the classical Rough Set model, neighborhood Rough Set model, and fuzzy Rough Set model, respectively. Extensive experiments conducted on twelve real-world datasets indicate the efficiency of our new framework.

1. Introduction

Feature selection aims to select a minimal subset of features from the original high-dimensional feature space, which is an essential technique for pattern recognition, machine learning, and data mining (Li et al., 2017). Feature selection methods can be broadly categorized as the filter (Cekik & Uysal, 2020), wrapper, and embedded (Yang et al., 2020) according to different selection strategies (Guyon & Elisseeff, 2003). In the past decade, feature selection has attracted many researchers' attention, and plenty of different feature selection methods have been proposed (Cai et al., 2018).

Streaming features are defined as features that flow in one by one over time, whereas the number of samples remains fixed (Wu et al., 2013). There are two main reasons for online streaming feature selection: (1) the features exist in a stream mode in practice; (2) the target datasets are too large to be loaded into the memory once, and we need to handle it in pieces. Specifically, for some real-world applications, not all the features can be required before learning, and the features may exist in a stream mode (Wu et al., 2013). For example, in bioinformatics, for the high cost of conducting wet-lab experiments, acquiring the complete set of features for every training instance is prohibitive, and it is impossible to wait for a complete set of features (Wang et al., 2013). In industrial production, the products processed by different equipments always go through different production processes over time, and continuously generate the streaming features of the same product (Rehman et al., 2018). Besides, with the rapid growth of data volume and dimensions, traditional batch-mode feature selection methods cannot meet the demand of efficiency any more (Wu et al., 2014). For high-dimensional datasets, even if the feature space is known, we can still apply streaming feature selection for the extra advantages, such as low time and space consumptions. Streaming feature selection presents a new perspective in dealing with high-dimensional datasets and has been demonstrated to be effective (Hu et al., 2018).

Unlike traditional feature selection methods, there are two main challenges for online streaming feature selection. First, we cannot

* Corresponding author at: Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education, Ministry of Education of China, Hefei, China.
E-mail addresses: doodzhou@ahu.edu.cn (P. Zhou), e20201047@stu.ahu.edu.cn (Y. Zhang), peipeili@hfut.edu.cn (P. Li), xwu@hfut.edu.cn (X. Wu).

https://doi.org/10.1016/j.eswa.2022.117520
Received 21 August 2021; Received in revised form 11 April 2022; Accepted 4 May 2022
Available online 16 May 2022

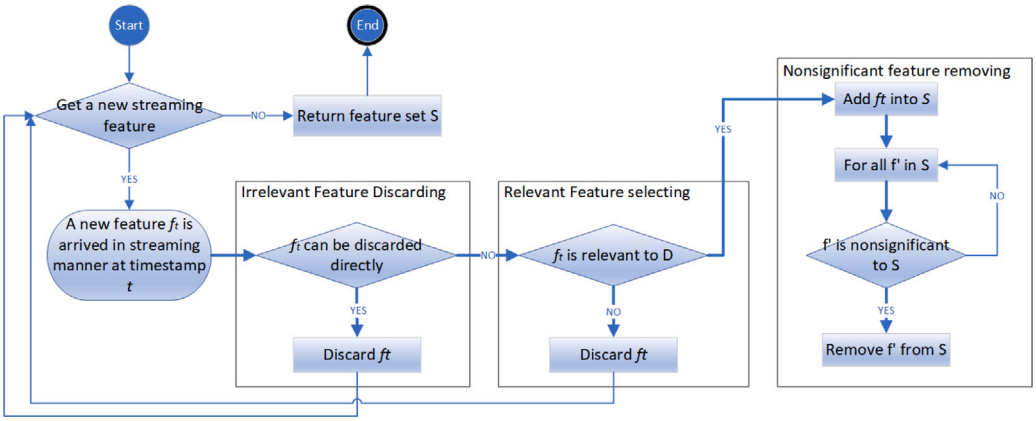


Fig. 1. A generalized streaming feature selection framework.

acquire the information of the entire feature space before learning. It is difficult for some feature selection methods to set proper parameter values in advance for all different datasets without prior knowledge. Thus, the streaming feature selection method should not depend on prior knowledge. Second, the methods must immediately decide whether to keep or discard the new arrival feature on the fly. Once a new arriving feature is discarded, we cannot use it again. Thus, we need to consider the new arrival feature's relationship and the selected feature subset as an integral. Meanwhile, we cannot compare each feature multiple times and rank them globally. In terms of these, most traditional feature selection methods assume all features are known before learning and cannot handle online streaming feature selection directly.

Generally speaking, features can be categorized into three disjoint groups, namely, strong relevance, weak relevance, and irrelevance (Kohavi & John, 1997). For online streaming feature selection, we summarize the procedures into three main steps: (1) irrelevant feature discarding; (2) relevant feature selecting; (3) nonsignificant/redundant feature removing, shown as Fig. 1. In general, we discard irrelevant features, select relevant features and remove redundancy(nonsignificant) features in the selected feature subset. Therefore, the critical issues for streaming feature selection framework are the calculation or judgment of the relationship between features.

1.1. Rough set based online streaming feature selection

There are many online steaming feature selection methods based on different techniques (Hu et al., 2018), such as regularized framework (Perkins & Theiler, 2003), statistical information (Wu et al., 2013), and mutual information (Yu et al., 2016). In this paper, we apply the Rough Set models (Pawlak, 1991) to measure the information between features for two critical advantages: (1) they does not require any domain knowledge other than the given dataset; (2) based on the measurement of dependency degree, they can measure the performance of a candidate feature subset as an integral (Yasmin et al., 2020). The classical Rough Set was initially designed for categorical data. For real-valued data, some extensions of classical Rough Set, such as Neighborhood Rough Set (Hu et al., 2021) and Fuzzy Rough Set (Jensen & Shen, 2008), were proposed to handle it.

Currently, researchers have proposed some Rough Set-based streaming feature selection methods, such as OS-NRRSARA-SA (Eskandari & Javidi, 2016), K-OFSD (Zhou et al., 2017), OFS-A3M (Zhou et al., 2019b), OM-NRS (Liu et al., 2018), and OFS-Density (Zhou et al., 2019a). All these Rough Set-based streaming feature selection methods get some excellent performance in experiments. However, all these methods mentioned above were proposed for specific streaming feature selection problems. For instance, OS-NRRSARA-SA is based on the classical Rough Set model and cannot deal with continuous data. K-OFSD was based on the Neighborhood Rough Set model and designed for high-dimensional and class-imbalanced streaming data. OM-NRS aimed to simultaneously solve online streaming feature selection and multi-label feature selection based on the neighborhood Rough Set model. As far as we know, there is no general online streaming feature selection framework that can assemble new algorithms for different problems in different areas.

1.2. Our contributions

With the in-depth analysis of the streaming feature selection problem and the feature relationship calculation from Rough Set perspective, we propose a new general assembly Rough Set-based Streaming Feature Selection Framework that can assemble new algorithms for different streaming feature selection problems named RS-SFSF, as shown in Fig. 2. There are five steps for RS-SFSF. Step 1, we assemble the feature relationship (dependency) calculation methods from three levels: Rough Set model, positive region, and consistency calculation. Specifically, we first choose an appropriate Rough Set model for the target streaming feature selection problem regarding the feature data type. Then, we design the positive region calculation method and consistency calculation function according to the specific problem constraints and sample distribution. Finally, we assemble these three levels choices into the feature relationship calculation component. Step 2 constructs the irrelevant feature discarding strategy and discards irrelevant streaming features directly for efficiency. Step 3 designs the relevant feature selecting strategy and selects relevant features into the candidate feature subset. Step 4 provides the nonsignificant feature removing strategy and removes nonsignificant features from the candidate feature subset for compactness. Step 5, we assemble all these components into a new streaming feature selection algorithm which we need. Our contributions are as follows:

- With the formal definition and in-depth analysis of the streaming feature selection problem, we summarize this issue into three main components: irrelevant feature discarding, relevant feature selecting, and nonsignificant feature removing.
- We give the formal definitions of feature relevance, irrelevance, and redundancy from the Rough Set perspective. Meanwhile, to maintain a high correlation and low redundancy feature subset, we present three evaluation criteria from the feature subset granularity.
- We propose a new generalized assembly Rough Set-based streaming feature selection framework in terms of definitions of feature relationships from the Rough Set perspective, named RS-SFSF. RS-SFSF can use different Rough Set models, dependency calculation methods, and feature processing strategies to build corresponding new algorithms for various streaming feature selection problems. Based on the RS-SFSF framework, researchers in different areas can quickly construct the algorithms they need step

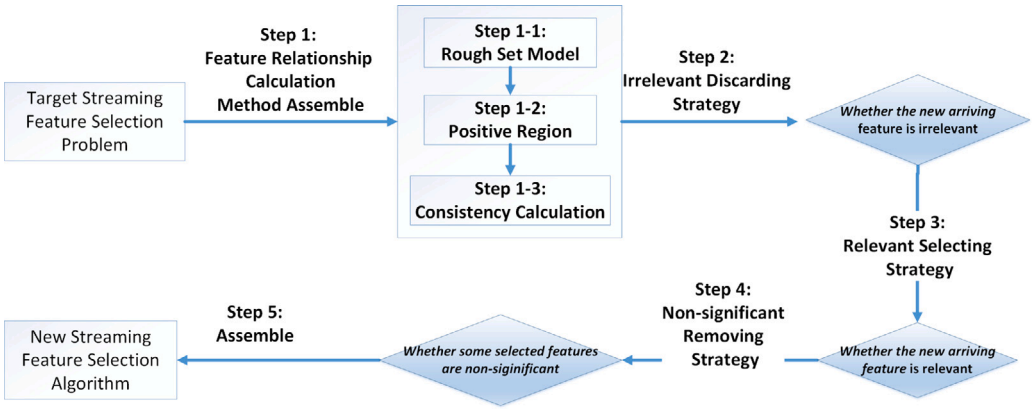


Fig. 2. Our new general assembly Rough Set based online streaming feature selection framework.

by step. Meanwhile, we summarize some existing Rough Set-based streaming feature selection methods within the RS-SFSF framework.

- To validate the effectiveness of RS-SFSF, we derive four new streaming feature selection algorithms based on the RS-SFSF framework with the classical Rough Set model, neighborhood Rough Set model (δ neighborhood relation and k-nearest neighborhood relation), and fuzzy Rough Set model, respectively. Extensive experimental studies indicate that RS-SFSF based algorithms can achieve better predictive accuracy with fewer selected features.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 gives a brief introduction to the classical Rough Set model and its two extensions. Section 4 presents the definitions of feature relationships from Rough Set perspective. Section 5 shows our new streaming feature selection framework and the derived new algorithms. Section 6 reports the experimental results, and we conclude the paper in Section 7.

2. Related work

From the data perspective, feature selection can be divided into two main categories: feature selection with static data and feature selection with streaming data (Li et al., 2017). For static data, all features should be prepared before feature selection taking place. There are many feature selection methods for static data (Cai et al., 2018). More specifically, Dai et al. (2018) proposed two quick feature selection algorithms based on the neighbor inconsistent pair, which can reduce the time consumption in finding a reduct. Wang et al. (2016) designed a new fitting model for feature selection, which can guarantee the membership degree of a sample to its category reaches the maximal value and effectively prevent samples from being misclassified. Zhang et al. (2016) proposed a fuzzy Rough Set-based information entropy for feature selection in a mixed dataset. For streaming data, it can be further divided into the data stream methods and feature stream methods. Neumann et al. (2017) proposed the software EFS (Ensemble Feature Selection) that makes use of multiple feature selection methods and combines their normalized outputs to a quantitative ensemble importance. Eight different feature selection methods have been integrated in EFS, which can be used separately or combined in an ensemble.

In this paper, we focus on feature selection with feature streams. More specifically, Grafting (Perkins & Theiler, 2003) was the first streaming feature selection method based on a regularized framework. The gradient retesting of Grafting over all the selected features increases the total time cost greatly, and it is not easy to choose a good value for the important regularization parameter before learning. Alpha-investing (Zhou et al., 2006) was one of the penalized likelihood ratio methods for streaming feature selection that can run very fast. However, Alpha-investing does not reevaluate the included features and can only select the first one or two features for sparse data. OSFS (Wu et al., 2013) was a conditional independence/dependence tests-based streaming feature selection method that can select a very compact feature subset. Nevertheless, conditional independence/dependence tests need enough training instances, leading to information missing for high dimensionality and small sample datasets. In terms of the Mutual Information theory, SAOLA (a Scalable and Accurate Online feature selection Approach) was proposed for extremely high-dimensional data based on novel online pairwise comparison techniques (Yu et al., 2016). Rahmaninia and Moradi (2018) proposed two online streaming feature selection methods, named OSFSMI and OSFSMI-k, for evaluating the relevancy and redundancy of features in a streaming manner. However, these Mutual Information based methods consider the relationship between pairs of features and cannot measure all the selected features as integral.

In addition to the methods mentioned above, many researchers have recently begun applying the Rough Set theory for streaming feature selection. In terms of the dependency degree model, the Rough Set based methods can measure the selected features as integral. More specifically, OS-NRRSARA-SA (Eskandari & Javidi, 2016) was a classical Rough Set based method that considers the boundary and positive regions during streaming feature selection. K-OFSD (Zhou et al., 2017) was a neighborhood Rough Set based streaming feature selection method that aims to deal with high-dimensional and class-imbalanced streaming data. Based on a new Neighborhood Rough Set relation Gap with adaptive neighbors, OFS-A3M (Zhou et al., 2019b) was a new non-parametric streaming feature selection method that does not need to specify any optimal parameter values before learning, which can select features with a high correlation, high dependency, and low redundancy. Considering the sample distribution problem is usually not uniform, and the dependency degree with a precisely equal constraint is too strict for real-world datasets, Zhou et al. (2019a) proposed a new streaming feature selection method based on an adaptive density neighborhood relation. Liu et al. (2018) proposed a new feature selection framework that can solve online streaming feature selection and multi-label feature selection simultaneously based on a new neighborhood relation. All these streaming feature selection methods demonstrate the effectiveness of applying Rough Set models for the streaming feature selection problem. However, there is no systematic analysis and generalized Rough set-based streaming feature selection framework as far as we know. Thus, in this paper, we study the streaming feature selection problem from the Rough Set perspective in-depth and propose a generalized assembly Rough Set-based framework for streaming feature selection.

Table 1
Nomenclature.

C	Condition feature set
D	Decision feature (Class attribute)
n	Sample size
m	Number of features
x_i	ith sample
f_j	jth feature
U	$\{x_1, x_2, \ldots, x_n\}$
X	A subset of U, $X \subset U$
B	A subset of C, $B \subset C$
S^t	The selected feature subset at time stamp t

3. Rough set models

We summarize some notations used in this article as shown in Table 1.

3.1. Classical rough set

For the classical Rough Set model (Pawlak, 1991), in terms of attributes B, the objects with the same feature values are drawn together and form an equivalence class, denoted by $[x]_B$. The family of elemental granules $\{[x_i]_B \mid x_i \in U\}$ builds a concept system to describe an arbitrary subset of the sample space. For B and X, the elemental granules of lower approximation and upper approximation are defined as follows:

$$\underline{B}X = \{[x_i]_B \mid [x_i]_B \subseteq X, x_i \in U\} \tag{1}$$

$$\overline{B}X = \{[x_i]_B \mid [x_i]_B \cap X \neq \emptyset, x_i \in U\} \tag{2}$$

The lower approximation is also called positive region, denoted as POS_B.

Definition 1. The dependency degree of B to D is defined as the ratio of consistent objects:

$$\gamma_B(D) = \frac{CARD(POS_B(D))}{|U|} \tag{3}$$

where $CARD(POS_B(D))$ denotes the number of positive region objects.

However, the classical Rough Set model cannot handle continuous data directly. Thus, some extensions of classical Rough Set were proposed, such as neighborhood Rough Set (Hu et al., 2008) and fuzzy Rough Set (Jensen & Shen, 2008).

3.2. Neighborhood rough set

In contrast to the classical Rough Set model, the neighborhood Rough Set model uses neighborhood relations to build the concept system (T. & Y., 1998). Different models can be built based on different neighborhood relationships. There are two main types of neighborhood relations, including (1) the fixed distance (δ neighborhood); (2) the fixed number of neighbors (k-nearest neighborhood). The lower approximation and upper approximation of δ neighborhood and k-nearest neighborhood are defined as follows.

Definition 2. For B and X, the lower and upper approximations of X in terms of the δ neighborhood relation are defined as

$$\underline{B_\delta}X = \{x_i \mid \delta(x_i) \subseteq X, x_i \in U\} \tag{4}$$

$$\overline{B_\delta}X = \{x_i \mid \delta(x_i) \cap X \neq \emptyset, x_i \in U\} \tag{5}$$

where $\delta(x_i)$ denotes the objects within a fixed radius δ around x_i.

Definition 3. For B and X, we define the lower and upper approximations in terms of the k-nearest neighborhood relation as

$$\underline{B_K}X = \{x_i \mid K(x_i) \subseteq X, x_i \in U\} \tag{6}$$

$$\overline{B_K}X = \{x_i \mid K(x_i) \cap X \neq \emptyset, x_i \in U\} \tag{7}$$

where $K(x_i)$ denotes the k-nearest neighbors around x_i.

3.3. Fuzzy rough set

In contrast to the classical Rough Set model and the neighborhood Rough Set model, the fuzzy Rough Set model uses fuzzy similarity relation to build the concept system (Zhao et al., 2019). Many fuzzy similarity relations, which describes the similarity between pairs of data samples, can be constructed for this purpose, such as:

$$\mu_{R_a}(x, y) = 1 - \frac{|a(x) - a(y)|}{|a_{max} - a_{min}|} \tag{8}$$

$$\mu_{R_a}(x, y) = exp(-\frac{(a(x) - a(y))^2}{2\delta_a^2}) \tag{9}$$

where a is a feature in C, $\mathcal{R}_a$ is the fuzzy similarity relation induced by feature a, x and y are two arbitrary instances, a_{max} and a_{min} are the maximal and minimal values on a, and δ is the variance of feature a.

Fuzzy equivalence classes are central to the fuzzy-rough set approach just like the crisp equivalence classes are central to Classical Rough Sets.

Definition 4. Given the fuzzy equivalence class $\mathcal{F}$, the fuzzy lower and upper approximations are defined as:

$$\underline{B_\mu}X(x) = \sup_{\mathcal{F} \in U/B} min(\mu_{\mathcal{F}}(x), \inf_{y \in U} max\{1 - \mu_{\mathcal{F}}(y), \mu_X(y)\}) \tag{10}$$

$$\overline{B_\mu}X(x) = \sup_{\mathcal{F} \in U/B} min(\mu_{\mathcal{F}}(x), \sup_{y \in U} min\{\mu_{\mathcal{F}}(y), \mu_X(y)\}) \tag{11}$$

where μ_F and μ_X are the fuzzy similarity relations induced by F and X respectively.

Paper (Radzikowska & Kerre, 2002) gave alternative definitions for the fuzzy lower and upper approximations where a T-transitive fuzzy similarity relation is used to approximate a fuzzy concept X.

$$\underline{B_\mu}X(x) = \inf_{y \in U} I(\mu_{\mathcal{R}_B}(x, y), \mu_X(y)) \tag{12}$$

$$\overline{B_\mu}X(x) = \sup_{y \in U} T(\mu_{\mathcal{R}_B}(x, y), \mu_X(y)) \tag{13}$$

where I is a fuzzy implicator and T is a t-norm. $\mathcal{R}_B$ is the fuzzy similarity relation induced by the subset of features B.

4. Definitions of feature relationships from rough set perspective

This section gives some definitions of feature relationships in terms of Rough Set theory from both the single feature granularity and feature subset granularity.

4.1. Single feature granularity

Features in C can be categorized into three disjoint groups, namely, strong relevance, weak relevance and irrelevance (Wu et al., 2013). From the Rough Set perspective, we use the dependency degree $\gamma_f(D)$ to measure the correlation between feature f and decision class D. We give the definition of feature significance as follows.

Definition 5 (*Feature Significance*). Given B and D, $f \notin B$, the significance of f to B on D is defined as:

$$\sigma_D(f, B) = \gamma_{B \cup f}(D) - \gamma_B(D). \tag{14}$$

where $\gamma_{B \cup f}(D)$ and $\gamma_B(D)$ denote the dependency degrees of feature set B to D with and without feature f respectively.

Definition 5 can measure the significance of a conditioned feature f to a candidate feature subset B in the context of decision feature D. Based on this, we give the definitions of strong relevance, weak relevance, and irrelevance from the Rough Set perspective as follows.

Definition 6 (*Strong Relevance, Weak Relevance, and Irrelevance (RS)*). Given C and D, $f \in C$,

(1) f is strongly relevant to D iff $\forall S \subseteq C\backslash\{f\}$ s.t. $\sigma_D(f, S) > 0$.

(2) f is weakly relevant to D iff it is not strongly relevant, and $\exists S \subset C\backslash\{f\}$ s.t. $\sigma_D(f, S) > 0$.

(3) f is irrelevant to D iff it is neither strongly nor weakly relevant, and $\forall S \subseteq C\backslash\{f\}$ s.t. $\sigma_D(f, S) = 0$.

We use the dataset MONK1 from UCI Machine Learning Repository to illustrate the feature relationships from the Rough Set perspective. MONK1 has 432 instances and six category features $C = \{a_1, a_2, \ldots, a_6\}$. The target concept D is defined by $D = (a_1 = a_2) \vee (a_5 = 1)$.

- Strongly relevant feature. We calculate the feature significance of feature a_5 to $\forall S \subseteq C\backslash\{a_5\}$. All the values are bigger than 0. We list some of them as follows:
  $\sigma_D(a_5, \{a_1\}) = 0.2339; \sigma_D(a_5, \{a_1, a_2\}) = 0.6694; \sigma_D(a_5, \{a_1, a_2, a_3\}) = 0.5565; \sigma_D(a_5, \{a_1, a_2, a_3, a_4\}) = 0.4032; \sigma_D(a_5, \{a_1, a_2, a_3, a_4, a_6\}) = 0.2661$; According to Definition 6, for $\forall S \subseteq C\backslash\{a_5\}$, $\sigma_D(a_5, S) > 0$. Thus, we can conclude that a_5 is a strongly relevant feature from the Rough Set perspective.
- Weakly relevant feature. For feature a_1, we calculate the feature significance of a_1 to some $S \subseteq C\backslash\{a_1\}$ and list some of them as follows:
  $\sigma_D(a_1, \{a_3\}) = 0$; $\sigma_D(a_1, \{a_3, a_4\}) = 0.0161$; $\sigma_D(a_1, \{a_6\}) = 0$; $\sigma_D(a_1, \{a_2\}) = 0.3306$;
  According to Definition 6, there exists $S_1 = \{a_3\}$ which satisfies $\sigma_D(a_1, S_1) = 0$. Thus, a_1 is not a strongly relevant feature. Meanwhile, there exists $S_2 = \{a_2\}$ which satisfies $\sigma_D(a_1, S_2) = 0.3306 > 0$. Thus, we can conclude that a_1 is a weakly relevant feature from the Rough Set perspective.
- Irrelevant feature. For feature a_6, the feature significance is 0. $\sigma_D(a_6, \{a_1\}) = 0$; $\sigma_D(a_6, \{a_1, a_2\}) = 0$; $\sigma_D(a_6, \{a_1, a_2, a_3\}) = 0$; $\sigma_D(a_6, \{a_1, a_2, a_3, a_4\}) = 0$; $\sigma_D(a_6, \{a_1, a_2, a_3, a_4, a_5\}) = 0$; .... Thus, a_6 is an irrelevant feature.

For streaming feature selection, the features flow in one by one over time. At time stamp t, assume the new arriving feature is f_t and the selected feature subset is S^{t-1}. Once a streaming feature is discarded, we cannot use it again. Thus, we cannot test $\forall S \subseteq C\backslash\{f_t\}$ during streaming feature selection to check whether f_t is strongly relevant to D. We refer to both strong relevant feature and weak relevant feature as a relevant feature in the following.

Theorem 1. *Given C and D, at time stamp t, f_t is the new arriving feature and S^{t-1} is the selected feature subset, f_t is relevant to D if $\sigma_D(f_t, S^{t-1}) > 0$.*

Proof. Let $S = S^{t-1}$. Because $\sigma_D(f_t, S) = \sigma_D(f_t, S^{t-1}) > 0$, therefore $\exists S \subset C\backslash\{f_t\}$, $\sigma_D(f, S) \neq 0$. Thus, f_t is relevant to D.

In other words, if the feature significance of f_i to S^{t-1} is bigger than 0, f_i is at least weakly relevant to D.

Theorem 2. *Given C and D, at time stamp t, f_t is the new arriving feature and S^{t-1} is the selected feature subset, f_t is not strongly relevant to D if $\sigma_D(f_t, S^{t-1}) = 0$.*

Proof. Suppose f_t is strongly relevant to D, then $\forall S \subseteq C\backslash\{f_t\}$, $\sigma_D(f_t, S) \neq 0$. Let $S = S^{t-1} \subseteq C\backslash\{f_t\}$, $\sigma_D(f_t, S) = \sigma_D(f_t, S^{t-1}) = 0$. Thus, f_t is not strongly relevant to D.

Thus, for a new arriving feature f_t, if the feature significance of f_i to S^{t-1} is 0, then we should consider to remove this feature.

Based on Markov blankets, Yu and Liu (2004) further divided weakly relevant features into redundant and non-redundant features. We give the definition of feature Redundancy from Rough Set perspective as follows:

Definition 7 (*Redundancy (RS)*). A feature $f \in C$ is a redundant feature if $\exists M \subseteq C\backslash\{f\}$ which makes $\forall f' \in C\backslash(M \cup \{f\})$, $\sigma_f(f', M) = 0$.

Unlike Mutual Information which computes the information between two features, such as $I(f; D)$, we consider the new arriving feature f_t, the selected feature subset S^{t-1}, and the target class D as an integral in terms of Rough Set theory. In other words, the relevancy and redundancy between features f_t and D should be determined in the context of the currently selected features subset S^{t-1}. If $\sigma_D(f, B) = \gamma_{B\cup f}(D) - \gamma_B(D) > 0$, then we consider feature f_t is relevant to D in the context of S^{t-1}. If $\sigma_D(f, B) = 0$, then we consider the feature f_t is redundant to D in the context of S^{t-1}.

To sum up, for the new arriving feature f_t and the selected feature subset S^{t-1}, if $\sigma_D(f_t, S^{t-1}) > 0$, we can safely add f_t into S^{t-1}. However, because the maximal value of γ is 1, it is unrealistic that the significance of all the features is bigger than 0 for high dimensional datasets. In other words, it is unrealistic that each feature in a dataset can increase the dependency degree of the candidate feature subset. Thus, there must exist a lot of features f which satisfies $\sigma_D(f, S) = 0$. Meanwhile, the definition of Rough Set based redundancy (Definition 7) is difficult to apply in the case that we cannot test all the subsets to find M. Thus, we need to consider the streaming feature selection at the feature subset granularity at the same time.

4.2. Feature subset granularity

Rough Set based methods have a huge advantage that it can calculate both the dependency degree of a single feature and the dependency degree of a candidate feature set.

Definition 8 (*Subset Dependency*). The dependency of selected feature subset S^t to D is defined as:

$$Dep(S^t, D) = \gamma_{S^t}(D). \tag{15}$$

From the Rough Set perspective, feature selection aims to select the target features with the maximal dependency degree. Thus, the problem of Rough Set based streaming feature selection can be generally formulated as

$$Max_{S^t} \quad \{Dep(S^t, D)\} \tag{16}$$

Based on this, we propose a baseline algorithm of Rough Set based streaming feature selection as follows.

Algorithm 1: Rough Set Based Streaming Feature Selection Baseline Algorithm (RS-SFS-BA)

Input: Streaming feature f_t at time stamp t;
Output: Selected feature subset S.

```
Initialize the selected feature set S to {};
while f_t is not empty do
    if the dependence degree of S ∪ f_t bigger than the dependence degree of S then
        add f_t into S;
    end
end
```

At timestamp t, if the new arriving feature f_t makes $\gamma_{S\cup f_t}(D)$ increase, f_t will be selected. However, for RS-SFS-BA, once feature f_t is selected, it will not be removed. Thus, we need more evaluation criteria to measure the quality of selected feature subset (Maji & Paul, 2011).

Definition 9 (*Subset Significance*). The significance of selected feature subset S^t to D is defined as:

$$Sig(S^t, D) = \frac{1}{|S^t|} \sum_{f_i \in S^t} \sigma_D(f_i, S^t \backslash \{f_i\}). \tag{17}$$

Besides the maximization of $Dep(S^t, D)$, we can maximize $Sig(S^t, D)$ to make each feature in S is significant to the selected feature subset.

Theorem 3 (*Jensen & Shen, 2008*). *Suppose B is a subset of conditional features, f is an arbitrary conditional attribute that belongs to the dataset, and D is the set of decision attributes. Then $\gamma_{B \cup f}(D) \geq \gamma_B(D)$.*

Proof. The proof of this theorem is available in Jensen and Shen (2008) on page 90.

Theorem 4. *Suppose $\sigma_D(f', S^t \backslash \{f'\}) = 0$, $f' \in S^t$. If we remove f' from S^t, the subset significance of the candidate feature subset will increase.*

Proof. Let $|S^t| = m'$ which consists of m' different features, $S^* = S^t \backslash \{f'\}$. Because $\sigma_D(f', S^t \backslash \{f'\}) = 0$, therefore $\sigma_D(f', S^t \backslash \{f'\}) = \gamma_{S^t}(D) - \gamma_{S^t \backslash \{f'\}}(D) = 0$, $\gamma_{S^t}(D) = \gamma_{S^t \backslash \{f'\}}(D)$.

According to Theorem 3, for $\forall f_i \in S^t, f_i \neq f'$, $\gamma_{S^t \backslash \{f_i\}}(D) \geq \gamma_{S^t \backslash \{f_i, f'\}}(D)$. Therefore, $Sig(S^t, D) = \frac{1}{|S^t|} \sum_{f_i \in S_t} \sigma_D(f_i, S^t \backslash \{f_i\})$

$= \frac{1}{m'}(\gamma_{S^t}(D) - \gamma_{S^t \backslash \{f_1\}}(D) + \gamma_{S^t}(D) - \gamma_{S^t \backslash \{f_2\}}(D) + \cdots + \gamma_{S^t}(D) - \gamma_{S^t \backslash \{f'\}}(D) + \cdots + \gamma_{S^t}(D) - \gamma_{S^t \backslash \{f_{m'}\}}(D))$

$\leq \frac{1}{m'}(\gamma_{S^t \backslash \{f'\}}(D) - \gamma_{S^t \backslash \{f_1, f'\}}(D) + \gamma_{S^t \backslash \{f'\}}(D) - \gamma_{S^t \backslash \{f_2, f'\}}(D) + \cdots + \gamma_{S^t \backslash \{f'\}}(D) - \gamma_{S^t \backslash \{f_{m'}, f'\}}(D))$

$= \frac{1}{m'}(\gamma_{S^*}(D) - \gamma_{S^* \backslash \{f_1\}}(D) + \gamma_{S^*}(D) - \gamma_{S^* \backslash \{f_2\}}(D) + \cdots + \gamma_{S^*}(D) - \gamma_{S^* \backslash \{f_{m'}\}}(D))$

$= \frac{m'-1}{m'} Sig(S^*, D)$, $Sig(S^t, D) < Sig(S^*, D)$.

Thus, the significance of the candidate feature subset will increase if we remove f' from S^t.

As we know, Rough Set based methods usually have a high time complexity due to the calculation of dependency degree. For high-dimensional real-world datasets, there are a huge number of irrelevant and redundancy features. Thus, to reduce the running time, we can discard the irrelevant and low relevance features in advance.

A simple idea for filtering new arriving features is to specify a threshold α. For feature f_t, if $\gamma_{f_t}(D) < \alpha$, it will be discarded directly for saving time. However, it is difficult to specify a proper value for all different datasets.

Definition 10 (*Subset Correlation*). The correlation of selected feature subset S^t to D is defined as:

$$Cor(S^t, D) = \frac{1}{|S^t|} \sum_{f_i \in S^t} \gamma_{f_i}(D). \tag{18}$$

During streaming feature selection, the value of $Cor(S^t, D)$ can automatically adjust in terms of the selected features. Thus, it can be a good threshold for feature filtering.

5. Rough set based streaming feature selection framework

In this section, we give a formal definition in the online streaming feature selection problem at first. Inspired by the challenges of the streaming feature selection issue, we proposed a generalized streaming feature selection framework, which consists of three main components including: (1) irrelevant feature discarding; (2) relevant feature selecting; and (3) nonsignificant feature removing, as shown in Fig. 1. In terms of Rough Set theory, we derived a novel assembly Rough Set based streaming feature selection framework, named RS-SFSF. Besides, we analyze the relationship between RS-SFSF and some existing Rough Set based streaming feature selection methods.

5.1. Definition of online streaming feature selection problem

Let $D \in R^{n \times 1}$ be the initialized decision class label, and $\{f_t \in R^{n \times 1} | t = 1, \ldots, m\}$ be a sequence of input features with fixed number of instances n. Suppose h is a mapping function from samples to class, and t is the current timestamp. For streaming feature selection, we cannot know the information about the entire feature space in advance. At each timestamp t, with the new arriving feature f_t, the problem of streaming feature selection is to find the feature subset S^t that can maximize the mapping function h_t as:

$$Max_{S^t} \quad \{h_t : S^t \rightarrow D\} \tag{19}$$

according to certain measurements.

Algorithm 2: A General Streaming Feature Selection Framework(SFSF)

Input: Streaming feature f_t at time stamp t;
Output: Selected feature subset S.

```
Initialize the selected feature set S to {};
while f_t is not empty do
    //Irrelevant Feature Discarding
    if f_t can be discarded directly then
        discard f_t;
    end
    else
        //Relevant Feature Selecting
        if f_t is a relevant feature then
            add f_t into S;
            //Non-significant Feature Removing
            if some feature f' in S is nonsignificant then
                remove f';
            end
        end
    end
end
```

5.2. A general streaming feature selection framework

Inspired by the challenges of the streaming feature selection and the flow chart as shown in Fig. 1, we proposed a generalized streaming feature selection framework, named SFSF, as shown in Algorithm 2. SFSF consists of three main components:

- **Irrelevant Feature Discarding**: When a new streaming feature f_t arriving at time stamp t, the feature filtering component checks whether f_t is irrelevant or contains very little information about the decision attribute. If true, f_t will be discarded directly.
- **Relevant Feature Selecting**: If f_t is a relevant feature to the decision attribute, f_t will be selected.
- **Non-significant Feature Removing**: If the selecting of f_t makes some features in the candidate feature subset S redundancy or nonsignificance, the redundant and non-significant features will be removed.

5.3. Rough set based streaming feature selection framework

Based on the SFSF framework and definitions of feature relationship from Rough Set perspective, we propose a novel general assembly Rough Set based streaming feature selection framework, named RS-SFSF, which aims to assemble new algorithms for the target streaming feature selection problems, as shown in Fig. 2. RS-SFSF consists of five main steps:

- Step 1: Assemble feature relationship calculation method. A vital issue of RS-SFSF is the calculation of relationships between

features. According to the definitions of feature relationship from the Rough Set perspective, the core of judging the relationship between features is the calculation of feature dependency. We abstract the dependency calculation from three levels: Rough Set model, positive region, and consistency calculation. Algorithm 3 shows a generalized dependency degree calculation method. Specifically, for different datasets, we should choose different Rough Set models at first. For discrete data, we can choose the classical Rough Set model. For continuous data, we can choose the neighborhood Rough Set model. For mixed data, the fuzzy Rough Set model is more suitable. Second, for different Rough Set models, the specific positive region calculation may be different. For example, in the k-nearest neighborhood Rough Set model, the $POS_B(D)$ is the k nearest samples for the target object. The final dependency degree for the condition feature subset B to decision classes D is the sum of each samples' positive region CARD value. For each $x_i \in U$, we can use different methods to calculate the CARD value of the positive region. Table 2 summarizes some common used CARD functions. Finally, we assemble the selected model, the designed dependency calculation method, and the CARD function into the feature relationship calculation component.

- Step 2: Design feature filtering strategy. The feature filtering component aims to discard irrelevant features directly. In practice, we compare the dependency degree of the new arrival feature with a threshold to determine whether it should be discarded. There are two commonly used strategies: (1) a predefined fixed threshold; (2) a dynamically changing threshold, such as the average dependency degrees of the selected features.
- Step 3: Design feature relevant selection strategy. The feature-relevant selecting component aims to select the features that relevant to the decision class. Because the Rough Set model can measure the dependency degree of a feature subset as integral, the most commonly used strategy is to check the relevance via whether adding the new arriving feature can increase the dependency degree of the selected feature subset.
- Step 4: Design non-significant removing strategy. When the feature-relevant selection component adds a new feature into the candidate feature subset, the nonsignificant removing component will check whether some selected features can be removed. According to Definition 5, for each feature in the selected feature subset, if the feature significance is zero, it will be removed as a nonsignificant feature.
- Step 5: Assemble all these components and strategies into the target new algorithm.

Algorithm 3: Dependency Degree Calculation Method

Input: the condition feature subset: B; the decision classes: D; Rough Set model: M;
Output: dependency degree on feature set B: γ_B;

1 n: the number of instances in U;
2 find the positive region as $POS_B(D)$ in terms of M;
3 calculate the value of $POS_B(D)$ as $CARD(POS_B(D))$;
4 $\gamma_B = CARD(POS_B(D))/n$;

To sum up, in terms of Rough Set theory, we can combine different dependency calculation methods and CARD calculation functions in different Rough Set models (classical Rough Set, neighborhood Rough Set, fuzzy Rough Set, Etc.) to assemble the feature relationship calculation component. Meanwhile, within the RS-SFSF framework, we can design different strategies to construct a new streaming feature selection algorithm for the specific application problem.

Table 2
Two common used CARD functions.

Card_Weight	Assume Num_P is the number of samples in $POS_B(D)$ which have the same class as x_i, and Num_S is the size of set $POS_B(D)$, $Card(POS_B(D)) = \frac{Num_P}{Num_S}$.
Card_Consistency	If all the classes of samples in $POS_B(D)$ are the same as the class of x_i, then $Card(POS_B(D)) = 1$; else $Card(POS_B(D)) = 0$.

5.4. Derived algorithms from RS-SFSF framework

In this section, we introduce four derived algorithms from the RS-SFSF framework with the classical Rough Set model, Neighborhood Rough Set model (δ neighborhood relation, and k-nearest neighborhood relation), and Fuzzy Rough Set model, named CRS-SFSF, NRS-SFSF(δ), NRS-SFSF(k), and FRS-SFSF. All of these four derived algorithms have the same three components as the RS-SFSF framework. The difference between these algorithms lies in the different implementations of feature relationship calculation component and different strategies.

Suppose the new arriving feature is f_t at timestamp t, the decision class is D, the selected feature subset is S and f' is a feature in S. For FRS-SFSF, we use Eq. (8) as the similarity function. Meanwhile, we choose the Lukasiewicz t-norm $max(x + y - 1, 0)$ and the Lukasiewicz fuzzy implicator $min(1 - x + y, 1)$ as T and I in Eq. (13) and Eq. (12). α is a predefined threshold for non-significant feature removing. The details of these four algorithms shown as Table 3. Based on Algorithm 3 and Algorithm 2, we can easily construct these four algorithms.

5.5. Time complexity of RS-SFSF

The time complexity of the RS-SFSF framework mainly depends on the time complexity of the dependency degree calculation function. Suppose the time complexity of the dependency degree calculation function is $O(Dep)$.

In the RS-SFSF framework, the feature filtering component's time complexity is $O(Dep)$. The time complexity of the relevant feature selecting component is $O(Dep)$ too. For the nonsignificant feature removing component, the time complexity is $O(Dep * |S|)$ where $|S|$ denotes the number of selected features. Thus, the time complexity of RS-SFSF is $O(Dep * |S| * m)$.

For $O(Dep)$, the time complexity of the classical Rough Set model is usually $O(n)$. For the neighborhood Rough Set model, the time complexity is usually $O(n^2)$. For the fuzzy Rough Set model, the time complexity is usually $O(n^2)$ too. Thus, the time complexity of RS-SFSF is between $O(n * |S| * m)$ and $O(n^2 * |S| * m)$.

When the algorithm selects all the candidate features, the worst time complexity of RS-SFSF is $O(n^2 * m^2)$. However, it is impossible to select all the condition features for real-world datasets. Meanwhile, the feature filtering component in RS-SFSF will significantly reduce the runtime. Experiments in Section 5 reveal that RS-SFSF based algorithms are much faster than the baseline. Thus, the time complexity will be much smaller for real-world applications.

5.6. Comparing with other rough set based streaming feature selection methods

In this section, we discuss the relationships between the proposed framework and several Rough Set based streaming feature selection algorithms, including OS-NRRSARA-SA (Eskandari & Javidi, 2016), K-OFSD (Zhou et al., 2017), OFS-A3M (Zhou et al., 2019b), and OFS-Density (Zhou et al., 2019a). We summarize six aspects including: Rough Set Model, Dependency Function, Card Function, Feature Filtering Component, Relevant Selecting Component, and Nonsignificant Removing Component. Details can be seen from Table 4.

From Table 4, we can observe the followings.

Table 3
Four derived algorithms based on RS-SFSF framework.

RS-SFSF Framework	CRS-SFSF	NRS-SFSF(δ)	NRS-SFSF(k)	FRS-SFSF
Rough Set Model	Classical Rough Set	Neighborhood Rough Set	Neighborhood Rough Set	Fuzzy Rough Set
Positive region	Equivalence Relation	δ Neighborhood Relation	k-nearest Neighborhood Relation	Fuzzy Equivalence Relation
Card Function	Card_Weight	Card_Weight	Card_Weight	Card_Weight
Feature Filtering	if $\gamma_{f_t}(D) = 0$, discard f_t	if $\gamma_{f_t}(D) \le Cor(S, D)$, discard f_t	if $\gamma_{f_t}(D) \le Cor(S, D)$, discard f_t	if $\gamma_{f_t}(D) \le Cor(S, D)$, discard f_t
Relevant Selecting	if $\gamma_{\{f_t\}\cup S} > \gamma_S$, select f_t	if $\gamma_{\{f_t\}\cup S} > \gamma_S$, select f_t	if $\gamma_{\{f_t\}\cup S} > \gamma_S$, select f_t	if $\gamma_{\{f_t\}\cup S} > \gamma_S$, select f_t
Nonsignificant Removing	if $\sigma_D(f', S\backslash\{f'\}) = 0$, remove f'	if $\sigma_D(f', S\backslash\{f'\}) = 0$, remove f'	if $\sigma_D(f', S\backslash\{f'\}) = 0$, remove f'	if $\frac{\gamma_{\{f_t\}\cup S} - \gamma_S}{\gamma_S} < \alpha$ and $\sigma_D(f', S\backslash\{f'\}) = 0$, remove f'

Table 4
RS-SFSF vs. other rough set based algorithms.

RS-SFSF Framework	OS-NRRSARA-SA	K-OFSD	OFS-A3M	OFS-Density
Rough Set Model	Classical Rough Set	Neighborhood Rough Set	Neighborhood Rough Set	Neighborhood Rough Set
Dependency Function	Classical Dependency Calculation	Dep_K (k-nearest neighborhood relation)	Dep_Adapted (Gap neighborhood relation)	Dependency_Density (Density neighborhood relation)
Card Function	Card_Weight	Card_Imbalanced	Card_Weight	Card_Weight
Feature Filtering	none	compared with a pre-defined parameter(α)	maximal $Cor(S^t, D)$	maximal $Cor(S^t, D)$
Relevant Selecting	maximal $Dep(S^t, D)$ OR noise resistant dependency > 0	maximal $Dep(S^t, D)$ OR maximal signal feature dependency	maximal $Dep(S^t, D)$	maximal $Dep(S^t, D)$
Nonsignificant Removing	maximal $Sig(S^t, D)$	none	maximal $Sig(S^t, D)$	maximal $Sig(S^t, D)$

- OS-NRRSARA-SA: OS-NRRSARA-SA was a classical Rough Set based method that can only deal with categorical data directly. There is no feature filtering component for OS-NRRSARA-SA. OS-NRRSARA-SA did not consider the high correlation of the selected feature subset. Thus, there is no guarantee that every feature in S is highly correlated.
- K-OFSD: K-OFSD was a Neighborhood Rough Set based method that is designed for high dimensional class-imbalanced data. K-OFSD was based on k-nearest neighborhood relation and designed the Card_Imbalanced function for class-imbalanced data. However, K-OFSD did not have the nonsignificant feature removing component. Thus, there is no guarantee that the features in the selected feature subset are non-redundant.
- OFS-A3M: OFS-A3M was a nonparametric method based on Neighborhood Rough Set. In the feature filtering step, OFS-A3M uses the subset correlation constraint to select high related features. OFS-A3M proposed a new Gap neighborhood relation, making the algorithm need not specify any parameters before learning.
- OFS-Density: OFS-Density was a novel streaming feature selection method based on the Density neighborhood relation. OFS-Density discarded the features whose dependency degree is less than the candidate feature subset correlation in the feature filtering step. Considering the equal constraint (the dependency degree of adding a new feature into the selected feature subset equals the dependency degree of originally selected feature subset) is too strict for real-world datasets, OFS-Density used a fuzzy equal constraint for nonsignificant feature analysis.

Meanwhile, the worst time complexity of OS-NRRSARA-SA, K-OFSD, OFS-A3M, and OFS-Density is $O(2^m * m * n^2)$, $O(m * n^2)$, $O(m^2 * n^2 * logn)$, and $O(m^2 * n^2 * logn)$ respectively. For K-OFSD, it has the minimal worst-case time complexity, for it does not have the nonsignificant feature removing component.

In general, all these Rough Set based algorithms mentioned above can be included in our new general framework. Thus, RS-SFSF is flexible and versatile.

6. Experiments

6.1. Experimental setup

In this section, we apply the proposed algorithms on twelve real-world datasets from cDNA microarray and NIPS 2003, as shown in Table 5.

We use three Matlab build-in classifiers: KNN(k=3), SVM(with the linear kernel), and CART to evaluate a selected feature subset in our experiments. We perform 5-fold cross-validation on each dataset, and all competing algorithms use the same training and testing data for each fold. The order of streaming features is random, and we run ten times for each dataset. All experimental results are conducted on a PC with AMD(R) 3700X, 3.6 GHz CPU, and 32 GB memory. We conduct the Friedman test at a 95% significance level to validate whether these competing algorithms have a significant difference and use the Nemenyi test as a post-hoc test (Demšar, 2006). Besides, the win/tie/loss (W/T/L for short) counts are summarized in the statistical performance.

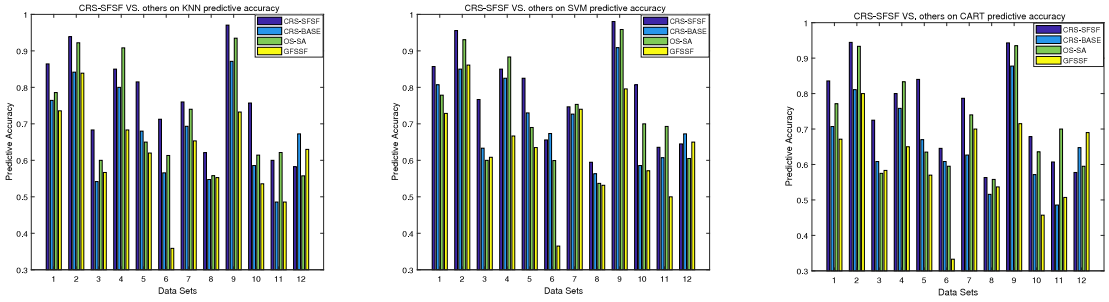
6.2. RS-SFSF with classical rough set model

For the classical Rough Set, the model cannot handle continuous data directly. We discretize the features of datasets in Table 5 into five equal intervals. We compare CRS-SFSF with OS-NRRSARA-SA (Eskandari & Javidi, 2016), GFSSF (Li et al., 2013), and the baseline Algorithm RS-SFS-BA with the classical Rough Set model (denotes as "CRS-BASE") on these datasets.

The predictive accuracy, running time, and the mean number of selected features are shown as Figs. 3 and 4. The running time of OS-NRRSARA-SA on datasets 2, 8, 9, and 12 are 117.8, 154.6, 183.8, and 149.7, respectively. The p-values of Friedman test on KNN, SVM and CART are 5.4095e−07, 3.5947e−10 and 1.3487e−04 respectively. Thus, there is a significant difference in predictive accuracy in cases of KNN, SVM, and CART. The p-values on running time and the mean number of selected features are 2.5934e−10 and 2.2029e−15. Thus, there is a significant difference in running time and the mean number

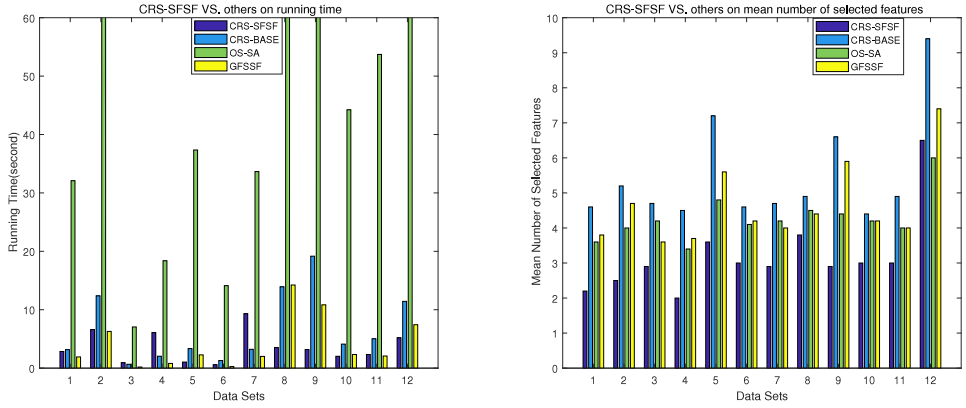
Table 5
Real-world Datasets.

Index	Dataset	Instances/Features	Feature characteristics	Classes
1	Leukemia	72/7,129	Real	2
2	LungCancer	181/12,533	Real	2
3	Colon	62/2,000	Real	2
4	Lymphoma	62/4,026	Real	3
5	Prostate	102/6,033	Real	2
6	Srbct	63/2,308	Real	4
7	Dlbcl	77/7,129	Integer	2
8	Breast	97/24,481	Real	2
9	Ovarian	253/15,154	Real	2
10	Leukemia(3c)	72/7,129	Integer	3
11	Leukemia(4c)	72/7,129	Integer	4
12	Arcene	200/10,000	Integer	2



(a) KNN (b) SVM (c) CART

Fig. 3. CRS-SFSF vs. competing algorithms in cases of KNN, SVM and CART.



(a) Running Time (b) Mean Number of Selected Features

Fig. 4. CRS-SFSF vs. competing algorithms on running time and mean number of selected features.

of selected features. According to the Nemenyi test, the value of CD (critical difference) is 1.3528 in Table 6.

From Fig. 3, Fig. 4 and Table 6, we can observe that:

- CRS-SFSF gets the best performance in cases of KNN, SVM, and CART on predictive accuracy. Meanwhile, CRS-SFSF is significantly better than CRS-BASE and GFSSF, according to the Nemenyi test. Besides, CRS-SFSF gets a higher predictive accuracy than OS-NRRSARA-SA on most of these datasets. This indicates the effectiveness of our proposed framework.
- On the running time, CRS-SFSF gets the minimum average value. Meanwhile, CRS-SFSF is significantly faster than OS-NRRSARA-SA. For some high-dimensional datasets, such as Breast and Ovarian, CRS-SFSF is much faster than "BASE" and OS-NRRSARA-SA. The saving time comes from the feature filtering component in CRS-SFSF. However, for some datasets, such as Lymphoma and Dlbcl, CRS-SFSF spends much more time than "BASE". The extra time is caused by the nonsignificant feature removing component in CRS-SFSF, aiming to select a compact feature subset.
- On the mean number of selected features, CRS-SFSF selects the fewest features on average among all these competing algorithms. CRS-SFSF gets a higher predictive accuracy than the competing algorithms with fewer features. This fully confirms the effectiveness of our new framework.

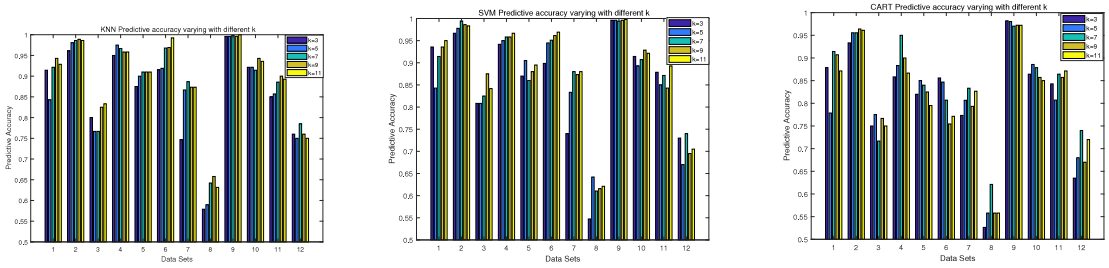
To sum up, in terms of our RS-SFSF framework, the new classical Rough Set based algorithm CRS-SFSF performs better than the compared algorithms on predictive accuracy with fewer selected features.

6.3. RS-SFSF with neighborhood rough set model

We can derive two new neighborhood Rough Set based algorithms within the RS-SFSF framework in terms of δ neighborhood relation and k-nearest neighborhood relation. For parameter k, we test the values of

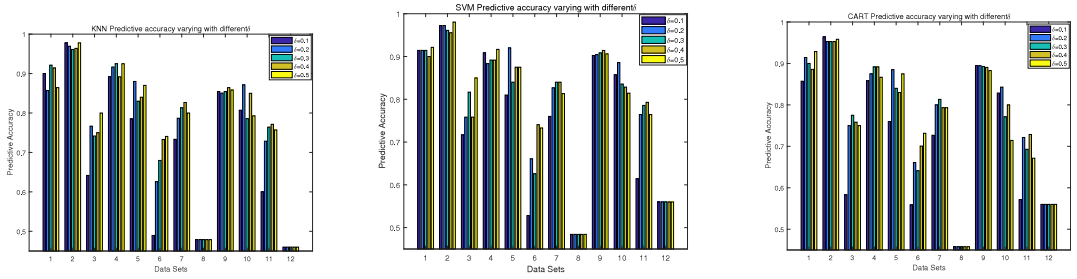
Table 6
The statistical performance of CRS-SFSF vs. compared algorithms.

		CRS-SFSF	CRS-BASE	OS-SA	GFSSF
KNN	W/T/L	9/0/3	1/0/11	2/0/10	0/0/12
	AVG.	**0.7629**	0.6707	0.7087	0.6160
	AVG. RANKS	**1.3333**	2.9583	2.0833	3.6250
SVM	W/T/L	7/0/5	2/0/10	3/0/9	0/0/12
	AVG.	**0.7766**	0.7152	0.7273	0.6377
	AVG. RANKS	**1.0**	3.4583	2.3333	3.2083
CART	W/T/L	9/0/3	0/0/12	2/0/10	1/0/11
	AVG.	**0.7455**	0.6573	0.7089	0.6011
	AVG. RANKS	**1.4167**	2.9167	2.2500	3.4167
Running time	W/T/L	5/0/7	0/0/12	0/0/12	7/0/5
	AVG.	**3.6**	6.6	70.5	4.2
	AVG. RANKS	1.8333	2.6667	4.0	**1.5**
Selected features	AVG.	**3.1**	5.4	4.2	4.6
	AVG. RANKS	**1.0833**	4.0	2.2500	2.6667



(a) KNN (b) SVM (c) CART

Fig. 5. KNN, SVM and CART Predictive accuracy varying with different k.



(a) KNN (b) SVM (c) CART

Fig. 6. KNN, SVM and CART Predictive accuracy varying with different δ.

$k = (3, 5, 7, 9, 11)$ on these datasets as shown in Fig. 5. For parameter δ, we test the values of $\delta = (0.1, 0.2, 0.3, 0.4, 0.5) * Max_{radius}$ on these datasets as shown in Fig. 6, where Max_{radius} denotes the maximum radius of the target object.

From Figs. 5 and 6, we can observe that the δ Neighborhood Rough Set model is more sensitive to parameter values than the k-nearest Neighborhood Rough Set model due to imbalanced sample distribution. Meanwhile, the algorithm with k-nearest neighborhood relation performs better than the algorithm with δ neighborhood relation on average. Thus, in the next experiments, we use NRS-SFSF(k) to compare with the competing streaming feature selection algorithms. Meanwhile, we set $k = 7$ in our experiments with the best performance on average.

We compare NRS-SFSF(k) with OSFS (Wu et al., 2013), SAOLA (Yu et al., 2016), OSFSMI (Rahmaninia & Moradi, 2018), OFS-A3M (Zhou et al., 2019b), OFS-Density (Zhou et al., 2019a) and the baseline Algorithm RS-SFS-BA using the neighborhood Rough Set model on these datasets (denotes as NRS-BASE(k)). We specify the same values of k for NRS-BASE(k) as NRS-SFSF(k). Meanwhile, the significance level α is set to 0.01 for OSFS and SAOLA.

Tables 7–11 summarize the predictive accuracy, running time, and the mean number of selected features of these competing algorithms. The p-values of the Friedman test on KNN, SVM, and CART are 0.3917, 0.4441, and 0.2910, respectively. Thus, there is no significant difference among these competing algorithms on predictive accuracy. The p-values of running time and the mean number of selected features are 3.3810e−09 and 5.6570e−18, respectively. Thus, there is a significant difference among these competing algorithms on running time and the mean number of selected features. According to the Nemenyi test, the value of CD (critical difference) is 2.6004.

From Tables 7–11, we have the following observations.

- On the predictive accuracy, there is no significant difference among all these competing algorithms. NRS-SFSF(k) gets the best performance with both KNN and CART. In the case of SVM, NRS-BASE(k) gets the best performance, and NRS-SFSF(k) just performs a little worse than it. The main reason is that SVM is robust to the number of selected features and tends to perform better with more features. Meanwhile, in Table 11, we can see that NRS-BASE(k) selects five more times features on average than

Table 7
NRS-SFSF vs. competing algorithms in case of KNN.

Dataset	NRS-SFSF(k)	NRS-BASE(k)	OSFS	SAOLA	OFS-A3M	OFS-Density	OSFSMI
Leukemia	0.9357	0.9286	0.8857	0.9214	0.9357	0.9214	**0.9571**
LungCancer	0.9833	0.9806	0.9806	0.9833	0.9833	**0.9889**	0.9611
Colon	0.775	0.825	0.8	0.75	0.7917	0.7333	**0.8**
Lymphoma	0.9333	**0.9833**	0.95	0.95	0.8917	0.95	0.925
Prostate	0.91	0.895	0.9	0.9	0.9	0.87	**0.915**
Srbct	**0.9573**	0.9448	0.9266	0.8944	0.721	0.8406	0.8035
Dlbcl	**0.9067**	0.8533	0.8867	0.8333	0.78	0.8867	0.7933
Breast	0.6368	0.6	0.6105	0.5579	**0.6579**	0.5947	0.6421
Ovarian	0.998	0.9941	0.9941	0.9961	**1**	0.9961	0.9784
Leukemia(3c)	0.9143	0.8571	0.8929	**0.9286**	0.85	0.9214	0.8571
Leukemia(4c)	**0.8786**	0.8286	0.8286	0.8643	0.8643	0.8214	0.75
Arcene	**0.805**	0.695	0.73	0.755	0.66	0.66	0.74
W/T/L	4/0/8	1/0/11	0/0/12	1/0/11	2/0/10	1/0/11	3/0/9
AVG.	**0.8861**	0.8654	0.8654	0.8611	0.8363	0.8487	0.8435
AVG. RANKS	**2.4583**	4.0833	4.1250	3.8750	4.3750	4.5833	4.5000

Table 8
NRS-SFSF vs. competing algorithms in case of SVM.

Dataset	NRS-SFSF(k)	NRS-BASE(k)	OSFS	SAOLA	OFS-A3M	OFS-Density	OSFSMI
Leukemia	**0.9571**	0.9357	0.9	0.9143	0.9429	0.95	0.9429
LungCancer	0.9806	0.9833	0.9778	0.9778	0.975	**0.9889**	0.9778
Colon	0.8083	0.7917	0.8	0.7917	0.8	**0.8167**	0.8083
Lymphoma	0.9333	**0.9917**	0.9667	0.9333	0.8917	0.9417	0.875
Prostate	0.885	0.905	0.9	**0.92**	0.895	0.875	0.915
Srbct	**0.951**	0.9448	0.9161	0.8867	0.7273	0.8189	0.7867
Dlbcl	0.9	**0.9267**	0.9067	0.8533	0.7933	0.8867	0.8067
Breast	0.6421	0.5895	0.5632	0.5526	**0.6947**	0.6316	0.6263
Ovarian	0.996	**1**	0.9941	0.998	**1**	0.9922	0.9803
Leukemia(3c)	0.9	0.9	0.8786	0.9143	0.8286	**0.9143**	0.8429
Leukemia(4c)	0.85	**0.8571**	0.8429	0.8357	0.85	0.8429	0.7643
Arcene	0.685	0.71	**0.76**	0.715	0.71	0.665	0.745
W/T/L	2/0/10	3/1/8	1/0/11	1/0/11	1/1/10	3/0/9	0/0/12
AVG.	0.8740	**0.8779**	0.8671	0.8577	0.8423	0.8603	0.8392
AVG. RANKS	3.2500	**3.0**	4.0833	4.3750	4.7083	3.7500	4.83333

Table 9
NRS-SFSF vs. competing algorithms in case of CART.

Dataset	NRS-SFSF(k)	NRS-BASE(k)	OSFS	SAOLA	OFS-A3M	OFS-Density	OSFSMI
Leukemia	0.85	0.8	0.8286	0.85	**0.9429**	0.9071	0.9214
LungCancer	**0.95**	0.9417	0.9361	0.9472	0.9389	0.9278	0.9472
Colon	0.7333	0.725	0.7417	0.7667	0.7417	0.775	**0.7917**
Lymphoma	**0.9083**	0.8083	**0.9083**	0.875	0.875	0.8417	0.85
Prostate	0.87	0.835	0.83	0.88	0.88	0.84	**0.885**
Srbct	0.8175	0.7944	**0.8818**	0.8392	0.7427	0.8161	0.8098
Dlbcl	0.86	0.82	0.8333	0.82	0.8133	**0.8667**	0.8133
Breast	0.5474	0.5632	0.5579	0.4895	**0.6263**	0.6158	0.5474
Ovarian	0.9781	0.9586	0.9682	0.9824	**0.9804**	0.9686	0.9606
Leukemia(3c)	0.8357	0.85	0.85	**0.9**	0.8143	0.8143	0.8429
Leukemia(4c)	**0.8786**	0.8	0.7571	0.8214	0.8214	0.8357	0.75
Arcene	**0.765**	0.63	0.735	0.67	0.65	0.705	0.695
W/T/L	3/1/8	0/0/12	1/1/10	1/0/11	3/0/9	1/0/11	2/0/10
AVG.	**0.8328**	0.7938	0.819	0.8201	0.8189	0.8261	0.8178
AVG. RANKS	**3.125**	5.5	4.0417	3.3333	4.0833	3.7917	4.1250

NRS-SFSF(k) with only a little improvement using SVM. However, this cannot deny the effectiveness of our new framework.

- OSFSMI is the fastest among all these competing algorithms on the running time, and NRS-BASE(k) is the slowest on average ranks. However, there is no significant difference between NRS-SFSF(k) and OSFSMI on running time. OFS-A3M is a little faster than NRS-SFSF(k), for it need not sort all the neighbors to find the k nearest ones. NRS-SFSF is faster than NRS-BASE for the feature filtering component can save time during streaming feature selection.
- On the mean number of selected features, OFS-A3M selects the fewest, and NRS-BASE(k) selects the most. In terms of the non-significant feature removing component, our new derived algorithm can select fewer features without much predictive accuracy loss. Thus, RS-SFSF can make the selected feature subset compact and informative.

In sum, in terms of our RS-SFSF framework, the new neighborhood Rough Set based algorithm NRS-SFSF(k) can get competing or better performance in the predictive accuracy with fewer selected features.

6.4. RS-SFSF with fuzzy rough set model

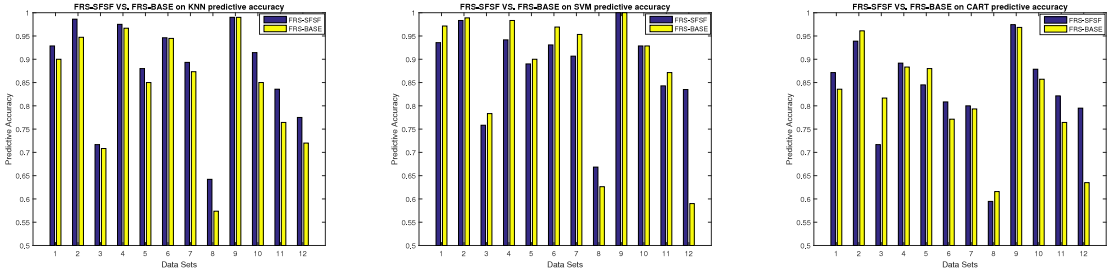
For FRS-SFSF, we compare it with FRS-BASE (the baseline algorithm RS-SFS-BA with fuzzy Rough Set model). Besides, we set $\alpha = 0.05$ as a practical value for FRS-SFSF, just like OFS-Density (Zhou et al., 2019a). The predictive accuracy, running time, and the mean number

Table 10
NRS-SFSF vs. competing algorithms on running time (second).

Dataset	NRS-SFSF(k)	NRS-BASE(k)	OSFS	SAOLA	OFS-A3M	OFS-Density	OSFSMI
Leukemia	1.9448	2.3996	2.0979	1.9214	1.3796	1.9315	**1.1655**
LungCancer	13.0779	15.5031	14.3279	13.3706	22.5779	274.6838	**4.6758**
Colon	0.5142	0.5366	0.562	0.4861	0.1576	0.4959	**0.1785**
Lymphoma	**1.0095**	1.0655	1.0131	1.1049	3.2354	1.0448	2.0227
Prostate	2.2653	2.8307	2.4355	2.2333	**0.8586**	2.3447	1.2132
Srbct	0.5796	0.629	0.603	0.5496	0.6002	0.5372	**0.433**
Dlbcl	2.1495	2.4536	2.1817	2.0384	1.0628	2.1136	**1.1397**
Breast	9.7487	12.6079	10.9088	9.6887	**1.8261**	28.986	1.9456
Ovarian	25.3432	31.0848	26.9498	25.2498	15.4249	24.8025	**7.6841**
Leukemia(3c)	1.939	2.432	2.0885	1.9134	1.3968	1.8939	**1.1654**
Leukemia(4c)	1.9977	2.5824	2.1716	1.9134	1.3534	1.8962	**1.2372**
Arcene	4.0241	5.1495	3.9732	3.7405	**1.0297**	3.6411	1.2356
W/T/L	1/0/11	0/0/12	0/0/12	0/0/12	3/0/9	0/0/12	8/0/4
AVG.	5.38	6.60	5.77	5.35	4.24	28.69	**2.0**
AVG. RANKS	4.2500	6.4167	5.4167	3.50	2.5833	4.0	**1.8333**

Table 11
NRS-SFSF vs. competing algorithms on mean number of selected features.

Dataset	NRS-SFSF(k)	NRS-BASE(k)	OSFS	SAOLA	OFS-A3M	OFS-Density	OSFSMI
Leukemia	9.7	52.3	8.2	3.3	2.4	23.7	8.3
LungCancer	5.5	50.4	5.5	11.1	3.6	46.4	9.1
Colon	11.8	33.8	17.4	5.9	1.5	3.6	4.6
Lymphoma	5.2	31.3	3.2	9.5	2.9	40.6	8.1
Prostate	11.2	47.9	11.5	4.6	1.7	11.7	6.1
Srbct	11.8	51.5	6.5	5.8	2.6	19	6.7
Dlbcl	7.5	43.6	9.4	2.8	2.2	14.2	7.8
Breast	12.4	57.4	34.9	8.2	2.1	15.4	7.7
Ovarian	5.1	88.1	3.7	7.4	3.3	9.2	9.4
Leukemia(3c)	12.2	58.6	11.6	6.8	2.8	21.2	7.6
Leukemia(4c)	19.6	74.5	15.1	5.3	2.5	21.6	8.2
Arcene	17	66.8	26.8	9.3	2.2	17	7.4
AVG.	10.7	54.6	12.8	6.6	**2.4**	20.3	7.5
AVG. RANKS	4.0833	6.9167	4.0417	3.0000	**1.0**	5.4583	3.5000



(a) KNN (b) SVM (c) CART

Fig. 7. NRS-SFSF vs. NRS-BASE in cases of KNN, SVM and CART.

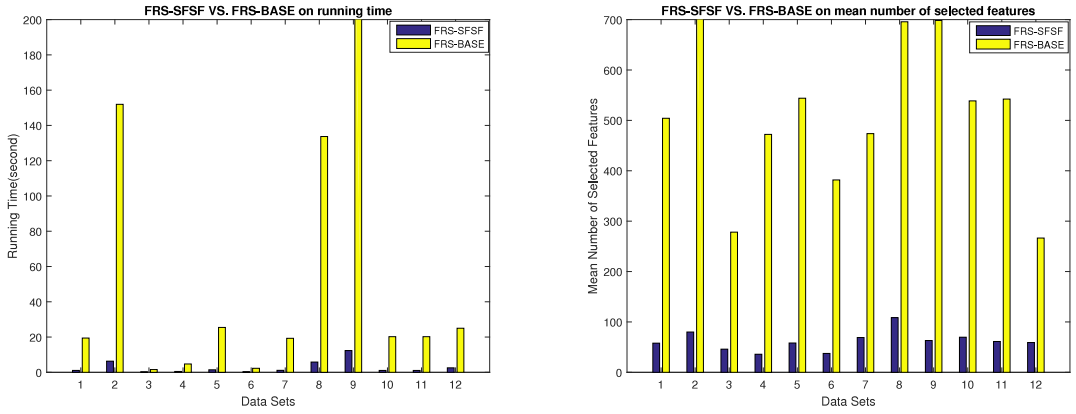
of selected features are shown as Figs. 7 and 8. The statistical performance of FRS-SFSF VS. FRS-BASE is shown as Table 12. According to the Nemenyi test, the value of CD (critical difference) is 0.5658.

The p-values of the Friedman test on KNN, SVM, and CART are 0, 0.0562, and 0.1641, respectively. Thus, there is a significant difference between FRS-SFSF and FRS-BASE with KNN on predictive accuracy. Meanwhile, there is no significant difference in cases of SVM and CART. The p-values of running time and the mean number of selected features are 0 and 0, respectively. Thus, there is a significant difference between FRS-SFSF and FRS-BASE on running time and the mean number of selected features.

In general, FRS-BASE spends 20 times more running time than FRS-SFSF and selects eight times more features than FRS-SFSF on average. Compared with FRS-BASE, our new framework based algorithm selects fewer features but achieves a competing or higher predictive accuracy. Thus, this fully demonstrates the effectiveness of our new proposed framework.

7. Conclusions

In this paper, to make full use of the advantages of the Rough Set model in data mining, we applied Rough Set concepts and methods into streaming feature selection and proposed a generalized assembly Rough Set based framework, named RS-SFSF. To maintain a feature subset with high correlation and low redundancy, RS-SFSF divided the streaming feature selection into three main components: irrelevant feature discarding, relevant feature selecting, and non-significant feature removing. With the definitions of feature relevance, irrelevance, and redundancy from the Rough Set perspective, we present a general feature dependency calculation method from three levels: Rough Set model, positive region, and consistency calculation. Based on the RS-SFSF framework, researchers in different fields can quickly construct new algorithms step by step. To validate our new framework's effectiveness, we derived four new RS-SFSF based streaming feature selection algorithms with the classical Rough Set model, neighborhood Rough Set model (δ neighborhood relation and k-nearest neighborhood relation),



(a) Running Time (b) Mean Number of Selected Features

Fig. 8. NRS-SFSF vs. NRS-BASE on running time and mean number of selected features.

Table 12
The statistical performance of FRS-SFSF vs. FRS-BASE.

		FRS-SFSF	FRS-BASE
KNN	W/T/L	12/0/0	0/0/12
	AVG.	**0.8735**	0.8406
	AVG. RANKS	1	2
SVM	W/T/L	2/2/8	8/2/2
	AVG.	**0.8851**	0.8804
	AVG. RANKS	1.75	**1.25**
CART	W/T/L	8/0/4	4/0/8
	AVG.	**0.8280**	0.8151
	AVG. RANKS	**1.3333**	1.6667
Running time	W/T/L	12/0/0	0/0/12
	AVG.	**2.82**	56.24
	AVG. RANKS	**1**	2
Selected features	AVG.	**62.1**	516.4
	AVG. RANKS	**1**	2

and fuzzy Rough Set model, respectively. Experiments conducted on twelve real-world datasets demonstrated that RS-SFSF could select a compact and informative feature subset.

Based on our new framework, it is easy to derive new practical streaming feature selection algorithms according to the specific application problems. For example, we can construct a new algorithm based on the neighborhood Rough Set model to handle mixed streaming features with the heterogeneous Euclidean-overlap metric (HEOM) based dependency function or a new algorithm with the fuzzy Rough Set model and a proper fuzzy similarity relation. However, Rough Set-based methods usually have a high time complexity. Thus, we will focus on reducing the time complexity of Rough Set-based streaming feature selection in the future. Meanwhile, the model's scalability and robustness will be discussed in our future work.

CRediT authorship contribution statement

Peng Zhou: Conceptualization, Methodology, Software, Writing – original draft, Funding acquisition. **Yunyun Zhang:** Software, Validation, Investigation, Writing – review & editing. **Peipei Li:** Formal analysis, Funding acquisition. **Xindong Wu:** Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grants (61906056, 61976077).

References

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: a new perspective. *Neurocomputing*, *300*, 70–79.
Cekik, R., & Uysal, A. K. (2020). A novel filter feature selection method using rough set for short text data. *Expert Systems with Applications*, *160*, Article 113691.
Dai, J., Hu, Q., Hu, H., & Huang, D. (2018). Neighbor inconsistent pair selection for attribute reduction by rough set approach. *IEEE Transactions on Fuzzy Systems*, *26*(2), 937–950.
Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*(1), 1–30.
Eskandari, S., & Javidi, M. (2016). Online streaming feature selection using rough sets. *International Journal of Approximate Reasoning*, *69*(C), 35–57.
Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.
Hu, M., Tsang, E. C., Guo, Y., Chen, D., & Xu, W. (2021). A novel approach to attribute reduction based on weighted neighborhood rough sets. *Knowledge-Based Systems*, *220*, Article 106908.
Hu, Q., Yu, D., Liu, J., & Wu, C. (2008). Neighborhood rough set based heterogeneous feature subset selection. *Information Sciences*, *178*(18), 3577–3594.
Hu, X., Zhou, P., Li, P., Wang, J., & Wu, X. (2018). A survey on online feature selection with streaming features. *Frontiers of Computer Science*, *12*(3), 479–493.
Jensen, R., & Shen, Q. (2008). *Computational intelligence and feature selection: rough and fuzzy approaches*. JOHN WILEY & SONS: IEEE Press Series on Computational Intelligence.
Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324.
Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys*, *50*(6), 1–45.

Li, H. G., Wu, X. D., Li, Z., & Ding, W. (2013). Group feature selection with streaming features. In *IEEE 13th international conference on data mining* (pp. 1109–1114).

Liu, J., Lin, Y., Li, Y., Weng, W., & Wu, S. (2018). Online multi-label streaming feature selection based on neighborhood rough set. *Pattern Recognition*, *84*, 273–287.

Maji, P., & Paul, S. (2011). Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data. *International Journal of Approximate Reasoning*, *52*, 408–426.

Neumann, U., Genze, N., & Heider, D. (2017). EFS: an ensemble feature selection tool implemented as R-package and web-application. *BioData Mining*, *10:21*(1), 1–9.

Pawlak, Z. (1991). *Rough sets - Theoretical aspects of reasoning about data*. Dordrecht, Boston: Kluwer Academic Publishers.

Perkins, S., & Theiler, J. (2003). Online feature selection using grafting. In *Proceedings of the 20th international conference on machine learning* (pp. 592–599).

Radzikowska, A. M., & Kerre, E. E. (2002). A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems*, *126*(2), 137–155.

Rahmaninia, M., & Moradi, P. (2018). OSFSMI: ONline stream feature selection method based on mutual information. *Applied Soft Computing*, *68*, 733–746.

Rehman, M. H. u., Ahmed, E., Yaqoob, I., Hashem, I. A. T., Imran, M., & Ahmad, S. (2018). Big data analytics in industrial IoT using a concentric computing model. *IEEE Communications Magazine*, *56*(2), 37–43.

T., L., & Y., G. (1998). Computing on binary relations I: Data mining and neighborhood systems. In *Proceedings of the rough sets in knowledge discovery* (pp. 107–121).

Wang, C., Qi, Y., Shao, M., Hu, Q., Chen, D., Qian, Y., & Lin, Y. (2016). A fitting model for feature selection with fuzzy rough sets. *IEEE Transactions on Fuzzy Systems*, *25*(4), 741–753.

Wang, J., Zhao, P., Hoi, S. C., & Jing, R. (2013). Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering*, *26*(3), 698–710.

Wu, X., Yu, K., Ding, W., Wang, H., & Zhu, X. (2013). Online feature selection with streaming features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(5), 1178–1192.

Wu, X., Zhu, X., Wu, G., & Ding, W. (2014). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, *26*(1), 97–107.

Yang, Z., Ye, Q., Chen, Q., Ma, X., Fu, L., Yang, G., Yan, H., & Liu, F. (2020). Robust discriminant feature selection via joint L2, 1-norm distance minimization and maximization. *Knowledge-Based Systems*, *207*, Article 106090.

Yasmin, G., Das, A. K., Nayak, J., Pelusi, D., & Ding, W. (2020). Graph based feature selection investigating boundary region of rough set for language identification. *Expert Systems with Applications*, *158*, Article 113575.

Yu, L., & Liu, H. (2004). Eficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, *5*(12), 1205–1224.

Yu, K., Wu, X., Ding, W., & Pei, J. (2016). Scalable and accurate online feature selection for big data. *ACM Transactions on Knowledge Discovery from Data*, *11*(2), 1–39.

Zhang, X., Mei, C., Chen, D., & Li, J. (2016). Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy. *Pattern Recognition*, *56*, 1–15.

Zhao, H., Wang, P., Hu, Q., & Zhu, P. (2019). Fuzzy rough set based feature selection for large-scale hierarchical classification. *IEEE Transactions on Fuzzy Systems*, *27*(10), 1891–1903.

Zhou, J., Foster, D. P., Stine, R. A., & Ungar, L. H. (2006). Streamwise feature selection. *Journal of Machine Learning Research*, *3*(2), 1532–4435.

Zhou, P., Hu, X., Li, P., & Wu, X. (2017). Online feature selection for high-dimensional class-imbalanced data. *Knowledge-Based Systems*, *136*, 187–199.

Zhou, P., Hu, X., Li, P., & Wu, X. (2019). OFS-density: A novel online streaming feature selection method. *Pattern Recognition*, *86*, 48–61.

Zhou, P., Hu, X., Li, P., & Wu, X. (2019). Online streaming feature selection using adapted neighborhood rough set. *Information Sciences*, *481*, 258–279.