Online Scalable Streaming Feature Selection via Dynamic Decision

PENG ZHOU, SHU ZHAO, and YUANTING YAN, Key Laboratory of Intelligent Computing and Signal Processing (Anhui University), Ministry of Education, School of Computer Science and Technology, Anhui University

XINDONG WU, Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei University of Technology, and Mininglamp Academy of Sciences, Mininglamp Technology

Feature selection is one of the core concepts in machine learning, which hugely impacts the model's performance. For some real-world applications, features may exist in a stream mode that arrives one by one over time, while we cannot know the exact number of features before learning. Online streaming feature selection aims at selecting optimal stream features at each timestamp on the fly. Without the global information of the entire feature space, most of the existing methods select stream features in terms of individual feature information or the comparison of features in pairs. This article proposes a new online scalable streaming feature selection framework from the dynamic decision perspective that is scalable on running time and selected features by dynamic threshold adjustment. Regarding the philosophy of "Thinking-in-Threes", we classify each new arrival feature as selecting, discarding, or delaying, aiming at minimizing the overall decision risks. With the dynamic updating of global statistical information, we add the selecting features into the candidate feature subset, ignore the discarding features, cache the delaying features into the undetermined feature subset, and wait for more information. Meanwhile, we perform the redundancy analysis for the candidate features and uncertainty analysis for the undetermined features. Extensive experiments on eleven real-world datasets demonstrate the efficiency and scalability of our new framework compared with state-of-the-art algorithms.

CCS Concepts: • **General and reference → Cross-computing tools and techniques;**

Additional Key Words and Phrases: Feature selection, feature streams, scalable feature selection, three-way decision

This work is supported in part by the National Key Research and Development Program of China under grant 2016YFB1000901, the National Natural Science Foundation of China under grants (61906056, 61976077, 61806002), and the Program for Changjiang Scholas and Innovative Research Team in University (PCSIRT) of the Ministry of Education under grant IRT17R32.

Authors' addresses: P. Zhou, S. Zhao, and Y. Yan, Key Laboratory of Intelligent Computing and Signal Processing (Anhui University), Ministry of Education, School of Computer Science and Technology, Anhui University, 111 Jiulong Road, Hefei, Anhui, China; emails: {doodzhou, zhaoshuzs, ytyan}@ahu.edu.cn; X. Wu, Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei University of Technology 193 Tunxi Road, Hefei, Anhui, China 230009, and Mininglamp Academy of Sciences, Mininglamp Technology, China; email: xwu@hfut.edu.cn.

1556-4681/2022/03-ART87 $15.00

https://doi.org/10.1145/3502737

ACM Reference format:
Peng Zhou, Shu Zhao, Yuanting Yan, and Xindong Wu. 2022. Online Scalable Streaming Feature Selection via Dynamic Decision. *ACM Trans. Knowl. Discov. Data.* 16, 5, Article 87 (March 2022), 20 pages.
https://doi.org/10.1145/3502737

1 INTRODUCTION

Feature selection aims at selecting a minimal subset from the original datasets that can retain the optimum salient characteristics necessary [11]. With the increase of data volume and dimension, feature selection has become a fundamental and necessary technology for machine learning and data mining [9]. Traditional feature selection methods assume that all the instances and features in the target datasets can be required before learning. However, in some real-world applications, we are more likely faced with data streams or feature streams, or both [8]. For example, in social media, Twitter produces more than 500 million tweets every day, and numerous slang words (features) are continuously being generated. In an industrial production line, the same product needs to go through multiple processes, and different processes continue to generate various features for this product. Online streaming feature selection deals with features arriving one by one over time while the number of instances remains fixed [21]. There are two main challenges for online streaming feature selection: (1) the entire feature space is unknown before learning; (2) the algorithm needs to decide whether to retain or discard each new arriving streaming feature on the fly.

Generally speaking, features can be categorized into three disjoint groups, namely, strongly relevant, weakly relevant, and irrelevant [6]. Strongly relevant features provide information for the outcome in any context, while irrelevant features provide no information. Weakly relevant features provide information for the outcome in some context. Yu and Liu [31] further divided weakly relevant features into redundant and non-redundant features based on Markov blankets. Ideally, feature selection methods aim at selecting all strongly relevant features, weak and nonredundant features, and none of the irrelevant features. However, in practice, it is impossible to apply these definitions directly for high-dimensional datasets due to the curse of dimensionality [29]. Thus, most of the existing online streaming feature selection methods approximate the feature relationships in terms of specific measurements and focus on the selection of the most informative streaming features [5].

From the decision perspective, we consider online streaming feature selection as making decisions for each new arrival feature to minimize the overall decision risks. In other words, we wish the decision risk for each new arriving feature "as low as possible" [4]. Based on the philosophy of "Thinking-in-Threes" that understanding and processing a whole through three distinct and related parts [25] and the superiority of three-way decision [23], for each new arriving feature, we can make one of the following three decisions: selecting, discarding, or delaying. Suppose $\gamma_f(d) \in [0, 1]$ denotes the membership grade between feature f and decision class d. With a pair of thresholds α and β, each feature f can be classified into one of the following three regions: discarding ($0 \leq \gamma_f(d) \leq \alpha$), delaying ($\alpha < \gamma_f(d) < \beta$), and selecting ($\beta < \gamma_f(d) \leq 1$), shown as Figure 1. Meanwhile, we use the membership grade as a measurement to approximate and classify the features into three disjoint groups: strongly relevant, weakly relevant, and irrelevant. In other words, we select strongly relevant features and discard irrelevant features. For weakly relevant features, we delay making the decisions and wait for more information. Without the information of the entire feature space, the three-way decision can reduce the risk for weakly relevant streaming features. Meanwhile, with the dynamic adjustment of α and β, we can make decisions for each new arrival feature with low risk. As far as we know, this is the first work considering online streaming feature selection from a decision-making perspective.

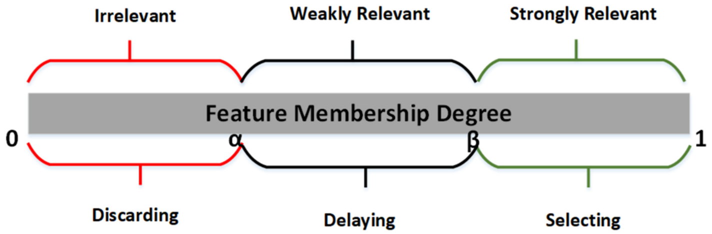


Fig. 1. The three regions of selecting (strongly relevant), discarding (irrelevant), and delaying (weakly relevant) in terms of feature membership degree and two thresholds α and β.

During online streaming feature selection, suppose we get a new arriving feature f_t at timestamp t, and we should decide to retain or discard it on the fly. Theoretically, we can store all the streaming features and do not discard anyone. However, the storage space of the machine running the algorithm is always limited. Meanwhile, irrelevant and redundant features can be discarded directly, and it is unnecessary to store these useless features. Besides, it will take much more running time as more and more features are stored and be used in the algorithm. Therefore, unlike traditional feature selection methods that can compare features many times, the discarded features in online streaming feature selection cannot be required and used again. Thus, most of the existing online streaming feature selection methods select features in individual feature information or compare features in pairs. Meanwhile, with the tremendous growth in the instances and dimensionalities of datasets, the scalability of the feature selection algorithm is crucial. For a scalable online streaming feature selection method, the running time, and the number of selected features should have a linear or sublinear relationship with the dimensionality of the target datasets while maintaining a satisfactory performance on the final selected features. However, most of the existing streaming feature selection methods focus on the performance of final selected features and ignore the scalability on running time and number of selected features, which is significantly essential for very large-scale and streaming datasets.

Motivated by this, we propose a new **Online Scalable Streaming Feature Selection framework from a Dynamic Decision perspective**, (**OSSFS-DD**), shown as Figure 2. To minimize the overall decision risks during online streaming feature selection, we try to select the most informative features and delay making decisions for undetermined features. Based on the idea of a three-way decision, OSSFS-DD updates the values of α and β in terms of the global statistical information that can control the size of the candidate feature subset and undetermined feature subset linear to the dimensionality of the target datasets. Meanwhile, in terms of the information theory [18, 37], we discard redundant features in the candidate feature subset if the joint information of two features (the information of each feature is bigger than β) is small than $2 * \beta$. If two features (the information of each feature is smaller than β) in the undetermined feature subset provide an information bigger than $2 * \beta$, we move these two features from the undetermined feature subset into the candidate feature subset. Our main contributions can be summarized as follows:

- We first give the formal definition of the online streaming feature selection problem from the decision-making perspective. Regarding the philosophy of "Thinking-in-Threes", we classify each new streaming feature into selecting, discarding, or delaying. As far as we know, this is the first attempt to handle this problem via dynamic decisions.
- We propose a new online scalable streaming feature selection framework via dynamic decisions to minimize the overall decision risks. With the dynamical update of global statistical information, we adjust the thresholds dynamically that guarantee the selected features are the most informative ones and the size of the candidate feature subset linear to the

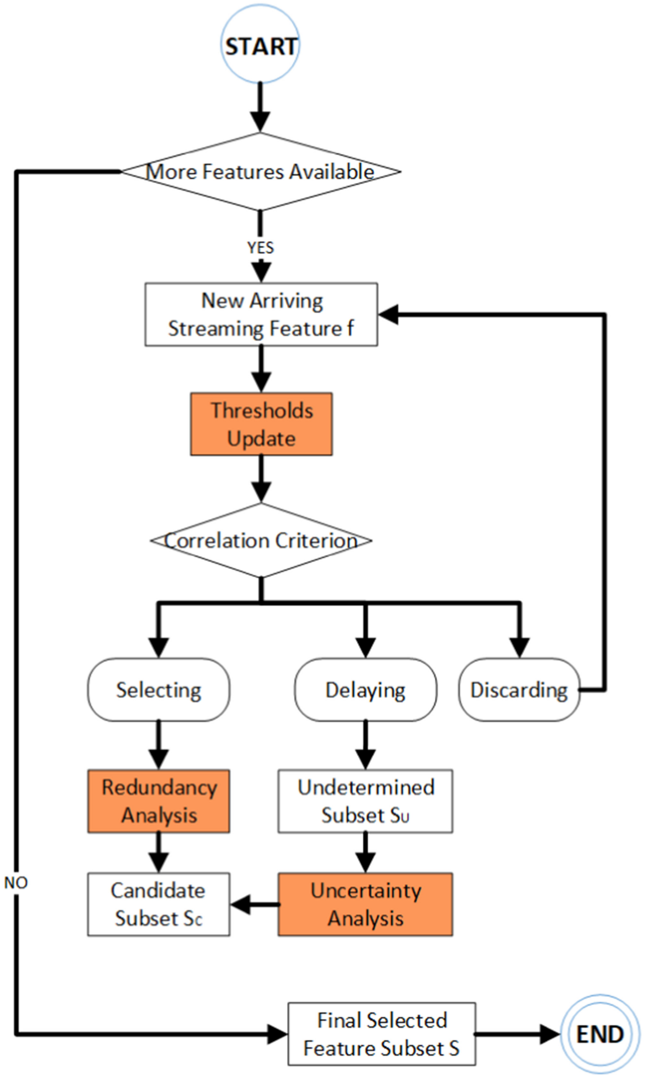


Fig. 2. Our new online scalable streaming feature selection framework via dynamic decisions. For each new arriving streaming feature f, we make one of the following three decisions: selecting, discarding, or delaying.

dimensionality of target datasets. Meanwhile, to compact the selected features, we discard the redundant features in the candidate feature subset and move features from the undetermined feature subset into the candidate feature subset with uncertainty analysis.

— Extensive experiments on eleven real-world datasets indicate the effectiveness and scalability of our proposed method compared with seven state-of-the-art streaming feature selection algorithms.

The rest of this article is organized as follows. Section 2 describes the related work. Section 3 gives the problem formalization from the decision-making perspective and proposes a new scalable framework to handle it. Experimental analyses are presented in Section 4. Finally, Section 5 makes a brief conclusion.

2 RELATED WORK

In terms of different selection strategies, feature selection methods can be broadly categorized as the filter, wrapper, and embedded [3]. Filter methods evaluate the feature importance according to specific criteria independent of any learning algorithms, while wrapper methods evaluate the selected features' quality with a predefined learning algorithm. Embedded methods perform feature selection in the process of model construction. With the rapid increase in data, feature selection became more and more important in machine learning and data mining. For example, in Web service, a covering-based quality prediction method was proposed via neighborhood-aware matrix factorization and was validated on a real-world dataset containing 1,974,675 Web service invocation records [32]. Meanwhile, the dimensionality is extremely high for some big data real-world applications, and the features may exist in a stream mode. For streaming feature selection, we cannot require all the features before learning. Thus, most of the existing streaming feature selection methods are designed in filter mode [5].

Specifically, Grafting [15] was the first online streaming feature selection method that is based on the stagewise gradient descent and treats feature selection as an integral part of learning a predictor within a regularized framework. Alpha-investing [33] was a method based on streamwise regression for online feature selection and used the penalized likelihood ratio to measure the relevance for the new arriving features. Wu et al. [21] presented two algorithms (OSFS and fast-OSFS) that contained two major steps: online relevance analysis (discarding irrelevant features) and online redundancy analysis (eliminating redundant features) in terms of the conditional independence/dependence test. Li et al. [7] proposed two methods at the group and individual feature levels, respectively, by exploiting entropy and mutual information in information theories. Yu et al. [29] proposed the SAOLA approach for high-dimensional data that employed novel online pairwise comparison techniques to maintain a parsimonious model over time in an online manner based on the mutual information theory. For multilabel streaming feature selection, Lin et al. [10] introduced fuzzy mutual information to evaluate the quality of features and designed efficient algorithms to conduct multilabel feature selection when the feature space is completely known or partially known in advance. Rahmaninia et al. [17] proposed two online streaming feature selection methods, named OSFSMI and OSFSMI-k, for evaluating the relevancy and redundancy of features in terms of the mutual information in a streaming manner. Wu et al. [19] focused on the problem of online feature selection from capricious streaming features and proposed a new method that adopts latent factor analysis to preprocess capricious streaming features for completing their missing entries before conducting feature selection. SFS-FI [37] considered the interaction between features during online streaming feature selection and proposed a new method that can select features to interact with each other. Although the above methods are based on different technologies for online streaming feature selection, most of them include relevant feature selection and redundant feature removal components. Inspired by this, our new framework also considers the relevance and redundancy between features during online feature selection.

Besides, considering the most critical advantages that do not require any domain knowledge other than the given dataset, many researchers have begun applying the Rough Set theory for streaming feature selection. More specifically, based on the classical Rough Set model, Eskandari et al. [2] proposed a new method for online streaming feature selection that considers both the boundary and positive regions and uses a noise-resistant dependency measure to search for reduces. Zhou et al. [34] proposed a k-nearest neighborhood relation-based online streaming feature selection method from the Neighborhood Rough Set perspective for high-dimensional and class-imbalanced data. Liu et al. [13] proposed a new feature selection framework based on neighborhood Rough Set that can solve online streaming feature selection and multi-label feature selection simultaneously. OFS-A3M [36] was a new non-parametric streaming feature selection method based on the gap neighborhood relation for streaming feature selection that aims at selecting features with high correlation, high dependency, and low redundancy. Zhou et al. [35] proposed the OFS-Density method based on adaptive density neighborhood relation that can select features with high relevance and low redundancy. Most of these Rough Set-based methods use the dependence degree to measure stream features with high time complexity. These Rough Set-based methods have demonstrated the effectiveness of applying Rough Set theory for the problem of online streaming feature selection. However, all these algorithms have a common shortcoming of high time complexity. Therefore, we attempt to design an efficient online streaming feature selection framework in this article.

Recently, some new works have studied the online streaming feature selection from other perspectives. For example, GF-CSF [19] conducted online feature selection from capricious streaming features, where features flow in one by one with some random missing entries while the number of data instances remains fixed. I-SFS and G-SFS [14] were two streaming feature selection methods for multi-label datasets where the multiple labels are reduced to a lower-dimensional space. These two methods grouped the similar labels before performing the selection method to improve the selection quality and make the model efficient. LOSSA [20] was a latent-factor-analysis-based online sparse-streaming-feature selection algorithm, which aims at implementing online feature selection from sparse streaming features. OCFSSFs [27] was an online causal feature selection method for streaming features through mining Markov blanket containing PC (parents and children) and spouses.

Nevertheless, all these methods mentioned above are not scalable to running time and selected features simultaneously. Therefore, this article tries to consider the problem from the decision-making perspective and use global statistic information to select a scalable ratio of the most informative features.

3 THE PROPOSED METHOD

This section first gives the formal definition of online streaming feature selection from the dynamic decision perspective. Then, we present a brief introduction of different feature relationships and the idea of a three-way decision. After that, we propose our new OSSFS-DD framework and point out three main issues that need to be solved during online streaming feature selection. We discuss our new framework in detail at last.

3.1 Problem Definition

Let $F = \{f_i | i = 1, \ldots, T\}$ be a sequence of streaming features, where $f_i = [x_1^i, x_2^i, \ldots, x_n^i]^T$ is a pattern of n samples received at the ith timestamp. $d = [y_1, y_2, \ldots, y_n]^T$ is the observed decision class of the n samples, where y_i is the class label. At timestamp t, we get a new feature f_t and should decide whether retain or discard f_t on the fly. Meanwhile, for discarded features, we cannot use and select them again.

Definition 1 (Online Streaming Feature Selection from Decision-making Perspective). Suppose $E(f)$ is a decision evaluation function on feature f, $E(\cdot) \in [0, 1]$, $0 \leq \alpha < \beta \leq 1$. For streaming feature f_t at timestamp t, there are three-way decisions defined as follows:

(1) Selecting region: $SEL_{(\alpha,\beta)} = \{f | E(f) \geq \beta\}$;
(2) Discarding region: $DIS_{(\alpha,\beta)} = \{f | E(f) \leq \alpha\}$;
(3) Delaying region: $DEL_{(\alpha,\beta)} = \{f | \alpha < E(f) < \beta\}$;

Online streaming feature selection aims at making decisions for each streaming feature that minimize the overall decision risks

$$Min \quad \sum_{f_t \in F} [1 - E(f_t)]. \tag{1}$$

As we know that, feature selection is an NP-hard problem [12]. Therefore, most feature selection methods adopt greedy strategies and try to choose the best features at each round. From the decision-making perspective, we can consider online streaming feature selection as a series of dynamic decisions that aim at minimizing the overall decision risks.

For a dataset $\mathbf{D}$, suppose the feature set is C, and the decision class is d. Each feature f in C can be categorized into three disjoint groups: strongly relevant, weakly relevant, and irrelevant as follows [6].

Definition 2 (Strong Relevance, Weak Relevance, and Irrelevance). Given nonempty feature set C and decision class d, for each $f \in C$,

— f is strongly relevant to d, iff $\forall S \subseteq C \backslash \{f\}$ s.t. $P(d|S) \neq P(d|S, f)$.
— f is weakly relevant to d, iff it is not strongly relevant, and $\exists S \subset C \backslash \{f\}$ s.t. $P(d|S) \neq P(d|S, f)$.
— f is irrelevant to d, iff it is neither strongly nor weakly relevant, and $\forall S \subseteq C \backslash \{f\}$ s.t. $P(d|S) = P(d|S, f)$.

where $P(d|S)$ denotes the posterior probability of d condition on S.

Based on Markov blankets, Yu and Liu [31] further divided weakly relevant features into redundant and non-redundant features. Ideally, feature selection aims at selecting all strongly relevant and weakly non-redundant features. However, due to the curse of dimensionality, it is impossible to apply these definitions directly. Thus, a commonly used method is to approximate the relationship through specific feature membership measurements.

Suppose $\gamma_f(d) \in [0, 1]$ denotes the membership grade between feature f and decision class d. With a pair of thresholds α and β ($0 < \alpha < \beta < 1$), we can classify features into three disjoint regions, as shown in Figure 1. Specifically, we discriminate different feature relationships via membership grade as follows:

Definition 3. Given nonempty feature set C and decision class d, for each $f \in C$, $\gamma_f(d) \in [0, 1]$ denotes the membership grade between f and d. With a pair of thresholds α and β ($0 < \alpha < \beta < 1$), we consider:

— f is strongly relevant to d, if $\beta \leq \gamma_f(d) \leq 1$;
— f is weakly relevant to d, if $\alpha < \gamma_f(d) < \beta$.
— f is irrelevant to d, if $0 \leq \gamma_f(d) \leq \alpha$.

Thus, in terms of Definition 3, we can classify each new arriving streaming feature based on a specific feature membership metric.

In general, selecting strongly relevant and weakly non-redundant features is "low risk" during feature selection. On the contrary, selecting irrelevant and redundant features will bring high risks to the final performance. In this article, we use $R(f) = 1 - \gamma_f(d)$ to measure the risk of each streaming feature. Thus, for streaming feature selection, we aim at maximizing the overall membership degrees as follows:

$$Max \quad \sum_{f_t \in F} \gamma_{f_t}(d). \tag{2}$$

3.2 Our New Framework

During online streaming feature selection, the discarded features cannot be used and selected again. Thus, for weakly relevant features, there are considerable risks in making the decision (selecting or discarding) immediately. Three-way decision is a philosophy of thinking in threes, a methodology of working with threes, and a mechanism of processing through threes [24]. Furthermore, Yao [25] proposes the **trisecting-acting-outcome** (**TAO**) model of three-way decision as (1) to divide the whole into three parts, (2) to devise strategies to process the three parts, and (3) to optimize a desirable outcome. Inspired by the idea of "Thinking-in-Threes" and the superiority of three-way decision [23], we make one of the three decisions (selecting, discarding, and delaying) for each new arriving feature, as shown in Figure 2.

In general, for each new arriving feature f_t, if f_t is a strongly relevant feature, we add it into the candidate feature subset S_C; If f_t is an irrelevant feature, we discard it directly; If f_t is a weakly relevant feature, we add it into the undetermined feature subset S_U, and wait for more information to make the decision. Thus, our new online streaming feature selection framework can achieve "low risk" for each new arriving feature with these three strategies.

As shown in Figure 2, there are three main issues that need to be solved during online streaming feature selection: (1) how to choose proper thresholds of α and β that can decrease the decision risk for each new arriving feature; (2) how to remove redundancy for the accepted features in S_C; (3) how to deal with the features in the undetermined feature subset S_U.

In this article, we use **Normalized Mutual Information**(**NMI**) to calculate the membership score between discrete features as

$$NMI(X,Y) = \frac{2MI(X,Y)}{H(X)+H(Y)}, \tag{3}$$

where $MI(X,Y)$ denotes the mutual information between X and Y, H(X) and H(Y) denote the entropy of X and Y, respectively [18]. Thus, $NMI(f,d)$ is the membership score between discrete feature f and the decision class d, where $0 \leq NMI(f;d) \leq 1$. For features with continuous values, we adopt the best-known measure of Fisher's Z-test [16] to calculate the membership scores. In a Gaussian distribution $Normal(\mu, \Sigma)$, the population partial correlation $P(f_i, Y|S)$ between feature f_i and the feature Y given a feature subset S is calculated as follows:

$$P(f_i, Y|S) = \frac{-((\sum_{f_i YS})^{-1})_{f_i Y}}{((\sum_{f_i YS})^{-1})_{f_i f_i}((\sum_{f_i YS})^{-1})_{YY}}. \tag{4}$$

In Fisher's Z-test, under the null hypothesis of conditional independence between f_i and Y given S, $P(f_i, Y|S) = 0$. With the given significance level α and the p-value returned by Fisher's Z-test p, under the null hypothesis of the conditional independence, if $p > \alpha$, f_i, and Y are uncorrelated; otherwise, if $p \leq \alpha$, f_i, and Y are correlated to each other. For simplicity, we use $I(f;d)$ to denote the membership degree of $NMI(f;d)$ for discrete feature f in the next.

3.2.1 Thresholds Update. Assume that the data in the target dataset obey a normal distribution, and the streaming features arrive at random. Then, the membership scores of all features in the whole feature space should obey a normal distribution too, where μ and σ denote the mean value and the standard deviation, as shown in Figure 3.

We use NMI to calculate the value of each feature in Datasets SRBCT, LYMPHOMA, LEUKEMIA, and BREAST (shown in Table 1), and divide the value range of the entire feature space into 50 equal intervals. Figure 4 shows the distribution of the NMI on these four datasets, where each bar denotes the number of features in the value range. The membership scores NMI obey a normal on these datasets.

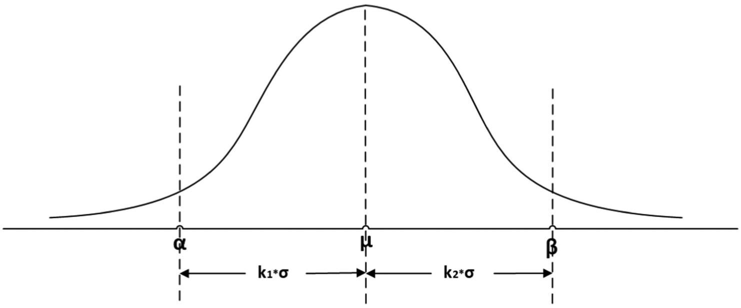
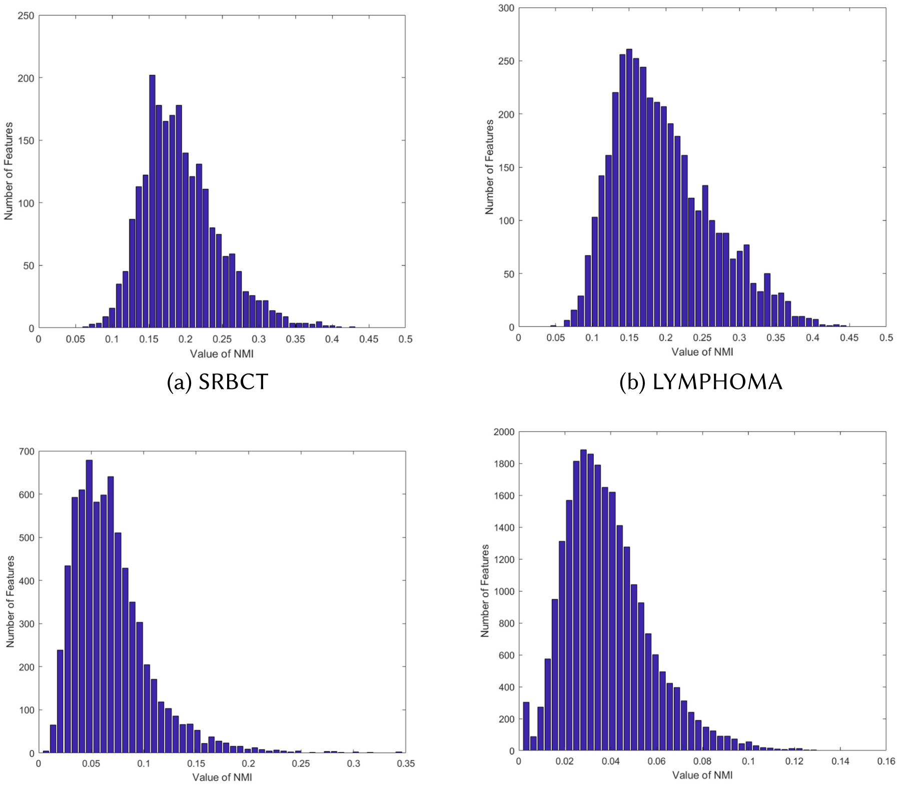


Fig. 3. A normal distribution. With different k_1 and k_2, we can make the selected features in the most informative regions statistically.



(a) SRBCT

(b) LYMPHOMA

(c) LEUKEMIA

(d) BREAST

Fig. 4. The distribution of NMI on Datasets SRBCT, LYMPHOMA, LEUKEMIA, and BREAST. The NMI membership scores on features obey a normal distribution on these datasets.

We choose the values of α and β as $\mu + k * \sigma$, where $k = \pm1, \pm2, \pm3$ denote the middle region is about 68%, 95%, and 99.7% [26]. In other words, without considering redundancy, for streaming feature f_t, if $I(f_t; d) > (\mu + 2 * \sigma)$, f_t is the 2.5% most informative features and the risk of selecting f_t will be lower than the other 97.5% features.

We cannot know the mean value and standard deviation of the whole feature space for online streaming feature selection before learning. Thus, without the information of the entire feature space, these two thresholds (α, β) cannot be fixed in advance. However, we can update μ and σ using each new arriving feature.

THEOREM 1. *At timestamp $t-1$, suppose the mean value is u_{t-1} and the standard deviation is σ_{t-1}. For the new arriving feature f_t at timestamp t, suppose the membership degree is γ_t. Then we can update the mean value and the standard deviation as follows:*

$$\mu_t = \mu_{t-1} + \frac{\gamma_t - \mu_{t-1}}{t}, \tag{5}$$

$$\sigma_t = \sqrt{\frac{(t-2) * \sigma_{t-1}^2 + (\gamma_t - \mu_{t-1})(\gamma_t - \mu_t)}{t-1}}. \tag{6}$$

PROOF 1. $\mu_t = \frac{\sum_{i=1}^{t} \gamma_i}{t} = \frac{\sum_{i=1}^{t-1} \gamma_i + \gamma_t}{t} = \frac{\mu_{t-1} * (t-1) + \gamma_t}{t} = \mu_{t-1} + \frac{\gamma_t - \mu_{t-1}}{t}$.

Suppose $F_t = \sum_{i=1}^{t} (\gamma_i - \mu_t)^2$, then $\sigma_{t-1}^2 = \frac{\sum_{i=1}^{t-1} (\gamma_i - \mu_{t-1})^2}{t-2} = \frac{F_{t-1}}{t-2}$.

$F_n - F_{n-1} = \sum_{i=1}^{t} (\gamma_i - \mu_t)^2 - \sum_{i=1}^{t-1} (\gamma_i - \mu_{t-1})^2$
$= \sum_{i=1}^{t} (\gamma_i - \mu_{t-1} + \mu_{t-1} - \mu_t)^2 - \sum_{i=1}^{t-1} (\gamma_i - \mu_{t-1})^2$
$= (\gamma_t - \mu_{t-1})^2 + \sum_{i=1}^{t-1} (\gamma_i - \mu_{t-1})^2 + 2(\mu_{t-1} - \mu_t) \sum_{i=1}^{t} (\gamma_i - \mu_{t-1}) + \sum_{i=1}^{t} (\mu_{t-1} - \mu_t)^2 - \sum_{i=1}^{t-1} (\gamma_i - \mu_{t-1})^2$
$= (\gamma_t - \mu_{t-1})^2 + 2(\mu_{t-1} - \mu_t)(n * \mu_t - n * \mu_{t-1}) + n * (\mu_{t-1} - \mu_t)^2$.

Substituting Equation (1) and simplification can be obtained: $F_n - F_{n-1} = (\gamma_t - \mu_{t-1})(\gamma_t - \mu_t)$. Thus, $\sigma_t = \sqrt{\frac{F_t}{t-1}} = \sqrt{\frac{(t-2) * \sigma_{t-1}^2 + (\gamma_t - \mu_{t-1})(\gamma_t - \mu_t)}{t-1}}$.

Thus, in terms of Equations (5) and (6), we can dynamically update μ and σ during the online streaming feature selection with each new arriving feature f_t. Meanwhile, we dynamically update the values of α and β as $\alpha = \mu + k_1 * \sigma$, and $\beta = \mu + k_2 * \sigma$, respectively. Therefore, with different values of k_1 and k_2, we can make the number of selected features scalable to the number of dimensions for different datasets.

3.2.2 *Redundancy Analysis.*

For high-dimensional datasets, there always contain a lot of redundant features. Thus, it is necessary to analyze the redundancy for the selected features. In terms of information theory, we can analyze the redundancy among three features. For more than three features, it is impossible to do the redundancy analysis because of the exponentially increasing complexity. Specifically, for features f_1, f_2, and decision class d, if

$$I(\{f_1, f_2\}; d) < I(f_1; d) + I(f_2; d), \tag{7}$$

we consider that there is a **redundancy** between f_1 and f_2 on d. In other words, if the information of $\{f_1, f_2\}$ is less than the sum of each feature, f_1 and f_2 must contain some common information.

THEOREM 2. *If $I(f_2; d|f_1) < I(f_2; d)$ or $I(f_1; d|f_2) < I(f_1; d)$, then f_1 and f_2 are redundant features on d.*

PROOF 2. $I(\{f_1, f_2\}; d) = I(f_1; d) + I(f_2; d|f_1) = I(f_2; d) + I(f_1; d|f_2)$. If $I(f_2; d|f_1) < I(f_2; d)$ or $I(f_1; d|f_2) < I(f_1; d)$, $I(\{f_1, f_2\}; d) < I(f_1; d) + I(f_2; d)$. Thus, f_1 and f_2 are redundant features on d.

On the condition of f_2, if the information between f_1 and d decreases, there must exist redundancy between f_1 and f_2 on d. Combined with the dynamic threshold β, we check the redundancy between two features $(f_1, f_2 \in S_C)$ in the candidate feature subset if

$$I(\{f_1, f_2\}; d) < 2\beta. \tag{8}$$

Theorem 3. *If $f_1, f_2 \in S_C$, $I(\{f_1, f_2\}; d) < 2\beta$, then $I(f_2; d|f_1) < \beta$ and $I(f_1; d|f_2) < \beta$.*

Proof 3. $I(\{f_1, f_2\}; d) = I(f_1; d) + I(f_2; d|f_1) = I(f_2; d) + I(f_1; d|f_2) < 2\beta$. For $f_1, f_2 \in S_C$, $I(f_1; d) > \beta$ and $I(f_2; d) > \beta$. Thus, $I(f_2; d|f_1) < \beta$ and $I(f_1; d|f_2) < \beta$.

In other words, for two features f_1, f_2 in the candidate feature subset S_C, the individual information of both two features is bigger than β. However, if the combined information of these two features is smaller than $2 * \beta$, which means on the condition of one feature (e.g., f_1), the information of the other feature ($I(f_2; d|f_1)$) will decrease and be smaller than β. Thus, we should remove one of these two features (the smaller one between $I(f_1; d)$ and $I(f_2; d)$) from the candidate feature subset.

Once our new framework selects a new streaming feature during online streaming feature selection, we can check the redundancy between this new feature and the currently selected features. For example, suppose the size of the currently selected feature subset is $|S_C|$, then the time complexity of redundancy analysis is $O(|S_C|^2/2)$. In the previous section, we analyzed that the size of S_C is scalable to the number of dimensions by the dynamical parameter adjustment.

3.2.3 Uncertainty Analysis. If the membership degree of a new arriving feature is between α and β, it will be added into the undetermined feature subset S_U and wait for more information.

At timestamp i, suppose the new arriving feature is f_i, and $\alpha < I(f_i; d) < \beta$. Thus, f_i will be added into S_U. If there exists a feature $f_j (f_j \in S_U)$ that makes

$$I(\{f_i, f_j\}; d) \geq 2\beta, \tag{9}$$

then we can select both f_i and f_j into the candidate feature subset S_C. In other words, if the combined information of two features in the undetermined feature subset is bigger than 2β, both these two features will be considered as candidate features.

Theorem 4. *If $f_1, f_2 \in S_U$, $I(\{f_1, f_2\}; d) > 2\beta$, then $I(f_2; d|f_1) > I(f_2; d)$ and $I(f_1; d|f_2) > I(f_1; d)$.*

Proof 4. $I(\{f_1, f_2\}; d) = I(f_1; d) + I(f_2; d|f_1) = I(f_2; d) + I(f_1; d|f_2) > 2\beta$. For $f_1, f_2 \in S_U$, $I(f_1; d) < \beta$ and $I(f_2; d) < \beta$. Then, $I(f_2; d|f_1) > \beta > I(f_2; d)$ and $I(f_1; d|f_2) > \beta > I(f_1; d)$.

In other words, for two features f_1, f_2 in the undetermined feature subset S_U, the individual information of both two features is smaller than β. However, if the combined information of these two features is bigger than $2 * \beta$, which means on the condition of one feature(e.g., f_1), the information of the other feature ($I(f_2; d|f_1)$) will increase and be bigger than β. Then, both these two features can be moved into the candidate feature subset.

There may be many noncommitment features for high-dimensional datasets, and we cannot keep these features all the time. Thus, we should flush S_U regularly when the size of S_U reaches the threshold N_{S_U}. In other words, for features in S_U that cannot satisfy the Equation (9), they will be discarded directly during the feature subset flushing. The time complexity of undetermined analysis is $O(N_{S_U}^2/2)$.

3.2.4 The Proposed Framework. Based on these three solutions mentioned above, we propose a new scalable online streaming feature selection framework via dynamic decision, named OSSFS-DD, as shown in Algorithm 1.

Specifically, at timestamp t, OSSFS-DD gets a new streaming feature f_t. In Step 4, OSSFS-DD calculates the membership degree $\gamma_{f_t}(d)$, and updates the values of μ, σ in terms of Equations (5) and (6). Then OSSFS-DD gets the dynamical threshold values of α and β in Step 5. For $\gamma_{f_t}(d)$ and (α, β), there are three different processings. (1) If $\gamma_{f_t}(d) \leq \alpha$, f_t will be discarded directly in Step 7. (2) If $\gamma_{f_t}(d) \geq \beta$, f_t will be added into the candidate feature subset S_C in Step 9. Meanwhile, OSSFS-DD removes redundant features in S_C that satisfy Equation (8). (3) If $\alpha < \gamma_{f_t}(d) < \beta$, f_t will

ALGORITHM 1: The OSSFS-DD framework

```
Require:
    Input: decision class $d$;
    Parameters: $k_1, k_2, N$
Ensure:
    $S_C$: the selected feature subset;
1:  Initialization: $\mu = \sigma = \alpha = \beta = 0, S_C = S_U = \{\}$
2:  Repeat
3:      get a new streaming feature $f_t$ at timestamp $t$;
4:      calculate $\gamma_{f_t}(d)$, and update $\mu, \sigma$ in terms of Equations (5) and (6);
5:      $\alpha = \mu + k_1 * \sigma, \beta = \mu + k_2 * \sigma$;
6:      IF $\gamma_{f_t}(d) \leq \alpha$
7:          discard $f_t$;
8:      ELSE IF $\gamma_{f_t}(d) \geq \beta$
9:          $S_C = S_C \cup \{f_t\}$;
10:         discard redundant features in $S_C$ in terms of Equation (8);
11:     ELSE
12:         $S_U = S_U \cup \{f_t\}$;
13:         IF $|S_U| == N$
14:             move features from $S_U$ into $S_C$ in terms of Equation (9), empty $S_U$;
15:         END
16:     END
17: Until no features are available;
18: Output: selected feature subset $S_C$
```

be added into the undetermined feature subset S_U in Step 12. Besides, if the undetermined feature subset S_U is full ($|S_U| == N$), OSSFS-DD checks whether there exist some features in S_U that satisfy Equation (9) and moves them into the candidate feature subset S_C. For the other features in S_U, there will be discarded directly. When there are no more features to arrive, OSSFS-DD return the final selected features in S_C.

3.3 Algorithm Analysis

Suppose the total number of features is m for the target dataset. Then, as a scalable online streaming feature selection framework, the running time and number of selected features should be linear or sublinear to m.

On running time, OSSFS-DD first calculates the membership degree between the new arriving feature and class labels and then uses the calculation to update the statistical information and compare it with the dynamically updated thresholds. The time complexity is $O(1)$ from Step 1 to Step 3. Then, OSSFS-DD compares the membership degree with the dynamical parameter values from Step 4 to Step 14. Step 8 removes the redundant features from the candidate feature subset, and the time complexity is $O(|S_C|^2/2)$. Step 10 to Step 13, OSSFS-DD checks whether some features can be moved from the undetermined feature subset into the candidate feature subset when the undetermined feature subset S_U is full. The time complexity of uncertainty analysis is $O(|S_U|^2/2)$.

Because of the dynamic threshold adjustment, the number of selected features can be statistically controlled within a certain range. Suppose we set $\alpha = \mu + 2 * \sigma$ and $\beta = \mu + 3 * \sigma$ for OSSFS-DD. Then, statistically, the maximum size of S_C is about $0.15\% * m$, and the maximum size of S_U is around $2.35\% * m$. Then, the maximum number of selected features is $2.5\% * m$ and the time

Table 1. Real-world Datasets

Index	Data Set	Instances	Features	Classes	Feature Type
1	SRBCT	63	2,308	4	Real
2	LYMPHOMA	62	4,026	3	Real
3	PROSTATE	102	6,033	2	Real
4	LEUKEMIA	72	7,129	2	Real
5	DLBCL	77	7,129	2	Integer
6	ARCENE	200	10,000	2	Integer
7	DEXTER	600	20,000	2	Integer
8	BREAST	97	24,481	2	Real
9	MADELON	2,600	500	2	Integer
10	GINA	3,468	970	2	Integer
11	GISETTE	7,000	5,000	2	Integer

complexity of OSSFS-DD is about $0.03\% * m^2$. Thus, OSSFS-DD is very efficient on running time and scalable on the number of selected features.

4 EXPERIMENTS

4.1 Experiment Setup

4.1.1 Datasets. This section applies the proposed online streaming feature selection method and competing algorithms on eleven real-world datasets [22, 30],[1] as shown in Table 1.

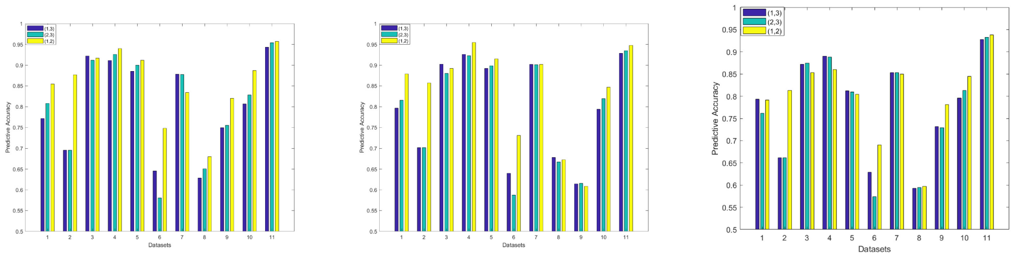
4.1.2 Evaluation Metrics. We use three basic classifiers, KNN (k = 5), SVM(with the linear kernel) and CART in Matlab, to evaluate a selected feature subset in our experiments. We perform 5-fold cross-validation on each data set where feature selection is training on 4/5 data samples and testing on the rest 1/5 data. All competing algorithms use the same training and testing data for each fold. The order of streaming features is random for each dataset. We run each dataset ten times and report the average prediction accuracy, running time, and the mean number of selected features.

To further analyze the performance of OSSFS-DD against its rivals, we conduct the Friedman test at a 95% significance level under the null-hypothesis to validate whether OSSFS-DD and its rivals have a significant difference. If the null-hypothesis at the Friedman test is rejected, we proceed with the Nemenyi test as a post-hoc test [1].

4.1.3 Comparing Algorithms. We compare OSSFS-DD with seven state-of-the-art streaming feature selection methods, including: Alpha-investing [33], Fast-OSFS [21], SAOLA [29], OSFSMI [17], GFSSF [7], OFS-Density [35] and OFS-A3M [36]. The significance level α is set to 0.01 for Fast-OSFS, OSFSMI, and SAOLA. For Alpha-investing, the parameters are set to the values used in [33]. All these algorithms mentioned above are implemented in MATLAB [28].[2]

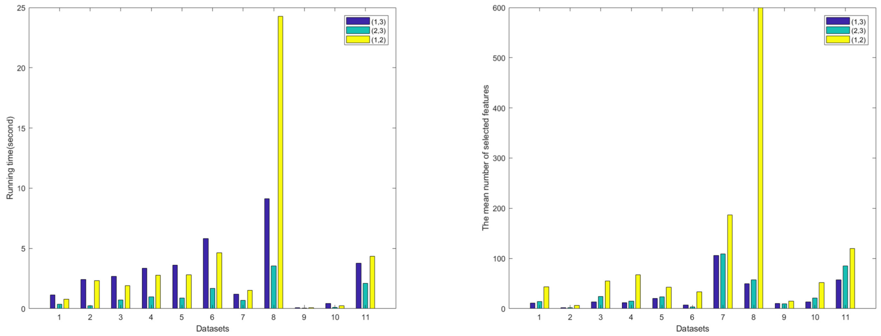
4.1.4 Computational Device. All experimental results are conducted on a PC with AMD 3700X, 3.6 GHz CPU, and 32 GB memory.

[1] Public available at http://www.cs.binghamton.edu/~lyu/KDD08/data/, and http://archive.ics.uci.edu/ml/index.php.
[2] Public available at https://github.com/kuiy/LOFS, and https://github.com/doodzhou/OSFS.



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

Fig. 5. Predictive accuracy on KNN and SVM varying with different values of (k_1, k_2).



(a) Running Time(seconds) (b) Mean number of selected features

Fig. 6. Running time and mean number of selected features varying with different values of (k_1, k_2).

4.2 Parameter Analysis

In Section 3.2.1, we have analyzed that the values of α and β can be chosen as $\mu \pm k{*}\sigma$. Statistically, there are three cases of k that are often be used, where $k = 1, 2, 3$ denote the middle region is about 68%, 95%, and 99.7%, respectively. To reduce the running time and select a more compact feature subset, we specify $\alpha = \mu + k_1 * \sigma$ and $\beta = \mu + k_2 * \sigma$. Meanwhile, we only need to analyze three pairs of parameters $(k1, k2)$ as $(1, 3)$, $(2, 3)$, and $(1, 2)$, where $k1$ must be smaller than $k2$. Besides, N_{S_U} is the maximum size of the undetermined subset, and it has little effect on the results of our algorithm, as long as its value is not too small. Therefore, we set $N_{S_U} = 100$ for OSSFS-DD as an empirical value in the experiments. In general, it does not need the information of the whole feature space to set the parameters for our new algorithm. The predictive accuracy on KNN, SVM and CART varying with different values of (k_1, k_2) on these datasets can be seen as Figure 5. The running time and the mean number of selected features varying with different parameter values can be seen as Figure 6.

The p-values of the Friedman test on KNN, SVM, CART, running time, and mean number of selected features are 0.0027, 0.0164, 0.6909, and 1.8693e-08, 4.9013e-09, respectively. Thus, there is a significant difference among these three cases on the predictive accuracy with KNN and SVM, running time, and mean number of selected features. Meanwhile, there is no significant difference among these three cases on predictive accuracy with CART. According to the Nemenyi test, the value of **critical difference** (**CD**) is 0.9980. We list the average ranks in Table 2.

From Figure 5, Figure 6, and Table 2, we have the following observations:

Table 2. The Average Ranks Varying with Different Values of k_1, k_2

	$(k_1, k_2) = (1, 3)$	$(k_1, k_2) = (2, 3)$	$(k_1, k_2) = (1, 2)$
KNN	2.5000	2.2273	1.2727
SVM	2.2273	2.4091	1.3636
CART	2.0000	2.1818	1.8182
Running Time	2.7273	1.0000	2.2727
Mean Number of Selected Features	1.2273	1.7727	3.0000

- On predictive accuracy, according to the statistical test, $(k_1, k_2) = (1, 2)$ gets the best performance in cases of KNN, SVM, and CART. There is a significant difference between $(k_1, k_2) = (1, 2)$ and $(k_1, k_2) = (1, 3)(2, 3)$ on predictive accuracy with KNN and SVM. Meanwhile, there is no significant difference between $(k_1, k_2) = (1, 3)$ and $(k_1, k_2) = (2, 3)$. The bigger value of k_2, the fewer number of features can be make the decision "acceptance" directly. If $k_2 = 3$, ideally and statistically, only 0.15% of features can be added into the candidate feature subset directly. For real-world applications, this ratio may be much lower. Thus, $(k_1, k_2) = (1, 2)$ gets higher accuracy than the other two cases, especially on datasets LYMPHOMA(2) and ARCENE(6).
- On running time, a lower value of k_1 means more features to be considered in the undetermined analysis, while a lower value of k_2 indicates more features in redundancy analysis. Both two smaller values will lead to more time-consuming. In Figure 6(a), $(k_1, k_2) = (1, 3)$ spends more running time than $(k_1, k_2) = (1, 2)$ on most of these datasets for the big size of undetermined feature subset. However, on dataset BREAST(8), $(k_1, k_2) = (1, 2)$ consumes much more time than $(k_1, k_2) = (1, 3)$. For dataset BREAST, the NMI values of the features are very close, leading to a big candidate feature subset and the frequent refreshing of the undetermined feature subset. Both these two analyses will consume much running time.
- On the number of selected features, $(k_1, k_2) = (1, 2)$ selects more features than the other two cases on all these datasets. Statistically, $(k_1, k_2) = (1, 2)$ will select 2.5% features into the candidate feature subset and 13.5% features into the undetermined feature subset. However, due to the redundancy analysis and undetermined analysis, the final number of selected features will be smaller than 16% of the whole feature space.

In general, $(k_1, k_2) = (1, 2)$ considers more features than the other two cases and gets better performs on predictive accuracy. Meanwhile, $(k_1, k_2) = (1, 2)$ consumes more running time than the other two cases on average. In the next experiments, we specific $(k_1, k_2) = (1, 2)$.

4.3 OSSFS-DD vs. State-of-the-art Online Streaming Feature Selection Methods

Tables 3–5 summarize the predictive accuracy of OSSFS-DD against the other seven algorithms using the KNN, SVM, and CART classifiers. Tables 6 and 7 show the running time and the mean number of selected features, respectively. The p-values of the Friedman test on KNN, SVM, CART, running time, and the mean number of selected features are 5.5796e−07, 6.5954e−05, 0.0012, 1.1942e−09, and 4.5940e−06, respectively. Thus, there is a significant difference among these competing algorithms respectively on predictive accuracy, running time, and the number of selected features. According to the Nemenyi test, the value of CD is 3.1684. Figure 7 shows the statistical test of these competing algorithms in cases of KNN, SVM, and CART.

From Tables 3–7 and Figure 7, we have the following observations:

Table 3. Predictive Accuracy Using KNN as the Classifier

Data Set	OSSFS-DD	α-investing	Fast-OSFS	SAOLA	OSFSMI	GFSSF	OFS-A3M	OFS-Density
SRBCT	0.8972	0.3406	0.8483	0.865	0.8133	0.4566	0.8895	**0.9308**
LYMPHOMA	0.925	0.625	0.9417	0.9167	0.8833	0.7083	**0.9667**	0.95
PROSTATE	**0.93**	0.57	0.895	0.865	0.91	0.615	0.87	0.92
LEUKEMIA	**0.95**	0.7286	0.9357	0.9286	0.9	0.6321	0.9143	0.9321
DLBCL	**0.93**	0.75	0.8267	0.91	0.78	0.74	0.8767	0.8167
ARCENE	0.7638	0.6625	0.6875	0.6588	0.665	0.63	0.7913	**0.795**
DEXTER	0.8225	0.8675	0.7183	**0.835**	0.4983	0.66	0.7167	0.7967
BREAST	**0.7**	0.5421	0.6605	0.6737	0.6711	0.5658	0.6053	0.5895
MADELON	**0.8063**	0.647	0.5626	0.5595	0.6831	0.5038	0.5543	0.5171
GINA	0.8573	**0.9227**	0.8707	0.8067	0.786	0.6524	0.8441	0.816
GISETTE	0.8628	0.8397	0.8665	0.8391	0.5838	0.5942	0.8653	**0.8757**
AVG.	**0.8586**	0.6814	0.8012	0.8052	0.7430	0.6143	0.8085	0.8126
AVG. RANKS	**2.1818**	5.6364	3.5455	4.4545	5.4545	7.4545	3.9091	3.3636

The best results are highlighted in bold face in the tables.

Table 4. Predictive Accuracy Using SVM as the Classifier

Data Set	OSSFS-DD	α-investing	Fast-OSFS	SAOLA	OSFSMI	GFSSF	OFS-A3M	OFS-Density
SRBCT	**0.9769**	0.2608	0.8881	0.9098	0.7825	0.4678	0.928	0.9231
LYMPHOMA	0.9417	0.6833	0.9083	0.9333	0.9167	0.7583	**0.9583**	**0.9583**
PROSTATE	0.9	0.59	0.885	0.83	0.91	0.59	0.86	**0.92**
LEUKEMIA	**0.95**	0.7679	0.9429	0.9321	0.9036	0.6536	0.9393	0.925
DLBCL	**0.9467**	0.78	0.8167	0.9067	0.8333	0.7633	0.8767	0.83
ARCENE	0.7513	0.6875	0.6775	0.6413	0.6775	0.64	**0.7787**	0.7713
DEXTER	**0.8933**	0.8642	0.6958	0.8483	0.5567	0.6825	0.7517	0.7708
BREAST	0.6632	0.6	**0.7053**	0.6658	0.6921	0.5816	0.5921	0.5711
MADELON	0.612	0.6128	0.6117	0.6009	**0.6158**	0.5092	0.5192	0.5347
GINA	0.8286	**0.87**	0.8441	0.8069	0.7748	0.6824	0.8134	0.8058
GISETTE	0.8995	**0.9277**	0.8204	0.841	0.8804	0.6632	0.8615	0.7776
AVG.	**0.8512**	0.69492	0.7996	0.8105	0.7766	0.6356	0.8071	0.7988
AVG. RANKS	**2.2727**	4.7727	4.3182	4.3636	4.5000	7.5909	3.7727	4.4091

The best results are highlighted in bold face in the tables.

Table 5. Predictive Accuracy Using CART as the Classifier

Data Set	OSSFS-DD	α-investing	Fast-OSFS	SAOLA	OSFSMI	GFSSF	OFS-A3M	OFS-Density
SRBCT	0.8601	0.293	0.8531	0.8147	0.8021	0.4448	**0.8895**	0.8804
LYMPHOMA	0.9083	0.6	0.8083	**0.925**	**0.925**	0.725	0.9083	0.9083
PROSTATE	0.875	0.59	0.89	0.855	**0.9**	0.565	0.85	0.88
LEUKEMIA	0.8393	0.7286	**0.9107**	0.8786	0.875	0.5893	0.9071	0.8929
DLBCL	**0.8133**	0.7033	0.79	0.8033	0.7933	0.6467	0.76	0.7867
ARCENE	0.7012	0.6363	0.6575	0.6125	0.6588	0.665	**0.7438**	0.725
DEXTER	0.8358	**0.8567**	0.7242	0.8242	0.8192	0.725	0.7442	0.7933
BREAST	0.6158	0.5711	**0.6711**	0.6132	0.6342	0.5342	0.6	0.5474
MADELON	**0.7799**	0.6276	0.5464	0.5343	0.65	0.5057	0.5435	0.512
GINA	0.8178	**0.867**	0.8394	0.7858	0.7606	0.6476	0.8066	0.7881
GISETTE	0.9287	**0.9341**	0.9029	0.8862	0.9209	0.7225	0.899	0.8784
AVG.	**0.8159**	0.6734	0.7812	0.7757	0.7944	0.6155	0.7865	0.7811
AVG. RANKS	**2.9091**	5.0909	3.8182	4.5909	3.5909	7.3636	4.0909	4.5455

The best results are highlighted in bold face in the tables.

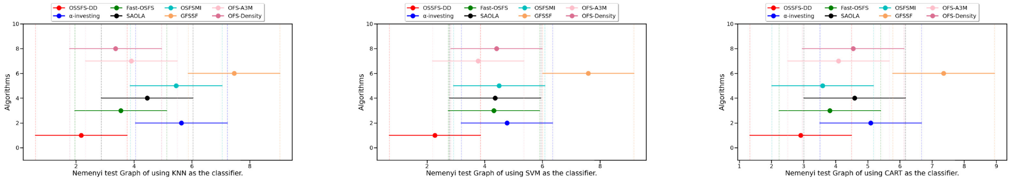
Table 6. Running Time (Seconds)

Data Set	OSSFS-DD	α-investing	Fast-OSFS	SAOLA	OSFSMI	GFSSF	OFS-A3M	OFS-Density
SRBCT	0.7607	**0.0392**	0.1866	0.2389	0.3017	0.2303	0.631	0.5671
LYMPHOMA	2.0213	**0.1633**	0.4413	0.829	1.639	0.7045	1.0567	1.1289
PROSTATE	1.9229	**0.3202**	0.462	0.6076	1.1464	1.645	2.9848	2.7309
LEUKEMIA	2.8202	**0.4374**	0.535	0.7767	1.1821	1.8329	2.1253	1.8987
DLBCL	2.8465	**0.4767**	0.5105	0.6631	1.1887	1.6636	2.2531	2.0382
ARCENE	4.6686	**0.9416**	0.9532	1.4379	3.0623	5.5925	12.2781	12.9937
DEXTER	**0.3128**	3.5849	14.1759	0.351	2,844.8914	0.5006	166.5361	165.5445
BREAST	23.8249	3.5185	**1.4441**	1.684	1.97	12.071	11.8465	13.3072
MADELON	0.0666	0.0488	0.0677	0.0394	**0.0422**	1.5712	175.7092	168.4572
GINA	0.455	3.0314	24.2839	**0.2155**	0.4018	10.6493	387.7432	434.7665
GISETTE	4.0412	171.79	8,340.5843	**1.4731**	1,568.2346	150.5005	8,963.0558	9,340.2829
AVG.	3.97	16.75	762.14	0.75	402.18	16.99	884.20	922.15
AVG. RANKS	5.5455	2.2727	3.1818	2.4545	4.3636	4.5455	6.8182	6.8182

The best results are highlighted in bold face in the tables.

Table 7. The Mean Number of Selected Features

Data Set	OSSFS-DD	α-investing	Fast-OSFS	SAOLA	OSFSMI	GFSSF	OFS-A3M	OFS-Density
SRBCT	62.1	1.3	4.4	16.7	7.3	3	8.5	6.4
LYMPHOMA	6.3	1	4.9	22.1	8.2	2.8	3.9	10.2
PROSTATE	56.3	2.3	2.9	10.7	6.6	3.5	25.7	6.6
LEUKEMIA	73.55	2.55	4.9	21.15	8.65	3	7.45	5.9
DLBCL	40	4.5	4.45	14.9	7.15	3	20.05	4.9
ARCENE	33.65	6	5.65	17.45	8.2	5.15	40.3	33.1
DEXTER	56.7	13.9	2.6	11.3	19,894.9	15.1	14.1	8.8
BREAST	600	5.3	4.6	18.15	7.45	3.2	32.5	12.4
MADELON	15.3	6.1	5.1	7.2	4.55	6.05	3.35	1.5
GINA	25.2	93.35	21.75	10.3	7.9	12.9	21.1	8.25
GISETTE	120.8	364.8	13.1	17.4	2,779.2	28.1	46.9	7.9
AVG.	99.0	45.5	6.7	15.2	2,067.2	7.8	20.3	9.6
AVG. RANKS	7.2727	3.4545	2.8182	5.5455	4.9545	2.8182	5.4545	3.6818



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

Fig. 7. The statistical test graph of these competing algorithms.

- OSSFS-DD vs. Alpha-investing: On predictive accuracy, according to the results of the statistical test, OSSFS-DD performs better than Alpha-investing in cases of KNN, SVM, and CART. Meanwhile, there is a significant difference between OSSFS-DD and Alpha-investing on KNN. Alpha-investing is the fastest among all these competing algorithms on running time, for it does not deal with redundancy between selected features. On the mean number of selected features, Alpha-investing can only select the first one or two features for some datasets, such as SRBCT, LYMPHOMA, and PROSTATE, which leads to the worst predictive accuracy on these datasets among all these competing algorithms. Thus, Alpha-investing cannot handle all different types of datasets. For OSSFS-DD, the number of selected features is scalable to the target datasets, ensuring the stability of performance.

- OSSFS-DD vs. Fast-OSFS: OSSFS-DD gets higher average predictive accuracy and lower average ranks than Fast-OSFS in cases of KNN, SVM, and CART. Thus, OSSFS-DD performs better than Fast-OSFS on predictive accuracy. On running time, Fast-OSFS is faster than OSSFS-DD on most of these datasets. However, on dataset GISETTE, Fast-OSFS spends 8,340 seconds, while OSSFS-DD only needs 4 seconds. Meanwhile, Fast-OSFS selects the fewest features among all competing algorithms on all these competing datasets. Fast-OSFS considers features individually in relevance and redundancy analysis and selects much fewer features on these datasets, leading to some critical information loss. Thus, OSSFS-DD is more stable than Fast-OSFS on running time and number of selected features while performing better on Predictive accuracy.
- OSSFS-DD vs. SAOLA: According to the Nemenyi test, OSSFS-DD performs better than SAOLA on predictive accuracy. On running time, SAOLA is faster than OSSFS-DD on most of these datasets. On the mean number of selected features, SAOLA selects fewer features than OSSFS-DD. SAOLA employs novel online pairwise comparison techniques that only consider the feature relationships between two. However, with the full use of global statistical information, OSSFS-DD can always select the top informative features during streaming feature selection, leading to superior predictive accuracy performance.
- OSSFS-DD vs. OSFSMI: On predictive accuracy, OSSFS-DD performs better than OSFSMI in cases of KNN, SVM, and CART. On running time, OSFSMI spends much more time than OSSFS-DD on average. On datasets DEXTER and GISETTE, OSFSMI spends 2,844 seconds and 1,568 seconds , while OSSFS-DD only needs 0.3 seconds and 4.0 seconds , respectively. On the mean number of selected features, OSFSMI selects 2,067 features on average. On datasets DEXTER and GISETTE, OSFSMI selects 19,894 and 27,79 features while getting worse performance on predictive accuracy than OSSFS-DD. Thus, OSSFS-DD is more stable than OSFSMI and can handle different datasets well.
- OSSFS-DD vs. GFSSF: GFSSF performs the worst on predictive accuracy among all these competing algorithms. Meanwhile, there is a significant difference between OSSFS-DD and GFSSF in cases of KNN, SVM, and CART. Thus, OSSFS-DD significantly performs better than GFSSF. On running time, OSSFS-DD is comparable with GFSSF. On the mean number of selected features, GFSSF selects much fewer features than OSSFS-DD. GFSSF is an information based method that can also be adapted to group streaming feature selection. However, the relevance and redundancy analysis for GFSSF is too strict, leading to missing some important features.
- OSSFS-DD vs. OFS-A3M: According to the average ranks, OSSFS-DD performs better than OFS-A3M in cases of KNN, SVM, and CART. OFS-A3M is a neighborhood Rough Set-based method with a high time complexity for large sample datasets. OSSFS-DD is faster than OFS-A3M on running time, especially on big sample datasets, such as DEXTER, MADELON, GINA, and GISETTE. Meanwhile, OFS-A3M uses the neighborhood information for feature selection that can be significantly affected by the sample distribution. On the number of selected features, OFS-A3M selects fewer features than OSSFS-DD on some datasets.
- OSSFS-DD vs. OFS-Density: OSSFS-DD gets higher predictive accuracy than OFS-Density, with seven of eleven datasets on KNN, eight of eleven datasets on SVM, and six of eleven datasets on CART. Like OFS-A3M, OFS-Density is also a neighborhood Rough Set-based method and has a high time complexity for large sample datasets. On running time, OFS-Density spends much more time than OSSFS-DD on average. On the number of selected features, OFS-Density selects much fewer features than OSSFS-DD. OFS-Density uses

neighborhood density information for feature selection that cannot handle unevenly distributed data sets well.

In sum, OSSFS-DD gets the best performance on predictive accuracy in cases of KNN, SVM, and CART. Meanwhile, OSSFS-DD is more scalable on running time and the number of selected features than other competing algorithms. Due to the dynamic adjustment of thresholds, OSSFS-DD can always select the most informative features and be scalable on the number of selected features.

5 CONCLUSION

This article proposes a new online scalable streaming feature selection framework from the dynamic decision perspective for the first time. Based on the philosophy of "Thinking-in-Threes", we use a pair of dynamically adjust thresholds to classify each new arriving feature into selecting, discarding, and delaying by the global statistical information. Meanwhile, we remove the redundancy for the candidate features and add feature pairs in the undetermined feature subset. Extensive experiments demonstrate the effectiveness and scalability of our new proposed method. This article only checks the redundancy and joint information between two features in redundancy and uncertainty analysis. In future work, we will consider the relationships for triples or more variables. Besides, scholars can try to apply other dynamic decision-making ideas and methods to the online streaming feature selection.

REFERENCES

[1] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1 (2006), 1–30.

[2] S. Eskandari and M. M. Javidi. 2016. Online streaming feature selection using rough sets. *International Journal of Approximate Reasoning* 69, C (2016), 35–57.

[3] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, (2003), 1157–1182.

[4] Bao Qing Hu. 2014. Three-way decisions space and three-way decisions. *Information Sciences* 281 (2014), 21–52.

[5] XueGang Hu, Peng Zhou, PeiPei Li, Jing Wang, and XinDong Wu. 2018. A survey on online feature selection with streaming features. *Frontiers of Computer Science* 12, 3 (2018), 479–493.

[6] Ron Kohavi and George H. John. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97, 1–2 (1997), 273–324.

[7] Hai Guang Li, Xing Dong Wu, Zhao Li, and Wei Ding. 2013. Group feature selection with streaming features. In *Proceedings of the IEEE 13th International Conference on Data Mining*. 1109–1114.

[8] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2018. Feature selection: A data perspective. *Computing Surveys* 50, 6 (2018), 1–45.

[9] Yun Li, Tao Li, and Huan Liu. 2017. Recent advances in feature selection and its applications. *Knowledge and Information Systems* 53, 3 (2017), 551–577.

[10] Yaojin Lin, Qinghua Hu, Jinghua Liu, Jinjin Li, and Xindong Wu. 2017. Streaming feature selection for multilabel learning based on fuzzy mutual information. *IEEE Transactions on Fuzzy Systems* 25, 6 (2017), 1491–1507.

[11] Huan Liu and Hiroshi Motoda. 2007. *Computational Methods of Feature Selection*. Chapman and Hall/CRC Press.

[12] Huan Liu and Lei Yu. 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* 17, 4 (2005), 491–502.

[13] Jinghua Liu, Yaojin Lin, Yuwen Li, Wei Weng, and Shunxiang Wu. 2018. Online multi-label streaming feature selection based on neighborhood rough set. *Pattern Recognition* 84 (2018), 273–287.

[14] Dipanjyoti Paul, Rahul Kumar, Sriparna Saha, and Jimson Mathew. 2021. Multi-objective cuckoo search-based streaming feature selection for multi-label dataset. *ACM Transactions on Knowledge Discovery from Data* 15, 6 (2021), 1–24.

[15] Simon Perkins and James Theiler. 2003. Online feature selection using grafting. In *Proceedings of the 20th International Conference on Machine Learning*. 592–599.

[16] Jose M. Peña. 2008. Learning gaussian graphical models of gene networks with false discovery rate control. In *Proceedings of the European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer, 165–176.

[17] Maryam Rahmaninia and Parham Moradi. 2018. OSFSMI: Online stream feature selection method based on mutual information. *Applied Soft Computing* 68 (2018), 733–746.

[18] C. E. A. Shannon. 2001. A mathematical theory of communication. *ACM Sigmobile Mobile Computing & Communications Review* 5, 1 (2001), 3–55.

[19] Di Wu, Yi He, Xin Luo, Mingsheng Shang, and Xindong Wu. 2019. Online feature selection with capricious streaming features: A general framework. In *Proceedings of the 2019 IEEE International Conference on Big Data*. 683–688.

[20] Di Wu, Yi He, Xin Luo, and MengChu Zhou. 2021. A latent factor analysis-based approach to online sparse streaming feature selection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2021), 1–15. DOI: https://doi.org/10.1109/TSMC.2021.3096065

[21] XinDong Wu, Kui Yu, Wei Ding, Hao Wang, and XingQuan Zhu. 2013. Online feature selection with streaming features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 5 (2013), 1178–1192.

[22] Kun Yang, ZhiPeng Cai, JianZhong Li, and GuoHui Lin. 2006. A stable gene selection in microarray data analysis. *BMC Bioinformatics* 7, 1 (2006), 228.

[23] Yiyu Yao. 2011. The superiority of three-way decisions in probabilistic rough set models. *Information Sciences* 181, 6 (2011), 1080–1096.

[24] Yiyu Yao. 2012. An outline of a theory of three-way decisions. In *Proceedings of the Rough Sets and Current Trends in Computing*. Springer Berlin Heidelberg, 1–17.

[25] Yiyu Yao. 2020. Tri-level thinking: Models of three-way decision. *International Journal of Machine Learning and Cybernetics* 11, 5 (2020), 947–959.

[26] Yiyu Yao and Cong Gao. 2015. Statistical interpretations of three-way decisions. *Lecture Notes in Computer Science* 9436 (2015), 309–320.

[27] Dianlong You, Ruiqi Li, Shunpan Liang, Miaomiao Sun, Xinju Ou, Fuyong Yuan, Limin Shen, and Xindong Wu. 2021. Online causal feature selection for streaming features. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–15. DOI: https://doi.org/10.1109/TNNLS.2021.3105585

[28] Kui Yu, Wei Ding, and XinDong Wu. 2016. LOFS: Library of online streaming feature selection. *Knowledge-Based Systems* 113, (2016), 1–3.

[29] Kui Yu, XinDong Wu, Wei Ding, and Jian Pei. 2016. Scalable and accurate online feature selection for big data. *ACM Transactions on Knowledge Discovery from Data* 11, 2 (2016), 1–39.

[30] Lei Yu, Chris Ding, and Steven Loscalzo. 2008. Stable feature selection via dense feature groups. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 803–811.

[31] Lei Yu and Huan Liu. 2004. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research* 5, 12 (2004), 1205–1224.

[32] Yiwen Zhang, Kaibin Wang, Qiang He, Feifei Chen, Shuiguang Deng, Zibin Zheng, and Yun Yang. 2019. Covering-based web service quality prediction via neighborhood-aware matrix factorization. *IEEE Transactions on Services Computing* 14, 5 (2019), 1333–1344.

[33] Jing Zhou, Dean P. Foster, Robertz A. Stine, and Lyle H. Ungar. 2006. Streamwise feature selection. *Journal of Machine Learning Research* 3, 2 (2006), 1532–4435.

[34] Peng Zhou, Xuegang Hu, Peipei Li, and Xindong Wu. 2017. Online feature selection for high-dimensional class-imbalanced data. *Knowledge-Based Systems* 136 (2017), 187–199.

[35] Peng Zhou, Xuegang Hu, Peipei Li, and Xindong Wu. 2019. OFS-Density: A novel online streaming feature selection method. *Pattern Recognition* 86 (2019), 48–61.

[36] Peng Zhou, Xuegang Hu, Peipei Li, and Xindong Wu. 2019. Online streaming feature selection using adapted neighborhood rough set. *Information Sciences* 481 (2019), 258–279.

[37] Peng Zhou, Peipei Li, Shu Zhao, and Xindong Wu. 2021. Feature interaction for streaming feature selection. *IEEE Transactions on Neural Networks and Learning Systems* 32, 10 (2021), 4691–4702.

Received July 2021; accepted November 2021