Feature Interaction for Streaming Feature Selection

Peng Zhou, Peipei Li, Shu Zhao, and Xindong Wu, *Fellow, IEEE*

***Abstract*— Traditional feature selection methods assume that all data instances and features are known before learning. However, it is not the case in many real-world applications that we are more likely faced with data streams or feature streams or both. Feature streams are defined as features that flow in one by one over time, whereas the number of training examples remains fixed. Existing streaming feature selection methods focus on removing irrelevant and redundant features and selecting the most relevant features, but they ignore the interaction between features. A feature might have little correlation with the target concept by itself, but, when it is combined with some other features, they can be strongly correlated with the target concept. In other words, the interactive features contribute to the target concept as an integer greater than the sum of individuals. Nevertheless, most of the existing streaming feature selection methods treat features individually, but it is necessary to consider the interaction between features. In this article, we focus on the problem of feature interaction in feature streams and propose a new streaming feature selection method that can select features to interact with each other, named Streaming Feature Selection considering Feature Interaction (SFS-FI). With the formal definition of feature interaction, we design a new metric named interaction gain that can measure the interaction degree between the new arriving feature and the selected feature subset. Besides, we analyzed and demonstrated the relationship between feature relevance and feature interaction. Extensive experiments conducted on 14 real-world microarray data sets indicate the efficiency of our new method.**

***Index Terms*— Feature interaction, feature selection, interaction gain, streaming feature selection.**

I. Introduction

THE era of big data is accumulated challenging state-of-the-art machine learning techniques to efficiently produce useful results on such high-dimensional data sets [1]. Feature selection aims to select a minimal size subset of the feature space, which can retain the optimum salient characteristics necessary from the original data sets [2]. With the removal of noisy, irrelevant, and redundant features, machine learning can benefit significantly from using only relevant data, such as better performance, less running time, and improved comprehensibility [3]–[7].

Traditional feature selection methods assume that all features are presented to a learner before learning takes place. Nevertheless, in many real-world applications, not all features can be presented before learning, and we are more likely faced with data streams or feature streams or both [8]–[10]. For example, Twitter produces more than 500 million tweets every day, and a large number of slang words (features) are continuously being generated [11]. Feature streams are defined as features that flow in one by one over time, whereas the number of training examples is fixed [12]. Streaming feature selection that deals with feature streams in an online manner has attracted much attention in recent years [12]–[20]. It presents great advantages when handling extremely high-dimensional data sets, such as low time and space consumption [21]. Two representative methods of them are online streaming feature selection (OSFS) [12] and a Scalable and Accurate Online feature selection Approach (SAOLA) [14]. OSFS selects strongly relevant and nonredundant features on the fly and contains two major steps: online relevance analysis (discarding irrelevant features) and online redundancy analysis (eliminating redundant features). To tackle the challenges in online feature selection from extremely high-dimensional data, SAOLA employs novel online pairwise comparison techniques and maintains a parsimonious model over time in an online manner.

In general, feature selection focuses on removing irrelevant and redundant features from the feature space and selecting the most relevant and informative ones [22]. An important but usually being ignored issue is feature interaction [23]. Interactive features are those that appear to be low relevant or even irrelevant with the class individually, but, when it is combined with other features, it may highly correlate to the class [24]–[26]. In other words, interactive features contribute to the class as an integer greater than the sum of individuals. A special case for this is the XOR problem. More precisely, there are two features f_1 and f_2 and the class c, where $c = f_1 \oplus f_2$ and $\oplus$ represents the XOR logic function. f_1 or f_2 does not carry any information about the class individually, but the two features completely determine the class if they are combined. A more common example is the Monks1 data set from the UCI Machine Learning Repository, which has 432 instances and six category features $a_1, a_2, \ldots, a_6$. The target concept c is defined by $c = (a_1 = a_2) \vee (a_5 = 1)$. We use mutual information (MI) [27] to calculate the information between

Manuscript received May 5, 2019; revised September 17, 2019, February 17, 2020, and June 29, 2020; accepted September 18, 2020. Date of publication October 6, 2020; date of current version October 6, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000901 and in part by the Natural Science Foundation of China under Grant 61906056, Grant 61673152, Grant 61876206, Grant 61976077, and Grant 61876001. *(Corresponding author: Xindong Wu.)*

Peng Zhou and Shu Zhao are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: doodzhou@ahu.edu.cn; zhaoshuzs@ahu.edu.cn).

Peipei Li is with the Department of Computer Science and Technology Engineering, Hefei University of Technology, Hefei 230009, China (e-mail: peipeili@hfut.edu.cn).

Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, Hefei University of Technology, Hefei 230009, China, and also with the Mininglamp Academy of Sciences, Beijing 100084, China (e-mail: wuxindong@mininglamp.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3025922

2162-237X © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

features and get $I(a_1; c) = 0.0753$, $I(a_2; c) = 0.0058$, and $I(a_5; c) = 0.2870$. However, $I(\{a_1, a_2\}; c) = 0.5147 > I(a_1; c) + I(a_2; c)$ and $I(\{a_1, a_2, a_5\}; c) = 1 > I(a_1; c) + I(a_2; c) + I(a_5; c)$, which means that the combination of features a_1, a_2, and a_5 can contribute more information than the sum of each features. Besides, for the real-world applications of high-throughput microarray data, finding interacting features is to search a small number of pertinent genes from hundreds of thousands of ones that reflecting the immune response mechanism involving both antigen presentation and immunoproteasome pathways [24].

For streaming feature selection, we cannot require the domain knowledge in advance and do not know the information of the entire feature space before learning. Most of the existing streaming feature selection methods, such as OSFS and SAOLA mentioned earlier, treat features individually and do not consider the interaction among features. Besides, many effective and efficient learning algorithms assume the independence of features. However, they may fail badly when the degree of feature interaction becomes critical [23].

Motivated by this, we focus on the problem of feature interaction and propose a new streaming feature selection method that can select features to interact with each other, named Streaming Feature Selection considering Feature Interaction (SFS-FI). The main contributions of this article are as follows.

1) We give the formal definition of feature interaction. Meanwhile, we analyze and demonstrate the relationship between feature relevance and feature interaction.
2) We provide a systematic analysis of two-way interaction, three-way interaction, and four-way interaction in feature selection. Meanwhile, we demonstrate that four-way interaction can be converted into the sum of some three-way interactions. Thus, we can approximate a higher-way interaction by considering the combination of all possible three-way interactions.
3) A new metric named interaction gain was proposed, which can measure the interaction degree between the new arriving feature and the selected feature subset. Besides, with the definitions of positive interaction and negative interaction, we present a new streaming feature selection method that can guarantee the selected features relevant to the class and interact with each other.
4) To investigate the effectiveness of our new method, experimental results conducted on 14 real-world microarray data sets indicate that SFS-FI can select relevant and interactive features on the fly.

The rest of this article is organized as follows. In Section II, we describe the related work. In Section III, with the definitions of feature relevance, redundancy, and interaction, we analyze and demonstrate the relationship between feature relevance and feature interaction. In Section IV, we give the problem formalization of streaming feature selection and propose a new algorithm that considers feature interaction during streaming feature selection. Experimental analyses are presented in Section V. Finally, we make a brief conclusion in Section VI.

II. Related Work

Feature selection has been studied for decades and can be broadly classified into three categories: filter, wrapper, and embedded [2].

Filter methods evaluate the feature importance according to certain criteria that are independent of any learning algorithms. For example, by using the concept of distance correlation, Kundu and Mitra [28] developed a novel similarity-based feature selection algorithm that does not need an exhaustive traversal of the search space. Yang *et al.* [29] investigated the incremental perspective for fuzzy rough set-based feature selection, which assumes that data can be presented in the sample subsets one after another. Wrapper methods evaluate the quality of the selected features with a predefined learning algorithm. The article [30] describes a novel wrapper feature selection algorithm for classification problems, which utilizes a genetic algorithm to wrap extreme learning machine to search for the optimum subsets in the huge feature space. Embedded methods perform feature selection in the process of model construction. The representative methods of embedded mode are regularized regression-based feature selection algorithms. Pang *et al.* [31] proposed a novel framework to solve the original $l_{2,0}$-norm constrained sparse regression-based feature selection problem. Besides, Bing *et al.* [32] present a comprehensive survey of the state-of-the-art work on evolutionary computation for feature selection and identified the contributions of these different algorithms.

For traditional feature selection methods, most of them concentrate on removing irrelevant and redundant features or selecting the most relevant features from a given candidate feature space. Among them, there are few works focusing on the feature interaction. More specifically, Jakulin and Bratko [33] focused on the assumption of independence of attributes for many effective and efficient learning algorithms and pointed out that these methods may fail badly when the degree of attribute dependencies becomes critical. They first formally define the degree of interaction between attributes through the deviation of the best possible "voting" classifier from the true relation between the class and the attributes in a domain. Furthermore, they [23] introduced an operational definition of a generalized n-way interaction by highlighting two models: the reductionistic part-to-whole approximation, where the whole model is reconstructed from models of the parts, and the holistic reference model, where the whole is directly modeled. An interaction is deemed significant if these two models are significantly different. Zhao and Liu [24] took up the challenge of feature interaction to design a special data structure for the feature quality evaluation and employ an information-theoretic feature ranking mechanism to efficiently handle feature interaction in the subset selection. Zeng *et al.* [26] proposed a novel feature selection algorithm considering feature interaction. They defined the interaction weight factor that can reflect the information whether a feature is redundant or interactive and designed an Interaction Weight-based Feature Selection (IWFS) algorithm. However, all these methods are designed for traditional feature selection, and they cannot be applied in the streaming feature selection directly.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

Contrary to traditional feature selection, streaming feature selection assumes that the features arriving one by one over time, and we cannot require the whole feature space before learning. There are two major reasons for streaming feature selection: 1) the feature space is unknown or even infinite and 2) the feature space is known, but feature streaming offers many advantages. In order to handle this problem, many research works were proposed in recent years.

More specifically, Perkins and Theiler [34] first considered the problem of online feature selection and proposed the Grafting algorithm based on a stagewise gradient descent approach. Grafting treats feature selection as an integral part of learning a predictor within a regularized framework. Zhou *et al.* [35] proposed two algorithms of information-investing and alpha-investing, based on streamwise regression for online feature selection. Alpha-investing does not need a global model, and it is one of the penalized likelihood ratio methods. Wu *et al.* [12] presented an OSFS framework with two algorithms called OSFS and fast-OSFS. OSFS contains two major steps, including online relevance analysis (discarding irrelevant features) and online redundancy analysis (eliminating redundant features). Yu *et al.* [14] proposed the SAOLA approach for high-dimensional data. SAOLA employs novel online pairwise comparison techniques and maintains a parsimonious model over time in an online manner. Eskandari and Javidi [17] proposed a rough set-based method for OSFS, named OS-NRRSARA-SA. Unlike the classical Rough Set-based attribute reduction methods that only use the information contained in the positive region, OS-NRRSARA-SA considers the boundary and positive regions. Zhou *et al.* [16] proposed a new OSFS method for high-dimensional and class-imbalanced data, called K-OFSD. K-OFSD uses the dependence between condition features and decision classes for feature selection. Zhou *et al.* [18] proposed a new OSFS method OFS-A3M based on a new neighborhood rough set relation with adaptive neighbors. With the maximal-dependence, maximal-relevance, and maximal-significance evaluation criteria, OFS-A3M can select features with high correlation, high dependence, and low redundancy. Rahmaninia and Moradi [19] considered the challenges of high computational cost, the stability of the generated results, and the size of the final feature subset in OSFS and proposed two new methods called OSFSMI and OSFSMI-k. These two methods employ MI in a streaming manner to evaluate the relevancy and redundancy of features. In terms of neighborhood rough set theory, Zhou *et al.* [20] proposed a new OSFS method based on adaptive density neighborhood relation, named OFS-Density. By the density information of the surrounding instances and the fuzzy equal constraint, OFS-Density can select features with low redundancy. However, all these abovementioned methods consider the streaming features individually and ignore the problem of feature interaction during streaming feature selections.

III. Definitions of Relevance, Redundancy, and Interaction

We summarize some notations used in this article in Table I.

TABLE I
SUMMARY ON MATHEMATICAL NOTATION

Notation	Mathematical meanings
$\mathbb{D}$	Dataset
C	Condition feature set
D	Decision feature (Class attribute)
n	Sample size
m	Number of features
x_i	i^{th} sample
f_j	j^{th} feature
$P(.\|.)$	$P(D\|S)$ denotes the posterior probability of D condition on S
$I(.;.)$	$I(f;D)$ denotes the mutual information between f and D
S^t	The selected feature subset at time stamp t
$\|.\|$	$\|S\|$ the size of S
h	A mapping function from sample to class

Features in C can be categorized into three disjoint groups, namely, strong relevance, weak relevance and irrelevance as follows [36].

Definition 1 (Strong Relevance): Given C and D, $f \in C$, f is strongly relevant to D iff $\forall S \subseteq C\backslash\{f\}$ s.t. $P(D|S) \neq P(D|S, f)$.

Definition 2 (Weak Relevance): Given a condition feature space C and a decision attribute set D, $f \in C$, f is weakly relevant to D iff it is not strongly relevant and $\exists S \subset C\backslash\{f\}$ s.t. $P(D|S) \neq P(D|S, f)$.

Definition 3 (Irrelevance): Given a condition feature space C and a decision attribute set D, $f \in C$, f is irrelevant to D iff it is neither strongly nor weakly relevant and $\forall S \subseteq C\backslash\{f\}$ s.t. $P(D|S) = P(D|S, f)$.

Based on the Markov blankets, Yu and Liu [37] further divided weakly relevant features into redundant and nonredundant features.

Definition 4 (Markov Blanket): A Markov blanket of feature f, denoted as $M \subseteq C\backslash\{f\}$, makes all other features independent of f given M, that is, $\forall Y \in C\backslash(M \cup \{f\})$ s.t. $P(f|M, Y) = P(f|M)$.

Definition 5 (Redundancy): A feature $f \in C$ is a redundant feature, and hence, it should be discarded from C if it has a Markov blanket within C.

All these definitions mentioned earlier just consider the relationship between a single feature f and the class attribute D. However, features may contribute to the class by groups, and there exist interactions between them.

Let us use the data set Monks1 as an example to illustrate the problem of feature interaction. Meanwhile, we use MI to calculate the information between features. We calculate some values as follows:

$$I(a_1; c) = 0.0753, \quad I(a_2; c) = 0.0058$$
$$I(a_3; c) = 0.0047, \quad I(a_4; c) = 0.0263$$
$$I(a_5; c) = 0.2870, \quad I(a_6; c) = 7.5786e-04$$
$$I(\{a_1, a_2\}; c) = 0.5147 > I(a_1; c) + I(a_2; c) = 0.0811$$
$$I(\{a_1, a_4\}; c) = 0.0968 < I(a_1; c) + I(a_4; c) = 0.1016$$
$$I(\{a_1, a_5\}; c) = 0.3616 < I(a_1; c) + I(a_5; c) = 0.3623$$
$$I(\{a_1, a_2, a_5\}; c) = 1 > I(a_1; c) + I(a_2; c) + I(a_5; c) = 0.3681$$

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

$$I(\{a_1, a_2, a_3, a_5\}; c) = 1 > I(a_1; c) + I(a_2; c) + sI(a_3; c) + I(a_5; c) = 0.3728$$
$$I(\{a_1, a_2, a_3, a_4, a_5\}; c) = 1 > I(a_1; c) + I(a_2; c) + I(a_3; c) + I(a_4; c) + I(a_5; c) = 0.3991.$$

From these computing results, we can observe that the following holds.

1) For some feature combinations, such as $\{a_1, a_2\}$ and $\{a_1, a_2, a_5\}$, the values of these combinations are bigger the sum of individual features, which means that these features can contribute the class greater as an integral. In such a case, we can say that there is an interaction between these features. For some others, such as $\{a_1, a_4\}$ and $\{a_1, a_5\}$, the values of these combinations are smaller than the sum of individual features. Thus, not all feature combinations have interaction between them.
2) For $\{a_1, a_2, a_3, a_5\}$ and $\{a_1, a_2, a_3, a_4, a_5\}$, the information of integral is also more than the sum of individuals. However, the remove of features a_3 and a_4 will not decrease the information of these two combinations. Then, a_3 and a_4 are redundant features to the combinations. Thus, for an interactive feature combination, each of the features should be indispensable.

Therefore, for interactive features, the information of the combined feature set is more than the sum of each feature. Meanwhile, each feature in the interactive feature set should be indispensable. Based on this, we give the formal definition of feature interaction as follows.

Definition 6 (Feature Interaction): Given C and D, INT is a nonempty feature set, and $INT \subset C$. If $P(D|INT) > \sum_{f \in INT} P(D|f)$ and for $\forall f' \in INT$, $P(D|INT\backslash\{f'\}) < P(D|INT)$, then features in INT are said to have an interaction with each other on D, and we call INT an interactive feature set.

Theorem 1: Given INT and D, $f \in INT$. If INT is an interactive feature set, then f is strong relevant to D.

Proof 1: Because INT is an interactive feature set, for $\forall f \in INT$ and $S = INT\backslash\{f\}$, $P(D|INT) = P(D|S, f) > P(D|S)$, and $P(D|S, f) \neq P(D|S)$. Thus, f is strong relevant to D.

According to Theorem 1, for a feature space $INT \cup D$, if INT is an interactive feature set, all the features in INT are strong relevant features.

Theorem 2: Given C and D, $f \in C$. If $\exists INT \subseteq C(f \in INT$ and $|INT| \geq 2)$ is an interactive feature set, then f is relevant to D.

Proof 2: Suppose that $S = INT\backslash\{f\}$, and $S \subset C\backslash\{f\}$. Because INT is an interactive feature set, $P(D|S, f) > P(D|S)$, and $P(D|S, f) \neq P(D|S)$. Thus, f is relevant to D.

According to Theorem 2, for a feature $f \in C$, if there exist one more features in C that is interactive with f, then f is a relevant feature.

A given feature is relevant to the class when either individually or combined with other features, and it provides information about D. Selecting interactive features can reveal the implicit relationships in the given data sets. Ideally, we wish that the selected feature subset is an interactive feature set, and it can provide the maximal information about D. However, in real-world applications, it is unrealistic to guarantee that every feature in the selected feature set is strongly relevant to D. Meanwhile, the formal definitions of relevance and irrelevance are hard to apply on high-dimensional data sets directly as we cannot test all the subsets in C. Thus, we need to design a new method that can efficiently select features relevant to the class and interact with other features.

IV. Our New Online Feature Streams Selection Approach

In this article, we focus on the problem of feature interaction during streaming feature selection. We first give the problem formalization of streaming feature selection. Then, we give a systematic analysis of n-way interaction in feature selection and point out the neglected issues in existing feature selection methods. Finally, we propose the designed new metric named interaction gain and present a new streaming feature selection approach considering feature interaction within the selected feature subset.

A. Problem Formalization

Let $SFS = (C, D, h, t)$ denote a streaming feature selection framework. $C = [x_1, x_2, \ldots, x_n]^T$ is the condition feature set that contains n instances. D is the decision attribute consisted of n samples over the class labels $L = \{l_1, l_2, \ldots, l_k\}$, where l_i denotes the value of a class label and k is the number of distinct class labels. At each time stamp t, we get a new feature f_t without knowing the exact number of feature space in advance. With the selected feature subset S^{t-1} after time stamp $t-1$ and the new arriving feature f_t at time stamp t, the problem of streaming feature selection is to select a subset of $S^{t-1} \cup \{f_t\}$, which can make the mapping

$$h_t : x_i \rightarrow L \tag{1}$$

"as good as possible" according to certain evaluation criteria.

Most of the existing streaming feature selection methods consist of two main components: 1) irrelevant features discarding/relevant features selecting and 2) redundant features eliminating. Suppose that the new arriving feature is f_t at time stamp t. In the first step, if the new arriving feature is judged to be irrelevant, then it will be discarded directly for time-saving. However, it is difficult to apply Definition 3 to judge the irrelevant features. Thus, most of these algorithms calculate the information between f_t and D with different measures and then compare the value with a predefined threshold. Only if the information between f_t and D is bigger than the threshold, f_t will be considered as a candidate feature. Otherwise, it will be discarded as an irrelevant feature. In the second step, these algorithms will evaluate whether f_t is redundant. There are three possible results.

1) f_t is not redundant and it can increase the information of the candidate feature subset. If so, f_t will be added to the candidate feature subset.
2) f_t is not redundant, but it may make some features in the candidate feature subset be redundant. Then, redundant features will be removed.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

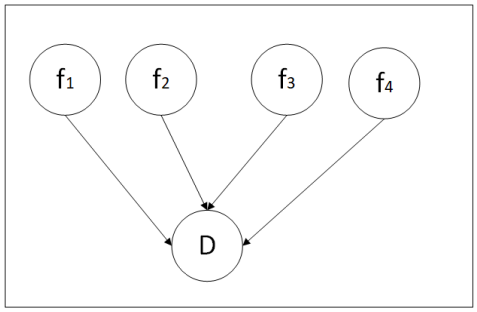

Fig. 1. Two-way interaction.

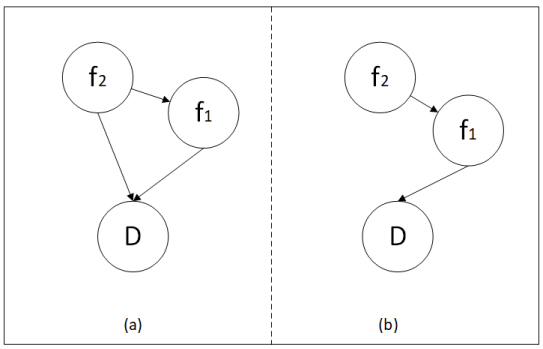

Fig. 2. Three-way interaction.

3) f_t is redundant and cannot contribute to the class. Then, it will be removed directly. Unlike existing streaming feature selection methods, our new method will select features from the perspective of feature interaction.

B. Analysis of N-Way Interaction in Feature Selection

In general, feature selection aims to select relevant and nonredundant features from the condition feature set. Nevertheless, for high-dimensional data sets, the definitions of strong relevance, weak relevance, and irrelevance are hard to be applied directly because we cannot test all the subsets of C. Thus, many feature selection methods use an approximate evaluating approach to judge the feature types.

Suppose that there are four features in a condition set $C = \{f_1, f_2, f_3, f_4\}$ and the decision feature is D. We use MI to measure the correlation between features.

1) Two-Way Interaction: For some MI-based feature selection methods [14], [38], if $I(f; D) = 0$ or $I(f; D) < \alpha$ (where α is a user-defined parameter), then f will be discarded as an irrelevant feature. These methods just consider the two-way interaction in feature selection, as shown in Fig. 1.

For two-way interaction, it just considers the relationship between pairs of features. For features f_1, f_2, f_3, and f_4, only if $I(f_i; D) > \alpha$ (α is a user-defined parameter), f_i will be considered as a candidate feature. Two-way interaction is easy to be applied. For example, SAOLA [14] is a two-way interaction method that employs the pairwise comparison techniques. However, these methods ignore the interaction among more than two features in C.

2) Three-Way Interaction: For three-way interaction, it considers the relationship among three features. Fig.2 illustrates the three-way interaction relationships in feature selection.

For three-way interaction in feature selection, there are two cases.

1) $I(f_1; D) > 0$, $I(f_2; D) > 0$, and $I(f_1; D|f_2) > I(f_1; D)$.
2) $I(f_1; D) > 0$, $I(f_2; D) = 0$, and $I(f_1; D|f_2) > I(f_1; D)$.

In case 1), both two-way interaction feature selection methods and three-way interaction feature selection methods may select both f_1 and f_2. However, in case 2), two-way interaction methods will not consider f_2 as a candidate feature for $I(f_2; D) = 0$.

Definition 7 (Three-Way Interaction Value): Given f_1, $f_2 \in C$ and D, $INT = \{f_1, f_2\}$ is an interactive feature set. We define the interaction value between f_1 and f_2 on D as

$$Int_D(f_1, f_2) = I(D; \{f_2, f_1\}) - I(D; f_1) - I(D; f_2). \quad (2)$$

Theorem 3: Given $INT = \{f_1, f_2\}$ and D, INT is an interactive feature set on D equivalent to $Int_D(f_1, f_2) > 0$.

Proof 3: If INT is an interactive feature set, according to Definition 6, we have $I(D; \{f_1, f_2\}) > \sum_{f \in INT} I(D; f) = I(f_1; D) + I(f_2; D)$. Thus, $Int_D(f_1, f_2) = I(D; \{f_2, f_1\}) - I(D; f_1) - I(D; f_2) > 0$. On the other hand, if $Int_D(f_1, f_2) = I(D; \{f_2, f_1\}) - I(D; f_1) - I(D; f_2) > 0$, we can conduct that $INT = \{f_1, f_2\}$ is an interactive feature set on D. Thus, $INT = \{f_1, f_2\}$ is an interactive feature set on D equivalent to $Int_D(f_1, f_2) > 0$.

Theorem 4: $Int_D(f_1, f_2) = I(D; f_2|f_1) - I(D; f_2) = I(D; f_1|f_2) - I(D; f_1)$

Proof 4: $I(D; \{f_2, f_1\}) = I(D; f_2) + I(D; f_1|f_2) = I(D; f_1) + I(D; f_2|f_1)$, then $Int_D(f_1, f_2) = I(D; \{f_2, f_1\}) - I(D; f_1) - I(D; f_2) = I(D; f_2) + I(D; f_1|f_2) - I(D; f_1) - I(D; f_2) = I(D; f_1|f_2) - I(D; f_1) = I(D; f_1) + I(D; f_2|f_1) - I(D; f_1) - I(D; f_2) = I(D; f_2|f_1) - I(D; f_2)$. Thus, $Int_D(f_1, f_2) = I(D; f_2|f_1) - I(D; f_2) = I(D; f_1|f_2) - I(D; f_1)$.

If $Int_D(f_1, f_2) > 0$, there is an **interaction (positive interaction)** between f_1 and f_2 on D. In other words, for interactive features f_1 and f_2, $Int_D(f_1, f_2) = I(D; f_1|f_2) - I(D; f_1) > 0$ and $I(D; f_1|f_2) > I(D; f_1)$, which means that f_2 can increase the information between D and f_1. We call positive interaction as interaction for short.

If $Int_D(f_1, f_2) < 0$, there is a **redundancy (negative interaction)** between f_1 and f_2 on D. If $Int_D(f_1, f_2) = I(D; f_1|f_2) - I(D; f_1) < 0$, in condition of f_2, the information between f_1 and D is smaller than $I(D; f_1)$. Thus, there must be some redundancy between f_1 and f_2 on D.

If $Int_D(f_1, f_2) = 0$, $I(D; \{f, f_1\}) - I(D; f_1) - I(D; f_2) = I(D; f_1) + I(D; f_2|f_1) - D(D; f_1) - I(D; f_2) = I(D; f_2|f_1) - I(D; f_2) = 0$, $I(D; f_2|f_1) = I(D; f_2)$, and f_1 and f_2 are irrelevant (noninteraction) on D.

3) Four-Way Interaction: For four-way interaction, it considers the relationship among four features, as shown in Fig.3.

For four-way interaction in feature selection, there are three different cases.

1) $I(f_1, D) > 0$, $I(f_2, D) = 0$, $I(f_3, D) = 0$, $I(f_1; D|f_2) > I(f_1; D)$, and $I(f_1; f_2|f_3) > I(f_1; f_2)$.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

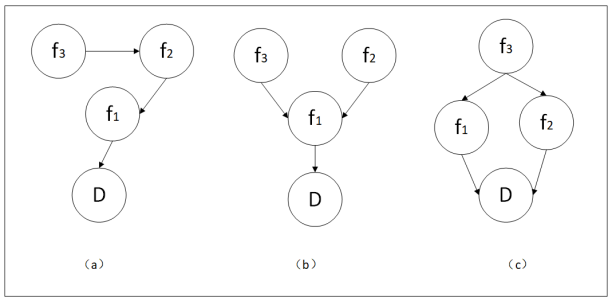

Fig. 3. Four-way interaction.

2) $I(f_1, D) > 0$, $I(f_2, D) = 0$, $I(f_3, D) = 0$, and $I(f_1; D|f_2, f_3) > I(f_1; D)$.
3) $I(f_1, D) > 0$, $I(f_2, D) > 0$, $I(f_3, D) = 0$, $I(f_1; D|f_3) > I(f_1; D)$, $I(f_2; D|f_3) > I(f_2; D)$.

In case 1), due to $I(f_1; D|f_2) > I(f_1; D)$, according to Theorem 4, we can conduct that $Int_D(f_1, f_2) > 0$. For the same reason, due to $I(f_1; f_2|f_3) > I(f_1; f_2)$, then $Int_{f_1}(f_2, f_3) > 0$. Thus, for four-way interaction 1), it equals to two three-way interactions, namely $INT_1 = \{f_1, f_2, D\}$ and $INT_2 = \{f_1, f_2, f_3\}$.

In case 2), for $I(f_1; D|f_2, f_3) > I(f_1; D)$, we can know that $Int_D(f_1, f_2) > 0$ and $Int_D(f_1, f_3) > 0$. Thus, for four-way interaction 2), it equals to two three-way interactions, namely $INT_1 = \{f_1, f_2, D\}$ and $INT_2 = \{f_1, f_3, D\}$.

In case 3), for $I(f_1; D|f_3) > I(f_1; D)$, then we can conduct that $Int_D(f_1, f_3) > 0$. For $I(f_2; D|f_3) > I(f_2; D)$, we can conduct that $Int_D(f_2, f_3) > 0$. Thus, for Four-way interaction 3), it equals to two three-way interactions: $INT_1 = \{f_1, f_2, D\}$ and $INT_2 = \{f_1, f_3, D\}$.

Theorem 5: Given $INT = \{f_1, f_2, f_3\}$ and D, if INT is a four-way interaction feature set, then it can be converted into some three-way interactions, and the sum of these three-way interaction values is bigger than 0.

Proof 5: As INT is an interactive feature set, according to Definition 6, we have $I(D; \{f_1, f_2, f_3\}) > \sum_{f \in INT} I(D; f) = I(f_1; D) + I(f_2; D) + I(f_3; D)$. For $I(D; \{f_1, f_2, f_3\}) = I(f_3; D|f_1, f_2) + I(f_2; D|f_1) + I(f_1; D)$, then $I(f_3; D|f_1, f_2) + I(f_2; D|f_1) + I(f_1; D) > I(f_1; D) + I(f_2; D) + I(f_3; D)$, and $[I(f_2; D|f_1) - I(f_2; D)] + [I(f_3; D|f_1, f_2) - I(f_3; D)] > 0$. There are three cases for this inequality.

1) $I(f_2; D|f_1) > I(f_2; D)$ and $I(f_3; D|f_1, f_2) > I(f_3; D)$.
2) $I(f_2; D|f_1) > I(f_2; D)$, $I(f_3; D|f_1, f_2) < I(f_3; D)$, and $[I(f_2; D|f_1) - I(f_2; D)] + [I(f_3; D|f_1, f_2) - I(f_3; D)] > 0$.
3) $I(f_2; D|f_1) < I(f_2; D)$, $I(f_3; D|f_1, f_2) > I(f_3; D)$, and $[I(f_2; D|f_1) - I(f_2; D)] + [I(f_3; D|f_1, f_2) - I(f_3; D)] > 0$.

For case 1), the interaction values for $\{f_1, f_2, D\}$, $\{f_1, f_3, D\}$, and $\{f_2, f_3, D\}$ are all positive. For case 2), there has positive interaction in $\{f_1, f_2, D\}$ but has redundancy in $\{f_1, f_3, D\}$ and $\{f_2, f_3, D\}$. Meanwhile, the sum of interaction values is bigger than 0. For case 3), there is redundancy in $\{f_1, f_2, D\}$ but has positive interaction in $\{f_1, f_3, D\}$ and $\{f_2, f_3, D\}$. Meanwhile, the sum of interaction values is bigger than 0. Thus, four-way interaction can be converted into some three-way interactions, and the sum of these three-way interaction values is bigger than 0.

In sum, for four-way interaction, we can convert it into the sum of some three-way interactions. Meanwhile, for real-world applications, it is easy to calculate the three-way interaction between features.

4) N-Way Interaction: For n-way interaction ($n > 4$), it will be very complicated with the increasing of n. Thus, it is unrealistic to apply high-way interaction in feature selection. However, we can approximate the high-way interaction by considering all possible three-way interactions. In other words, for more than three features, e.g., $\{f_1, f_2, f_3, D\}$, we can use $P(D|f_1, f_2)$, $P(D|f_1, f_3)$, and $P(D|f_2, f_3)$ to find all the possible three-way interaction relationships.

C. SFS-FI

In the streaming feature selection, we only get one feature at each time stamp. For each new arriving feature f_t, we should consider the relationship between f_t and feature $f'(f' \in S_{t-1})$ to discover the possible interactions. Thus, we define interaction gain in the following.

Definition 8 (Interaction Gain): Given C and D, S is a nonempty set, $S \subseteq C$, $f \in C$, and $f \notin S$. We define the interaction gain between f and S on D as

$$IG(f, S) = \sum_{f_i \in S} \{P(D|f, f_i) - P(D|f_i) - P(D|f)\}. \quad (3)$$

Theorem 6: Given C and D, $S \subseteq C$, $f \in C$, and $f \notin S$. If $IG(f, S) > 0$, f is relevant to D.

Proof 6: For $IG(f, S) = \sum_{f_i \in S} \{P(D|f, f_i) - P(D|f_i) - P(D|f)\} > 0$, there is at least one feature f' that makes $P(D|f, f') - P(D|f') - P(D|f) > 0$. Suppose that $INT = \{f, f'\}$, and $P(D|f, f') > P(D|f') + P(D|f)$; then, INT is an interactive feature set on D. According to Theorem 2, f is relevant to D.

Definition 9 (Positive Interaction): For a new arriving feature f_t at time stamp t and the selected feature subset S_{t-1}, if $IG(f_t, S_{t-1}) > 0$, f_t will be considered as a positive interactive feature for S_{t-1}.

Definition 10 (Negative Interaction): For a new arriving feature f_t at time stamp t and the selected feature subset S_{t-1}, if $IG(f_t, S_{t-1}) < 0$, f_t will be considered as a negative interactive feature for S_{t-1}.

For high-dimensional real-world data sets, the value of interaction gain could be very small for may new arriving features. Thus, we need a threshold γ for IG to reduce the number of selected features and improve efficiency. For a new arriving feature f_t at time stamp t and the selected feature subset S_{t-1}, if $IG(f_t, S_{t-1}) \geq \gamma$, f_t will be considered as a **strong** interactive feature for S_{t-1}. Otherwise, if $0 < IG(f_t, S_{t-1}) < \gamma$, f_t will be considered as a **weak** interactive feature for S_{t-1}.

Theorem 7: If $IG(f_t, S_{t-1}) < 0$, there exist a redundancy between f_t and some features in S_{t-1}.

Proof 7: For $IG(f_t, S_{t-1}) = \sum_{f' \in S_{t-1}} \{I(D; \{f', f_t\}) - I(D; f') - I(D; f_t)\} = \sum_{f' \in S_{t-1}} \{I(D; f') + I(D; f_t|f') - I(D; f') - I(D; f_t)\} = \sum_{f' \in S_{t-1}} \{I(D; f_t|f') - I(D; f_t)\} < 0$.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

Then, there must exist one feature $f_i \in S_{t-1}$ at least, which makes $I(D; f_t|f_i) - I(D; f_t) < 0$. $Int_D(f_t, f_i) = I(D; f_t|f_i) - I(D; f_t) < 0$. Thus, there exist a redundancy between f_t and some features in S_{t-1}.

During streaming feature selection, suppose that f_t is the new arriving feature at timestamp t, and the candidate feature set is S_{t-1}; it can be divided into three cases.

1) If $IG(f_t, S_{t-1}) \geq \gamma$, f_t will be consider as a strong interactive feature for S_{t-1} and can be selected into S directly.
2) If $IG(f_t, S_{t-1}) \leq 0$, according to Theorem 7, f_t will be discarded directly as a redundant feature.
3) If $0 < IG(f_t, S_{t-1}) < \gamma$, f_t is a weak interactive feature for S_{t-1}. In order to avoid selecting too many features, we will check whether there exist some features in S_{t-1} having redundancy with f_t. This can be further divided into two situations.
   a) If $\exists Y \in S_{t-1}$, $I(f_t; D|Y) = 0$ holds, and f_t will be considered as a redundant feature and be discarded directly.
   b) Otherwise, if $\exists Y \in S_{t-1}$, $I(Y; D|f_t) = 0$ holds, Y will be discarded as a redundant feature.

For case 3), $I(f_t; D|Y)$ and $I(Y; D|f_t)$ exactly equal to 0, which is rare for real-world data sets [14]. Thus, we rewrite these two situations into: 1) if $\exists Y \in S_{t-1}$, $I(Y; D) > I(f_t; D)$ and $I(f_t; Y) \geq I(f_t; D)$ holds, and f_i is discarded and 2) if $\exists Y \in S_{t-1}$, $I(f_t; D) > I(Y; D)$ and $I(f_t; Y) \geq I(Y; D)$ holds, and Y can be removed from S_{t-1}.

To sum up, we propose the new streaming feature selection algorithm, which considers the feature interaction between features, named SFS-FI, as shown in Algorithm 1.

Algorithm 1 SFS-FI

Require:
C: the condition features;
D: the decision feature;
γ: the threshold for interaction gain;
Ensure:
S_t: the selected feature subset at time stamp t;
1: **Repeat**
2: get a new feature f_t at time stamp t;
3: **If** $IG(f_t, S_{t-1}) \geq \gamma$
4: $S_t = S_{t-1} \cup f_t$;
5: Go to Step 23;
6: **End If**
7: **If** $IG(f_t, S_{t-1}) \leq 0$
8: Discard f_t;
9: Go to Step 23;
10: **End If**
11: **If** $0 < IG(f_t, S_{t-1}) < \gamma$
12: **For** each feature f_i in S
13: **If** $I(f_i; D) > I(f_t; D) \& I(f_t; f_i) \geq I(f_t; D)$
14: Discard f_t;
15: Go to Step 23;
16: **End If**
17: **If** $I(f_t; D) > I(f_i; D) \& I(f_t; f_i) \geq I(f_i; D)$
18: $S_{t-1} = S_{t-1} - f_i$;
19: **End If**
20: **End For**
21: $S_t = S_{t-1} \cup f_t$;
22: **End If**
23: **Until** no features are available;
24: **return** S_t;

At step 3, we calculate the interaction gain between f_t and S_{t-1}. If $IG(f_t, S_{t-1})$ is bigger than the predefined threshold γ, f_t will be considered as a strong positive interactive feature and be added into S_{t-1}. At step 7, if $IG(f_t, S_{t-1}) \leq 0$, f_t will be discarded as a negative interactive feature. From step 11 to step 22, f_t is considered as a weak positive interactive feature, and the algorithm will check if there exist some features in S_{t-1} having a redundancy with f_t.

For data with discrete values, we use the measure of MI. The MI is a measure of the amount of information that one random variable has about another variable [27]. Formally, for two features X and Y, the MI is defined as follows:

$$I(X; Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i, y_j) log_2 \left(\frac{p(x_i, y_j)}{p(x_i) \cdot p(y_j)} \right). \quad (4)$$

For data with continuous values, we adopt the best known measure of Fisher's Z-test [39] to calculate correlations between features. 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:

$$P(f_i, Y|S) = \frac{-\left(\left(\sum_{f_i Y S} \right)^{-1} \right)_{f_i Y}}{\left(\left(\sum_{f_i Y S} \right)^{-1} \right)_{f_i f_i} \left(\left(\sum_{f_i Y S} \right)^{-1} \right)_{YY}}. \quad (5)$$

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 with each other.

Besides, SFS-FI is not just SAOLA [14] using a threshold γ. This is because SFS-FI is different from SAOLA in two aspects: 1) SAOLA just considers the relationship in feature pairs and it is one of the Two-way interaction methods, while SFS-FI is a three-way interaction method and considers the relationship more than two features and 2) SFS-FI divides the feature interaction into two categories, namely positive interaction and negative interaction. Only the features belonging to the negative interaction will be discarded directly. This reduces the possibility of the loss of important features. Experiment results in Section V demonstrate the efficiency of SFS-FI compared with SAOLA.

D. Time Complexity of SFS-FI

At time stamp t, suppose that the number of selected feature subset in S_{t-1} is $|S_{t-1}|$. From step 3 to step 10, we calculate $IG(f_t, S_{t-1})$ and compare it with γ and 0. The time complexity of these steps is $O(|S_{t-1}|)$. If $0 < IG(f_t, S_{t-1}) < \gamma$,

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

TABLE II
REAL-WORLD DATA SETS

Index	Data Set	Instances	Features	Classes	Feature Characteristics
1	ADA	4,562	48	2	Integer,Categorical
2	SYLVA	14,364	216	2	Integer,Categorical
3	MADELON	2,600	500	2	Integer
4	GINA	3,468	970	2	Integer
5	HIVE	4,229	1,617	2	Categorical
6	COLON	62	2,000	2	Real
7	SRBCT	63	2,308	4	Real
8	LYMPHOMA	62	4,026	3	Real
9	PROSTATE	102	6,033	2	Real
10	LEUKEMIA	72	7,129	2	Real
11	DLBCL	77	7,129	2	Integer
12	LEUKEMIA(3C)	72	7,129	3	Integer
13	ARCENE	200	10,000	2	Integer
14	BREAST	97	24,481	2	Real

we compare the information between f_t and each feature in S_{t-1}. The worst time complexity of steps 12–20 is $O(|S_{t-1}|)$. Thus, the worst time complexity of SFS-FI is $O(m^2)$. However, for real-world data sets, it is impossible that all features are weak positive interactive and SFS-FI can select all the features.

V. EXPERIMENTAL RESULTS

A. Experiment Setup

In this section, we apply the proposed streaming feature selection algorithm on 14 real-world data sets, including eight DNA microarray data sets [LEUKEMIA, COLON, LYMPHOMA, PROSTATE, SRBCT, DLBCL, BREAST, and LEUKEMIA(3C)], two NIPS 2003 data sets (MADELON and ARCENE), and four WCCI 2006 data sets (ADA, GINA, SYLVA, and HIVE), as shown in Table II. We analyze the value of parameter γ at first. Then, we compare SFS-FI with some state-of-the-art streaming feature selection approaches.

In our experiments, we use two basic classifiers, KNN ($k = 3$) and SVM, in MATLAB to evaluate a selected feature subset. We perform fivefold cross-validation on each data set. 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. We run each data set ten times and report the average prediction accuracy, running time, and the mean number of selected features. All experimental results are conducted on a PC with AMD 3700X, 3.6-GHz CPU, and 32-GB memory.

To further analyze the prediction accuracies of SFS-FI against its rivals, paired t-tests are conducted at a 95% significance level, and the win/tie/lose (W/T/L for short) counts are summarized. Meanwhile, we conduct the Friedman test at a 95% significance level under the null-hypothesis to validate whether SFS-FI, and its rivals have a significant difference in prediction accuracy. If the null-hypothesis at the Friedman test is rejected, we proceed with the Nemenyi test as a post-hoc test [40].

B. Analysis on Parameter γ

To test the effect of different values of γ, we apply SFS-FI with the values of $\gamma = 0.01, 0.03, 0.05$, and 0.09 on these

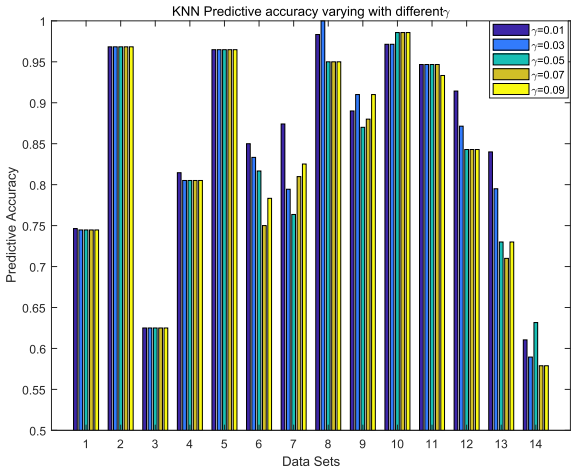

Fig. 4. KNN predictive accuracy varying with different values of γ.

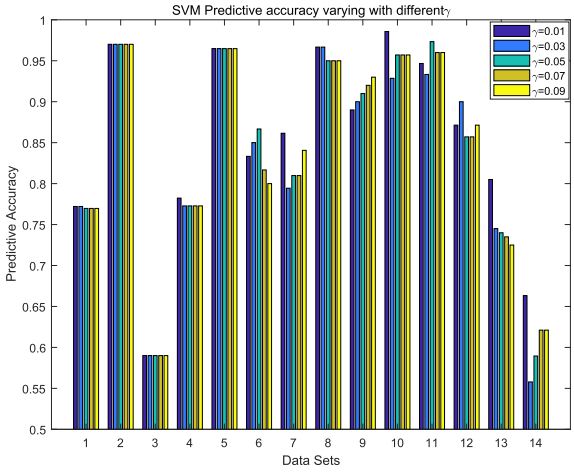

Fig. 5. SVM predictive accuracy varying with different values of γ.

14 data sets respectively. The significance level is set to 0.01 for Fisher's Z-test. Experimental results of predictive accuracy, running time, and the mean number of selected features can be seen from Figs. 4–7.

The p-values of the Friedman test on KNN and SVM are 0.1179 and 0.5095, respectively. Thus, there is no significant difference in predictive accuracy in both cases of KNN and SVM. Meanwhile, the p-values on running time and the mean

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

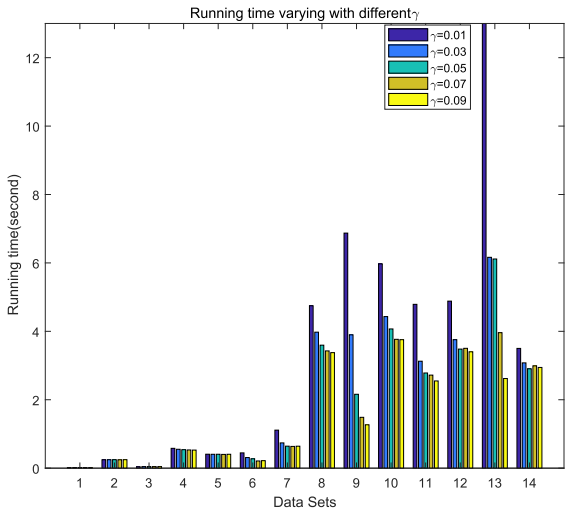

Fig. 6. Running time varying with different values of γ.

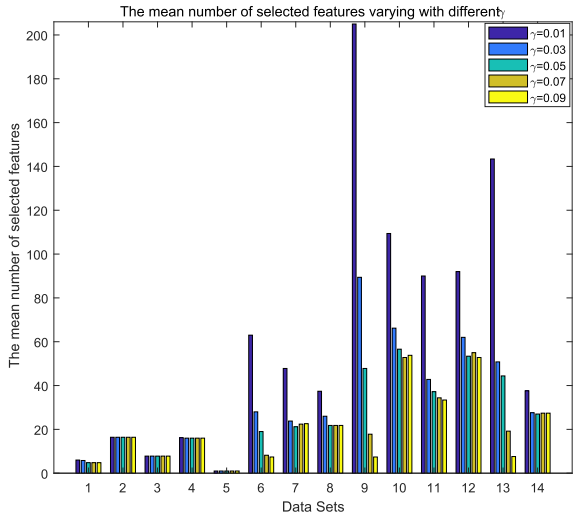

Fig. 7. Mean number of selected features varying with different values of γ.

number of selected features are 1.8087e^{-}07 and 7.9883e^{-}07. Thus, there is a significant difference between different values of γ on running time and the mean number of selected features.

From Figs. 4–7, we can observe the following.

1) On predictive accuracy, for some data sets, such as SYLVA, MADELON, and HIVE, different values of γ have the same predictive accuracy and select the same number of features, which means that there is very weak or even no feature interaction in these data sets. For data sets SRBCT and ARCENE, a smaller value of γ gets higher performance. Meanwhile, for data sets PROSTATE and LEUKEMIA, a bigger value of γ performs better. Different values of γ will affect the number of selected features that make the performance difference. In general, for strong interaction data sets, it prefers a bigger value of γ. For weak interaction data sets, a smaller value of γ performs better.
2) On running time, the difference between these five values of γ is large. If the data set has very weak or even no interaction, SFS-FI runs fast and spends almost the same time for all different values of γ. For others, a smaller value of γ makes SFS-FI considering more features on the fly, which leads to more running time.
3) On the mean number of selected features, the value of γ is inversely proportional to the number of selected features. Meanwhile, the value of γ is not necessarily as small as possible for it can increase the probability of selecting redundant features.

To sum up, for some data sets that have very weak or even no feature interaction, different values of γ cannot affect the performance of the final selected features. For others, a smaller value of γ can make SFS-FI considering more possible interactive features during streaming feature selection. In the following experiments, we set $\gamma = 0.01$ for our new method. Meanwhile, we will not conduct the experiments on data sets SYLVA, MADELON, GINA, and HIVE because there is very weak or even no feature interaction.

C. SFS-FI Versus Online Streaming Feature Selection Methods

We compare SFS-FI with other seven streaming feature selection methods, including: Grafting [34], Alpha-investing [35], OSFS and Fast-OSFS [12], SAOLA [14], OFS-Density [20], and OFS-A3M [18]. All aforementioned algorithms are implemented in MATLAB [41] and are applied on ten data sets (ADA, COLON, SRBCT, LYMPHOMA, PROSTATE, LEUKEMIA, DLBCL, LEUKEMIA(3C), ARCENE, and BREAST). The significance level α is set to 0.01 for OSFS, Fast-OSFS, SAOLA, and SFS-FI. For Grafting, the parameter λ is set to 0.5. For Alpha-investing, the parameters are set to the values used in [35].

Tables III and IV summarize the predictive accuracy of SFS-FI against the other seven algorithms using the KNN and SVM classifiers. Tables V and VI show the running time and the mean number of selected features respectively. The p-values of the Friedman test on KNN, SVM, running time, and the mean number of selected features are 1.5113e^{-}04, 7.3207e^{-}06, 5.4861e^{-}15, and 9.5349e^{-}08, 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 the critical difference (CD) is 3.3230.

From Tables III–VI, we have the following observations.

1) *SFS-FI Versus Grafting:* According to the average ranks and the value of CD, there is a significant difference between SFS-FI and Grafting in both cases of KNN and SVM. SFS-FI significantly performs better than Grafting on predictive accuracy. Meanwhile, SFS-FI runs much faster than Grafting. Grafting treats feature selection as an integral part of learning a predictor within a regularized framework. When a new feature arrives, Grafting needs to update the whole model, and this will result in high time complexity. Besides, for different data sets, the number of selected features by Grafting varies widely. For example, Grafting selects only one feature on DLBCL but 108 on ARCENE. This makes Grafting cannot well adapted to different types of data sets.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PREDICTIVE ACCURACY USING KNN AS THE CLASSIFIER

Data Set	SFS-FI	Grafting	Alpha-investing	OSFS	Fast-OSFS	SAOLA	OFS-Density	OFS-A3M
ADA	0.7836	0.8033	0.8044	**0.816**	0.8149	0.8158	0.8026	0.752
COLON	**0.85**	0.6833	0.5833	0.7167	0.7833	0.8333	0.7333	0.7333
SRBCT	0.8559	0.7119	0.7063	0.7636	0.7636	0.7329	**0.8839**	0.8615
LYMPHOMA	**1**	0.9167	0.7333	0.9167	0.9	0.9167	0.9833	0.95
PROSTATE	0.89	0.71	0.64	0.88	0.88	0.86	**0.91**	0.88
LEUKEMIA	**0.9714**	0.6571	0.6571	0.9	0.9286	0.9286	0.9	0.8714
DLBCL	**0.9067**	0.7467	0.76	0.8933	**0.9067**	**0.9067**	0.84	0.88
LEUKEMIA(3C)	**0.8714**	0.5714	0.6571	0.8143	**0.8714**	**0.8714**	0.8429	0.8571
ARCENE	0.8	0.69	0.72	0.595	0.645	0.63	**0.815**	0.805
BREAST	**0.6842**	0.6316	0.6	**0.6842**	0.6421	**0.6842**	0.5053	0.6737
AVG.	**0.8613**	0.7122	0.6861	0.7979	0.8135	0.8179	0.8216	0.8264
W/T/L	-	9/0/1	9/0/1	6/3/1	6/3/1	6/3/1	6/1/3	7/2/1
AVG. RANKS	**2.4**	6.55	6.85	4.5	3.9	3.65	3.9	4.25

TABLE IV
PREDICTIVE ACCURACY USING SVM AS THE CLASSIFIER

Data Set	SFS-FI	Grafting	Alpha-investing	OSFS	Fast-OSFS	SAOLA	OFS-Density	OFS-A3M
ADA	0.8099	0.8296	0.8397	**0.8456**	0.8434	0.8421	0.8412	0.752
COLON	0.8	0.5833	0.65	0.7333	0.8	**0.8167**	0.7833	0.7833
SRBCT	**0.8434**	0.7147	0.3161	0.7175	0.779	0.7944	0.8196	0.8154
LYMPHOMA	**0.9833**	0.8667	0.75	0.9167	0.8667	0.9167	**0.9833**	0.95
PROSTATE	0.93	0.7	0.62	0.91	0.86	0.83	**0.94**	0.85
LEUKEMIA	**0.9571**	0.6429	0.6571	0.9143	0.9	0.9286	0.9	0.8286
DLBCL	**0.96**	0.72	0.84	0.8533	0.8533	0.9333	0.88	0.8533
LEUKEMIA(3C)	**0.8857**	0.6143	0.7	0.8	0.8429	0.8429	0.8429	0.8571
ARCENE	0.8	0.65	0.725	0.66	0.69	0.645	0.785	**0.805**
BREAST	0.6421	0.6105	0.6421	**0.6737**	0.6632	0.6632	0.5789	0.6421
AVG.	**0.8611**	0.6932	0.674	0.8024	0.8098	0.8212	0.8354	0.8136
W/T/L	-	9/0/1	8/1/1	7/1/2	7/1/2	7/0/3	7/1/2	8/2/0
AVG. RANKS	**2.4**	7.25	6.6	4.15	4.1	3.7	3.55	4.25

TABLE V
RUNNING TIME (SECONDS)

Data Set	SFS-FI	Grafting	Alpha-investing	OSFS	Fast-OSFS	SAOLA	OFS-Density	OFS-A3M
ADA	0.0164	3.6009	0.0369	1.1432	0.3893	0.0164	90.4623	57.1206
COLON	0.3641	1.4201	0.0339	0.1605	0.1196	0.1308	0.5389	0.5711
SRBCT	0.62	2.204	0.0395	0.579	0.1996	0.3192	0.6285	0.6802
LYMPHOMA	4.2292	1.9803	0.1996	2.9709	0.7161	1.7185	1.3657	1.3785
PROSTATE	5.434	3.7117	0.3306	0.9354	0.4551	0.6274	2.5739	2.8499
LEUKEMIA	4.5388	2.3439	0.4196	1.4223	0.5318	0.7874	2.433	2.7548
DLBCL	3.629	48.4898	0.4322	1.16	0.5234	0.6922	2.1211	2.2796
LEUKEMIA(3C)	4.6668	28.6855	0.6556	1.5208	0.5367	0.7569	1.987	2.4077
ARCENE	13.3545	46.6979	1.1146	3.144	0.822	1.355	17.5661	11.2592
BREAST	2.7072	11.161	3.2197	1.8693	1.4318	1.7285	10.9976	11.8432
AVG.	3.95	15.02	0.64	1.49	0.57	0.81	13.06	9.31
AVG. RANKS	5.95	7.1	1.8	4.3	1.9	2.95	5.7	6.3

2) *SFS-FI Versus Alpha-Investing:* There is a significant difference between SFS-FI and Alpha-investing on predictive accuracy in both cases of KNN and SVM. SFS-FI significantly performs better than Alpha-investing on predictive accuracy. Alpha-investing runs very fast, but it gets the lowest mean predictive accuracy among all these competing algorithms. For most of these data sets, such as COLON, LEUKEMIA, and SRBCT, Alpha-investing can only select the first one or two features that lead to bad performance.

3) *SFS-FI Versus OSFS:* There is no significant difference between SFS-FI and OSFS on predictive accuracy with KNN and SVM. However, SFS-FI outperforms OSFS on six and seven of the ten data sets with KNN and SVM, respectively. OSFS runs a little faster than SFS-FI on average. Meanwhile, OSFS selects the fewest mean number of features among all these competing methods. Thus, some important information may be missing, which causes lower predictive accuracy.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
MEAN NUMBER OF SELECTED FEATURES

Data Set	SFS-FI	Grafting	Alpha-investing	OSFS	Fast-OSFS	SAOLA	OFS-Density	OFS-A3M
ADA	5.4	33.8	33.4	7.6	8.2	4.6	9	1
COLON	65.4	58.6	1	1.8	2.2	3	3.8	23.8
SRBCT	32.6	66.4	1	2.4	4.8	18.2	5.2	6.8
LYMPHOMA	45.6	65.2	2.8	2.8	5.6	34.8	16	3.8
PROSTATE	165.8	98	1.8	1.4	4	14	3.8	25
LEUKEMIA	101.8	70	1.2	2.4	4.8	20.4	3.8	12.2
DLBCL	86.4	1	2	2.4	4.6	13.4	5.6	18.6
LEUKEMIA(3C)	103.8	1.4	7.4	2.8	5.8	21	3.6	10
ARCENE	151.6	108.2	7	2.6	5.2	18.6	46.2	33.6
BREAST	30.4	70.6	3	2	4	18.4	6.8	38
AVG.	78.88	57.32	6.06	2.82	4.92	16.64	10.38	17.28
AVG. RANKS	7.1	6.2	2.55	1.95	3.6	5.1	4.4	5.1

4) *SFS-FI Versus Fast-OSFS:* There is no significant difference between SFS-FI and Fast-OSFS on predictive accuracy in the cases of KNN and SVM. However, SFS-FI gets higher predictive accuracy than FAST-OSFS on six of the ten data sets at least with both cases of KNN and SVM. Fast-OSFS is much faster than SFS-FI. As similar to OSFS, Fast-OSFS considers features individually and selects much fewer features on these data sets that lead to the loss of some important information.
5) *SFS-FI Versus SAOLA:* There is no significant difference between SFS-FI and SAOLA on predictive accuracy. However, SFS-FI outperforms SAOLA on six and seven of the ten data sets with KNN and SVM, respectively. SAOLA employs novel online pairwise comparison techniques and maintains a parsimonious model over time in an online manner. Therefore, SAOLA is faster than SFS-FI. However, SAOLA does not consider the interaction between features, which is important for the final performance.
6) *SFS-FI Versus OFS-Density:* There is no significant difference between SFS-FI and OFS-Density on predictive accuracy. However, SFS-FI gets higher predictive accuracy than OFS-Density on six of the ten data sets at least with KNN and SVM respectively. SFS-FI is faster than OFS-Density and selects more features. Based on neighborhood rough set theory, OFS-Density uses the density information for feature selection and considers the selected feature subset as an integral. However, for some data sets, such as COLON and LEUKEMIA, OFS-Density selects much fewer features for the very sparse of data distribution.
7) *SFS-FI Versus OFS-A3M:* According to the average ranks and the value of CD, there is no significant difference between SFS-FI and OFS-A3M on predictive accuracy in both cases of KNN and SVM. However, SFS-FI gets higher predictive accuracy than OFS-A3M on seven of ten data sets at least with both KNN and SVM. Like OFS-Density, OFS-A3M bases on the GAP neighborhood relationship and the dependence degree of selected feature subset for feature selection. However, the cap information can be greatly affected by the sample distribution.

In sum, SFS-FI gets the maximum average predictive accuracy and the minimum value of average ranks with both cases of KNN and SVM. Meanwhile, SFS-FI is faster than some of the competing algorithms. With the considering of feature interaction, SFS-FI selects the most features. However, the selection of interactive features demonstrates to have a contribution to the final performance.

VI. CONCLUSION

In this article, we study the problem of feature interaction and propose a new streaming feature selection method that can select features to interact with each other. With the formal definition of feature interaction, we analyze and demonstrate the relationship between feature relevance and feature interaction. Besides, we systematically analyze the two-way interaction, three-way interaction, and four-way interaction in feature selection and design the new metric interaction gain that can measure the interaction degree between the new arriving feature and the selected feature subset. In terms of interaction gain, we demonstrate that the features selected by our new method are relevant to the class and interact with each other. Experiments conducted on real-world microarray data sets indicate the efficiency of our new method. In our further work, more high-way interactions, including five-way interaction, will be analyzed and considered for streaming feature selection.

REFERENCES

[1] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.
[2] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. London, U.K.: Chapman & Hall, 2007.
[3] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, Oct. 2003.
[4] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[5] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," in *Proc. Conf. UAI*, 2011, pp. 1–8.
[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., Ser. B, Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.
[7] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, and Z. Dong, "Feature selection based on neighborhood discrimination index," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2986–2999, Jul. 2018.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.

[8] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for Web image search," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4649–4661, Nov. 2012.
[9] W. Ding *et al.*, "Subkilometer crater discovery with boosting and transfer learning," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 4, pp. 1–22, Jul. 2011.
[10] G. Ditzler, J. LaBarck, J. Ritchie, G. Rosen, and R. Polikar, "Extensions to online feature selection using bagging and boosting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4504–4509, Sep. 2018.
[11] J. Li *et al.*, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, Dec. 2017.
[12] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1178–1192, May 2013.
[13] J. Wang, P. Zhao, S. C. H. Hoi, and R. Jin, "Online feature selection and its applications," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 698–710, Mar. 2014.
[14] K. Yu, X. Wu, W. Ding, and J. Pei, "Scalable and accurate online feature selection for big data," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 2, pp. 1–39, Dec. 2016.
[15] J. Wang *et al.*, "Online feature selection with group structure analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3029–3041, Nov. 2015.
[16] P. Zhou, X. Hu, P. Li, and X. Wu, "Online feature selection for high-dimensional class-imbalanced data," *Knowl.-Based Syst.*, vol. 136, pp. 187–199, Nov. 2017.
[17] S. Eskandari and M. M. Javidi, "Online streaming feature selection using rough sets," *Int. J. Approx. Reasoning*, vol. 69, pp. 35–57, Feb. 2016.
[18] P. Zhou, X. Hu, P. Li, and X. Wu, "Online streaming feature selection using adapted neighborhood rough set," *Inf. Sci.*, vol. 481, pp. 258–279, May 2019.
[19] M. Rahmaninia and P. Moradi, "OSFSMI: Online stream feature selection method based on mutual information," *Appl. Soft Comput.*, vol. 68, pp. 733–746, Jul. 2018.
[20] P. Zhou, X. Hu, P. Li, and X. Wu, "OFS-density: A novel online streaming feature selection method," *Pattern Recognit.*, vol. 86, pp. 48–61, Feb. 2019.
[21] K. Yu, X. Wu, W. Ding, and J. Pei, "Towards scalable and accurate online feature selection for big data," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 660–669.
[22] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *11th Int. Conf. Mach. Learn.*, 1994, pp. 121–129.
[23] A. Jakulin and I. Bratko, "Testing the significance of attribute interactions," in *Proc. 21st Int. Conf. Mach. Learn. ICML*, 2004, p. 52.
[24] Z. Zhao and H. Liu, "Searching for interacting features in subset selection," *Intell. Data Anal.*, vol. 13, no. 2, pp. 207–228, 2009.
[25] G. Wang, Q. Song, B. Xu, and Y. Zhou, "Selecting feature subset for high dimensional data via the propositional FOIL rules," *Pattern Recognit.*, vol. 46, no. 1, pp. 199–214, Jan. 2013.
[26] Z. Zeng, H. Zhang, R. Zhang, and C. Yin, "A novel feature selection method considering feature interaction," *Pattern Recognit.*, vol. 48, no. 8, pp. 2656–2666, Aug. 2015.
[27] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 175–186, Jan. 2014.
[28] P. P. Kundu and S. Mitra, "Feature selection through message passing," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4356–4366, Dec. 2017.
[29] Y. Yang, D. Chen, H. Wang, and X. Wang, "Incremental perspective for feature selection based on fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1257–1273, Jun. 2018.
[30] X. Xue, M. Yao, and Z. Wu, "A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm," *Knowl. Inf. Syst.*, vol. 57, no. 2, pp. 389–412, Nov. 2018.
[31] T. Pang, F. Nie, J. Han, and X. Li, "Efficient feature selection via $\ell_{2,0}$-norm constrained sparse regression," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 880–893, May 2019.
[32] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.
[33] A. Jakulin and I. Bratko, "Analyzing attribute dependencies," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery, PKDD*, 2003, pp. 229–240.
[34] S. Perkins and J. Theiler, "Online feature selection using grafting," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 592–599.
[35] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar, "Streamwise feature selection," *J. Mach. Learn. Res.*, vol. 3, no. 2, pp. 1532–4435, 2006.
[36] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.
[37] L. Yu and H. Liu, "Eficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, no. 12, pp. 1205–1224, 2004.
[38] H. Li, X. Wu, Z. Li, and W. Ding, "Group feature selection with streaming features," in *Proc. IEEE 13th Int. Conf. Data Mining*, Dec. 2013, pp. 1109–1114.
[39] J. M. Pea, "Learning Gaussian graphical models of gene networks with false discovery rate control," in *Proc. Eur. Conf. Evol. Comput., Mach. Learn. Data Mining Bioinf. (EvoBIO)*, 2008, pp. 165–176.
[40] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[41] K. Yu, W. Ding, and X. Wu, "LOFS: A library of online streaming feature selection," *Knowl.-Based Syst.*, vol. 113, pp. 1–3, Dec. 2016.

Peng Zhou received the Ph.D. degree from the Hefei University of Technology, Hefei, China, in 2018.

He is currently a Lecturer with Anhui University, Hefei. His research interests include data mining and knowledge engineering.

Peipei Li received the B.S., M.S., and Ph.D. degrees from the Hefei University of Technology, Hefei, China, in 2005, 2008, and 2013, respectively.

She was a Research Fellow with Singapore Management University, Singapore, from 2008 to 2009. She was a Student Intern with Microsoft Research Asia, Beijing, China, from August 2011 to December 2012. She is currently an Associate Professor with the Hefei University of Technology, China. Her research interests include data mining and knowledge engineering.

Shu Zhao received the Ph.D. degree in computer science from Anhui University, Hefei, China, in 2007.

She is currently a Professor with the Department of Computer Science and Technology, Anhui University. Her current research interests include quotient space theory, granular computing, data mining, and machine learning.

Xindong Wu (Fellow, IEEE) received the Ph.D. degree in artificial intelligence from The University of Edinburgh, Edinburgh, U.K., in 1993.

He is currently a Professor with the Hefei University of Technology, Hefei, China, and the President of the Mininglamp Academy of Sciences, Beijing, China. His research interests include data mining, big data analytics, and knowledge engineering.

Dr. Wu is a fellow of the American Association for the Advancement of Science (AAAS). He is the Founder and the current Steering Committee Chair of the IEEE International Conference on Data Mining and the Founder and the current Editor-in-Chief of *Knowledge and Information Systems*. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2005 to 2008.

Authorized licensed use limited to: Anhui University. Downloaded on November 24,2021 at 07:16:28 UTC from IEEE Xplore. Restrictions apply.