This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3350630

Online Heterogeneous Streaming Feature Selection without Feature Type Information

Peng Zhou, Yunyun Zhang, Zhaolong Ling, Yuanting Yan, Shu Zhao, and Xindong Wu, Fellow, IEEE

Abstract—Feature selection aims to select an optimal minimal feature subset from the original datasets and has become an indispensable preprocessing component before data mining and machine learning, especially in the era of big data. However, features may be generated dynamically and arrive individually over time in practice, which we call streaming features. Most existing streaming feature selection methods assume that all dynamically generated features are the same type or assume we can know the feature type for each new arriving feature in advance, but this is unreasonable and unrealistic. Therefore, this paper first studies a practical issue of Online Heterogeneous Streaming Feature Selection without the feature type information before learning, named OHSFS. Specifically, we first model the streaming feature selection issue as a minimax problem. Then, in terms of MIC (Maximal Information Coefficient), we derive a new metric MIC_{Gain} to determine whether a new streaming feature should be selected. To speed up the efficiency of OHSFS, we present the metric MIC_{Cor} that can directly discard low correlation features. Finally, extensive experimental results indicate the effectiveness of OHSFS. Moreover, OHSFS is nonparametric and does not need to know the feature type before learning, which aligns with practical application needs.

Index Terms—Online Feature Selection, Streaming Feature, Heterogeneous Feature, Maximal Information Coefficient

1 Introduction

Feature selection aims to select the smallest sized subset of the original feature space that preserves the best salient features required from the dataset [1, 2]. With the explosive growth of data volume and dimension, feature selection has become an indispensable data preprocessing technique that is widely used in data mining, machine learning, and other fields [3]. By removing noisy, irrelevant, and redundant features, machine learning can gain significant benefits from feature selection, such as better performance, less running time, and better understandability [4, 5].

Traditional feature selection assumes that the entire feature space can be fully presented to the learner before learning [6]. To select an optimal feature subset, feature selection algorithms tend to traverse the entire dataset multiple times. However, in real-world applications, such as image analysis [7] and Martian crater detection [8], not all features can be acquired before learning. Features are generated and arrive one by one over time, while the number of samples remains fixed, which we call streaming features [9]. For example, because the high cost of conducting wet-lab experiments in bioinformatics, acquiring the complete set of features for every training instance is prohibitive, and it is impossible to wait for a complete set of features [10]. Besides, for the product to be processed in an industrial production line, it always requires multiple steps by different devices which dynamically generate different streaming features over time [11]. Online streaming feature selection that deals with feature streams in an online manner has attracted extensive attention recently [12].

Feature selection methods can be broadly categorized as the filter, wrapper, and embedded according to different selection strategies [13]. Filter methods select the features in terms of specific feature measurements, while wrapper methods use predefined classifiers as a black box to evaluate the selected features. Embedded methods perform feature selection in the process of model construction. Unlike traditional feature selection methods, there are two main challenges for streaming feature selection: (1) the entire feature space is unknown or even infinite, (2) and we must decide whether to retain or discard the new arrival feature on the fly [14]. Due to storage space limitations, once a new arriving feature is discarded, we cannot use it again. Therefore, most existing online streaming feature selection methods apply a filter model to select the optimal streaming features [15]. In other words, these methods always need to design some measurements to calculate the association between features.

Generally speaking, the feature type of the target dataset can be categorized into homogeneous features and heterogeneous features [16, 17]. For example, characteristics in medical diagnostic data may include a patient's gender, age, weight, and blood pressure, where the gender type is a categorical value, the age is an integer value, and the weight and blood pressure are numerical values. Existing streaming feature selection methods either design for single feature type or provide two versions of algorithms for both categorical and numerical features, respectively [12]. For instance, based on penalized likelihood ratio, mutual information, and classical rough set theory, α-investing [18], GFSSF (Group Feature Selection with Streaming Features) [19], and OS-NRRSARA-SA [20] are designed for categorical features respectively. In terms of neighborhood rough set theory, K-OFSD (Online

This work was supported in part by the National Natural Science Foundation of China under grants (62376001, 61906056, 62376002, 61806002, 62120106008).

Peng Zhou, Yunyun Zhang, Zhaolong Ling, Yuanting Yan and Shu Zhao are with the Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, and School of Computer Science and Technology, Anhui University, and Information Materials and Intelligent Sensing Laboratory of Anhui Province, Hefei 230601, China(e-mail: doodzhou@ahu.edu.cn, zhangyunyun1110@stu.ahu.edu.cn, zlling@ahu.edu.cn, ytyan@ahu.edu.cn, zhaoshuzs@ahu.edu.cn).
Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei 230009, China (e-mail: xwu@hfut.edu.cn).

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3350630

Feature Selection based on the Dependency in K nearest neighbors) [21] and OFS-A3M [22] are proposed for numerical features only. Besides, based on statistical tests, information theory, and Fisher's Z-test, OSFS (Online Streaming Feature Selection) [9], SAOLA (Scalable and Accurate On-Line Approach) [23], SFS-FI (Streaming Feature Selection considering Feature Interaction) [14], OSSFS-DD (Online Scalable Streaming Feature Selection via Dynamic Decision) [24] provide two versions of algorithms for both categorical and numerical features respectively. For mixed feature space, fuzzy rough set-based methods [25, 26] or hybrid metrics based methods [27, 28] were proposed. All these methods mentioned above implicitly assume that we can know the attribute type of each feature before learning. However, in real-world applications, the streaming features are arrived in a random order. Therefore, it is unreasonable and unrealistic to know all the attribute types for the infinite streaming features in advance.

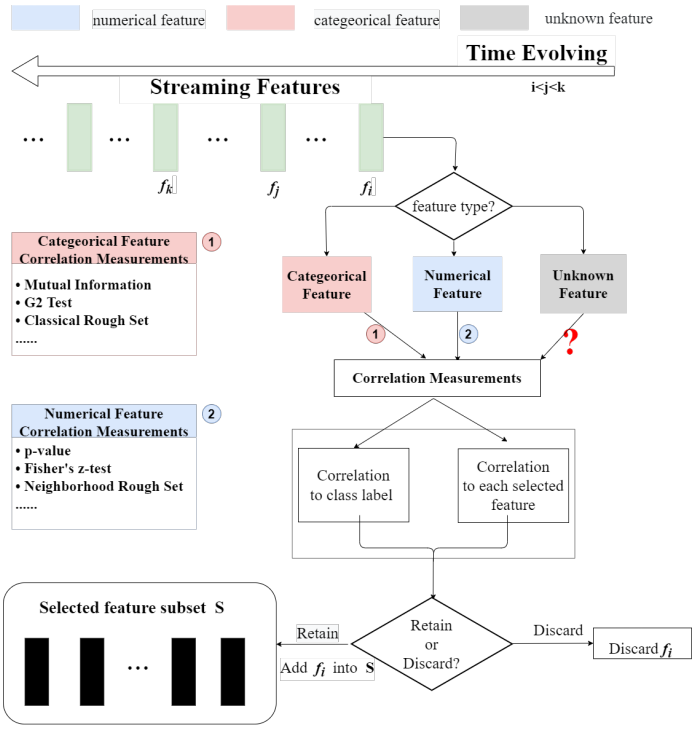


Fig. 1: Illustration of the problem of online heterogeneous streaming feature selection without the information of feature type.

As shown in Fig. 1, streaming features are being generated and arriving one by one as time goes on (from t_i to t_k). Suppose at each timestamp t, the new arriving streaming feature is f_t. Filter model streaming feature selection methods usually use specific measurements to calculate the correlation between features. Usually, streaming feature selection methods need to measure the correlation between a new arriving feature f_i and the class label C, and the correlation between f_i and each feature f' in the selected feature subset S. However, the streaming features may arrive in a random order. If we cannot know the feature type of the next arriving feature, how can we measure the correlations and decide whether to retain or discard this streaming feature? Motivated by this, this paper firstly studies a practical issue of online feature selection for the unknown type heterogeneous streaming features.

MIC (Maximal Information Coefficient), published in Science in 2011, aimed to identify interesting associations between pairs of variables for both functional and not [29]. MIC can measure the dependence between features and has two heuristic properties: generality and equitability. If a relationship exists between two variables, a grid can be drawn on the scatterplot of the two variables that partitions the data to encapsulate that relationship. MIC can examine all potentially interesting relationships in a dataset and ignore the feature types. Therefore, this paper applies MIC to measure the correlation between unknown type streaming features.

Specifically, we firstly pay attention to the issue of online heterogeneous streaming feature selection and give a formal definition of it. Based on information theory, we model the streaming feature selection issue as a minimax problem and propose two metrics to determine whether the new arriving feature should be selected. We theoretically demonstrate the validity of these two metrics. Based on these two new metrics, we propose a new online adaptive feature selection method for unknown type heterogeneous streaming features. The main contributions of this paper are as follows:

- We first present the exciting and practical issue of online heterogeneous streaming feature selection without the feature type information before learning and formally modeling it as a minimax problem.
- In terms of MIC, we derive a new metric MIC_{Gain} that can be used to determine whether a new unknown type streaming feature should be selected. To speed up the efficiency of online feature selection, we present the metric MIC_{Cor} that can directly discard new arriving features with low correlation. Meanwhile, we theoretically demonstrate the validity of these two metrics.
- Based on these two new metrics, we propose a new Online Heterogeneous Streaming Feature Selection method, named OHSFS. OHSFS is nonparametric and does not need to know the feature type of each streaming feature in advance, which is in line with practical application needs.
- Extensive experiments conducted on twenty-one real-world datasets and compared with four state-of-the-art traditional heterogeneous feature selection algorithms and five online streaming feature selection approaches indicate the effectiveness of OHSFS.

This new online heterogeneous streaming features method was first introduced in our conference paper [30]. In comparison with the preliminary conference version, we have improvements in the following three aspects: (1) We systematically elaborate on background, motivation, and related work from the heterogeneous streaming feature perspective; (2) We add two new detailed proofs (Proofs 1 and 2) to demonstrate the validity of two metrics MIC_{Gain} and MIC_{Cor}, which ensures the effectiveness of our new approach theoretically; (3) In experiments, we add more datasets and apply two new classifiers, Ensemble Learning and Neural Network, to improve the justification. Meanwhile, for each competing algorithm, we provide a more detailed analysis of the advantages and disadvantages of the comparison algorithms. In sum, the novelty of this manuscript lines in two aspects. First, at the

problem level, our new method aims to handle the issue of online heterogeneous streaming feature selection without feature type information. Second, at the algorithmic level, our proposed method can uniformly process new streaming features (numerical or categorical) without requiring their feature types at first. Therefore, OHSFS is in line with practical application needs for big data.

The rest of this article is organized as follows. In Section 2, we describe related work. In Section 3, the formal definition of the problem, the relevant theoretical knowledge of MIC, and a new method for heterogeneous streaming features without the feature type information is proposed. Section IV gives the experimental analysis. Finally, Section V gives a brief conclusion.

2 Related Work

Feature selection has been studied for many years and a large number of excellent algorithms have been proposed [6, 31, 32]. According to different data generation, we can divide feature selection into two categories: traditional feature selection for static data and online streaming feature selection for stream data [3].

2.1 Traditional Feature Selection Methods

According to the feature type of a dataset, feature selection methods can be divided into homogeneous (categorical or numerical) feature selection and heterogeneous (including both categorical and numerical) feature selection [16, 17, 33]. Most traditional filter model feature selection algorithms are designed for a single feature type, i.e., categorical or numerical. For example, MI [34] and classical rough set dependence degree [35] are two commonly used measurements for categorical feature selection. The Laplacian score [36] and Fisher score [37] are usually used in feature selection for numerical data. These measurements aim to calculate the correlation between the conditional feature and the class label and select the top features according to their scores.

In practical applications, features may be gathered in mixed types. That is, there are both categorical and numerical features in the dataset. Therefore, some traditional mixed feature selection algorithms are proposed to deal with heterogeneous feature space. Specifically, Zhang et al. [27] constructed a new information entropy measurement method based on fuzzy rough set theory for the mixed feature selection problem and proposed a new filter-wrapper model feature selection algorithm according to this measurement criterion. Yuan et al. [25] proposed the FRUAR algorithm for the feature selection problem of unsupervised mixed data. FRUAR is based on fuzzy rough sets to define the importance of a single feature and then designed a heuristic search algorithm to find the optimal feature subset. Yuan et al. [26] solved the feature interaction problem in the feature selection of unsupervised imbalanced mixed data and proposed a measure of uncertainty based on fuzzy complementary entropy, named EUIAR. In the work of Kim et al. [38], the feature space is divided into two subspaces according to the proposed method, corresponding to the numerical feature space and the categorical feature space respectively, and then the subspaces are sorted respectively, and finally the classification error rate is used to select the most useful feature subset. Wang et al. [39] designed an efficient hybrid feature selection algorithm based on the idea of feature space decomposition and fusion for the feature selection problem on large-scale hybrid feature space datasets. Inspired by the spectral feature selection method, S Solorio-Fernández et al. [40] proposed a new unsupervised filter feature selection method that can handle mixed features by combining the kernel function and a new spectral function-based feature measurement method. The attribute evaluation criteria of maximal information, minimal redundancy, and maximal interactivity are developed based on the proposed uncertainty measure. For mixed feature type datasets, mixed feature selection methods use different metrics to decrease the information loss in the feature space. However, these methods require complete knowledge of the feature space before learning.

Besides, many new traditional feature selection algorithms have been proposed. For text classification tasks, Labani et al. [41] proposed the MORDC algorithm, which focuses on searching in the solution space using a multi-objective evolutionary framework. Meanwhile, many population-based optimization methods have been proposed and successfully used to solve different optimization problems. Hamedmoghadam et al. [42] proposed a population-based feature selection method, named OFBO, which builds on the information diffusion mechanism of the bounded confidence model, searching the feasible solution space through the opinion formation process. Sharkawy et al. [43] optimized the selection of input features based on particle swarm optimization. SVM classifier based on PSO-based feature selection is proposed for detecting contaminated particles in transformer oil. To handle the challenge of the "curse of dimensionality" for evolutionary feature selection methods, Song et al. [44] proposed a new variable-size cooperative coevolutionary particle swarm optimization algorithm for feature selection. The proposed algorithm employs the idea of "divide and conquer" in a cooperative coevolutionary approach, and extensive experiments indicate the capability of obtaining good feature subsets. Furthermore, Song et al. [45] proposed a new hybrid feature selection algorithm using surrogate sample-assisted particle swarm optimization. Since the whole sample set is replaced by a small number of surrogate units, the proposed algorithm significantly reduces the cost of evaluating particles in particle swarm optimization.

Not only that, for the multi-label feature selection problem, Liu et al. [46] first constructed a label enhancement method based on instance information distribution, then reconstructed a neighborhood rough set model suitable for label distribution learning and finally proposed a feature-based forward greedy feature selection algorithm LDRS. However, LDRS also faces the problem of the relatively high time complexity of the algorithm.

2.2 Online Streaming Feature Selection Methods

For some real-world applications, features may exist in a streaming model, and we cannot know the whole feature space before learning [7, 8, 10]. Therefore, many online feature selection methods have been proposed to solve the issue of online streaming feature selection [12].

Specifically, Zhou et al. [18] proposed the Alpha-investing algorithm, which does not require a global model. However,

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3350630

Alpha-investing requires prior knowledge of the feature space structure to control the process of candidate feature selection heuristically. Wu et al. [9] proposed an online streaming feature selection framework, which includes two algorithms: OSFS and Fast-OSFS. OSFS mainly includes two steps: online correlation analysis, and online redundancy analysis. The online correlation analysis is to discard irrelevant features, and the online redundancy analysis eliminates redundant features. Yu et al. [23] proposed the SAOLA method for high-dimensional data by using a pairwise comparison method based on mutual information theory. Rahmaninia et al. [47] used a streaming method to evaluate the correlation and redundancy of features based on mutual information theory and proposed two online feature selection algorithms, named OSFSMI and OSFOMI-k. Zhou et al. [14] proposed a streaming feature selection algorithm SFS-FI considering the interaction between features, and the number of selected features increased due to the consideration of the interaction ability between features. Gao et al. [48] proposed a unified feature selection framework including three low-order information-theoretic terms for multi-label learning named Selected Terms of Feature Selection (STFS). The algorithm is designed primarily for multiple variables while taking into account high-order variable correlations. Rafie et al. [49] used mutual information theory and Pareto optimal set theory to select streaming features using a multi-objective search strategy to solve the problem that traditional multi-label feature selection cannot be applied to stream data scenarios.

Most existing streaming feature selection methods are designed for a single feature type or provide two versions of algorithms for both categorical and numerical features, respectively. However, besides the number of streaming features in practical applications, their feature type may also be unknown in advance. Therefore, this paper focuses on heterogeneous streaming features without the feature type information.

3 The Proposed Framework

This section describes the formal definition of the problem and the specific implementation of the proposed method. We summarize some symbols used in this paper in Table 1.

TABLE 1: Summary on Mathematical Notations

Notations	Definition
D	Target dataset
F	Feature space
C	Class label
$\|\cdot\|$	$\|S\|$: the size of set S
x_i	i^{th} sample
f_j	j^{th} feature
U	Sample space: $\{x_1, x_2, ..., x_n\}$
S_t	The selected feature subset after time stamp t
$MI(\cdot;\cdot)$	$MI(f;C)$: denote the mutual information between feature f and class label C

3.1 Problem Definition

Definition 1. Online Heterogeneous Streaming Feature Selection

Suppose F is the conditional feature space of the target dataset D, the class label is C, and the sample space is $U = \{x_1, x_2, ..., x_n\}$, where x_i is the i^{th} sample. For online heterogeneous streaming feature selection, we cannot known the exact number of $|F|$ in advance (e.g. $|F| \to \infty$). At timestamp t, the new arriving streaming feature is f_t ($f_t \in F$), and we do not know the attribute type of f_t. Meanwhile, we must decide whether to retain or discard the new arrival feature on the fly, and the selected feature subset after timestamp t is S_t. Streaming feature selection aims to maximize the information of S_t at each timestamp while making the size of $|S_t|$ as small as possible.

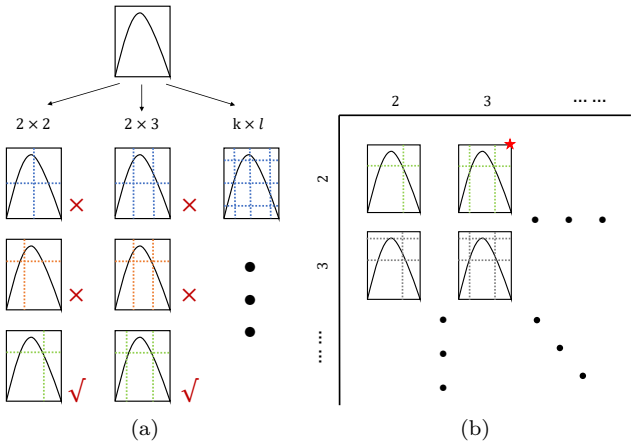


Fig. 2: Taking a parabola as an example, a schematic diagram of calculating MIC. (a) shows that for each pair (k, l), the MIC algorithm finds the k-by-l grid with the highest mutual information. (b) shows the maximum mutual information matrix $M(X;Y)$ composed of the highest mutual information value obtained by each pair (k, l).

Mutual information can measure the amount of information shared between S_t and C by measuring their dependency level [50]. Therefore, in terms of information theory, online streaming feature selection can be formalized as:

$$\begin{aligned} &min_{|S_t|} max\{MI(S_t;C)\} \\ &s.t. \quad |S_t| > 0. \end{aligned} \tag{1}$$

Similar to traditional feature selection methods, two main issues for streaming feature selection can be distinguished: feature measurement and search strategy [50]. This first one is to define an appropriate measure function to calculate the correlation for each new arriving feature. The second issue is to develop a search strategy that can decide whether retain or discard each streaming feature. There are many measure functions, such as Pearson Correlation Coefficient (PCC) [51], Spearman's Rank Correlation Coefficient (SPCC) [52] and Mutual Information (MI) [34], etc. However, most existing feature measure functions must know the feature type before calculation. Therefore, first of all, we need a measure function to calculate the correlation between unknown type streaming features.

3.2 Measure Function for Unknown Type Features

MIC has been proved to be an effective measure of the dependence of two variables and can capture a wide range of both functional and unfunctional associations [29]. As shown in Fig.2, MIC divides the variables x, y into $k * l$ grids over the whole coordinate system and finds the k-by-l girds with

the highest mutual information. Meanwhile, the x-axis and y-axis axes are divided dynamically in the calculation of the MIC. Therefore, MIC can calculate mutual information for both numerical and categorical data, making it adaptable to various applications.

Specifically, given a two-dimensional variable dataset $D = [X, Y] = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. The integers (k, l) can be any pair. The calculation of the $MIC(X;Y)$ is as follows [29]:

$$MIC(X;Y) = max\{M(X;Y)_{k,l}\}, \tag{2}$$

$$M(X;Y)_{k,l} = \frac{MI(X;Y|_{k,l})}{log(min\{k,l\})}, \tag{3}$$

where $MI(X;Y|_{k,l})$ denotes the mutual information value $MI(X;Y)$ divided according to the integers (k, l) on the two-dimensional variable dataset D. The size of k and l when the party mutual information is the maximum value can be obtained by the exhaustive method. $k \times l \leq B(n)$, B is a function of the sample size n expressed as $B(n) = n^{0.6}$.

MIC can measure the correlation between two variables of any type. A higher MIC value indicates a strong correlation between variables, and conversely, a lower MIC value implies a weak correlation between variables.

Let $S = [f_1, f_2, ..., f_N]$ be an N dimensional feature vector and C is the class label. MIC measures the amount of information shared between S and C by measuring their degree of correlation. Denote the joint distribution densities of S and C and their marginal distributions by $P(S, C)$, $P(S)$, and $P(C)$, respectively. The MIC between features and class label can be defined as follows:

$$\begin{aligned} MIC(S;C) &= MIC(f_1, f_2, ..., f_N; C) \\ &= \int P(S,C) log \frac{P(S,C)}{P(S)P(C)} d_S d_C. \end{aligned} \tag{4}$$

Although mutual information measurement [53] has good theoretical performance, accurate estimation of mutual information is impossible. Because to compute (5), the estimation of $P(S, C)$ is unavoidable, which is an NP-hard problem. Therefore, several approximations of Eq(5) have been proposed. The most representative method is the mRMR (Minimal Redundancy Maximal Relevance) [54] as

$$MI(S;C) = \sum_{i=1}^{N} MI(f_i;C) - \frac{1}{N-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} MI(f_i;f_j). \tag{5}$$

3.3 Seach Strategy for Streaming Features

Unlike traditional feature selection methods that actively search for optimal features, streaming feature selection can only passively receive streaming features and decide whether to retain or discard these features. At each timestamp, the ultimate goal of unknown type streaming feature selection is to maximize $MIC(S_t; C)$.

Suppose at timestamp t, the selected feature subset is S_t. It is impossible to calculate the information between a feature set S_t and a class label C directly [29]. Therefore, a more commonly used approach is to approximate it. To propose a new approximation, we formulate the unknown type streaming feature selection as:

$$max\{S_t^T Q_t S_t\}, \tag{6}$$

where Q_t is a symmetric information matrix constructed from the mutual information terms in as:

$$Q_t = \begin{bmatrix} MIC(f_1;C) & ... & -\frac{\beta}{2}MIC(f_1;f_N) \\ -\frac{\beta}{2}MIC(f_1;f_2) & ... & -\frac{\beta}{2}MIC(f_2;f_N) \\ ... & ... & ... \\ -\frac{\beta}{2}MIC(f_1;f_N) & ... & MIC(f_N;C) \end{bmatrix}, \tag{7}$$

where $S_t = [s_1, ..., s_N]^T$ is the selected feature vector, $s_i \in \{0, 1\}$, and β is a trade-off parameter.

At timestamp $t + 1$, suppose the new arriving feature is f_{t+1}, and we add f_{t+1} into the candidate feature subset. That is, the selected feature subset is $S_{t+1} = [S_t, f_{t+1}]$. If

$$S_{t+1}^T Q_{t+1} S_{t+1} > S_t^T Q_t S_t, \tag{8}$$

then, f_{t+1} can be retained. Otherwise, we should remove f_{t+1} from S_{t+1}.

Therefore, the condition for judging whether f_{t+1} should be selected is

$$S_{t+1}^T Q_{t+1} S_{t+1} - S_t^T Q_t S_t > 0. \tag{9}$$

Definition 2. To determine whether retain or discard the new arriving feature f_t at timestamp t, we define the metric MIC_{Gain} as follows:

$$MIC_{Gain}(f_t, S_{t-1}) = MIC(f_t;C) - \frac{1}{|S_{t-1}|} \sum_{f_i \in S_{t-1}} MIC(f_i;f_t), \tag{10}$$

where S_{t-1} is the selected feature subset at timestamp $t-1$, and C the class label.

Theorem 1. At timestamp $t + 1$, if $MIC_{Gain}(f_{t+1}, S_t) > 0$, then $S_{t+1}^T Q_{t+1} S_{t+1} > S_t^T Q_t S_t$.

Proof 1. Now let's start with $N = 2$ at timestamp t_2 and approximate the MIC value between the set S_2 and the label C as:

$$S_2 = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, N = 2$$

$$Q_2 = \begin{bmatrix} MIC(s_1;C) & -\frac{\beta}{2}MIC(s_1;s_2) \\ -\frac{\beta}{2}MIC(s_1;s_2) & MIC(s_2;C) \end{bmatrix}$$

$$\begin{aligned} S_2^T Q_2 S_2 &= \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}^T \begin{bmatrix} MIC(s_1;C) & -\frac{\beta}{2}MIC(s_1;s_2) \\ -\frac{\beta}{2}MIC(s_1;s_2) & MIC(s_2;C) \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \\ &= \begin{bmatrix} s_1 MIC(s_1;C) \\ -\frac{\beta}{2}s_1 MIC(s_1;s_2) + s_2 MIC(s_2;C) \end{bmatrix}^T \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \\ &= s_1^2 MIC(s_1;C) + s_2^2 MIC(s_2;C) - \beta s_1 s_2 MIC(s_1;s_2) \end{aligned}$$

Then at timestamp t_3, suppose we select f_3 and approximate the MIC value between S_3 and C as:

$$S_3 = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}, N = 3$$

$$Q_3 = \begin{bmatrix} MIC(s_1;C) & -\frac{\beta}{2}MIC(s_1;s_2) & -\frac{\beta}{2}MIC(s_1;s_3) \\ -\frac{\beta}{2}MIC(s_1;s_2) & MIC(s_2;C) & -\frac{\beta}{2}MIC(s_2;s_3) \\ -\frac{\beta}{2}MIC(s_1;s_3) & -\frac{\beta}{2}MIC(s_2;s_3) & MIC(s_3;C) \end{bmatrix}$$

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3350630

$$
\begin{aligned}
&S_3^T Q_3 S_3 \\
&= \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}^T \begin{bmatrix} MIC(s_1;C) & -\frac{\beta}{2}MIC(s_1;s_2) & -\frac{\beta}{2}MIC(s_1;s_3) \\ -\frac{\beta}{2}MIC(s_1;s_2) & MIC(s_2;C) & -\frac{\beta}{2}MIC(s_2;s_3) \\ -\frac{\beta}{2}MIC(s_1;s_3) & -\frac{\beta}{2}MIC(s_2;s_3) & MIC(s_3;C) \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \\
&= \begin{bmatrix} s_1 MIC(s_1;C) - \frac{\beta}{2}s_2 MIC(s_1;s_2) - \frac{\beta}{2}s_3 MIC(s_1;s_3) \\ -\frac{\beta}{2}s_1 MIC(s_1;s_2) + s_2 MIC(s_2;C) - \frac{\beta}{2}s_3 MIC(s_2;s_3) \\ -\frac{\beta}{2}s_1 MIC(s_1;s_3) - \frac{\beta}{2}s_2 MIC(s_2;s_3) + s_3 MIC(s_3;C) \end{bmatrix}^T \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \\
&= s_1^2 MIC(s_1;C) + s_2^2 MIC(s_2;C) + s_3^2 MIC(s_3;C) \\
&- \beta s_1 s_2 MIC(s_1;s_2) - \beta s_1 s_3 MIC(s_1;s_3) - \beta s_2 s_3 MIC(s_2;s_3)
\end{aligned}
$$

Therefore, in terms of (9), we calculate the information gain as:

$$
\begin{aligned}
&S_3^T Q_3 S_3 - S_2^T Q_2 S_2 \\
&= s_3^2 MIC(s_3;C) - \beta * s_1 s_3 MIC(s_1;s_3) - \beta * s_2 s_3 MIC(s_2;s_3)
\end{aligned}
$$

By analogy, we can get

$$
\begin{aligned}
&S_{t+1}^T Q_t S_{t+1} - S_t^T Q_t S_t \\
&= s_{t+1}^2 MIC(s_{t+1};C) - \beta \sum_{i=1}^{t} s_i s_{t+1} MIC(s_i;s_{t+1})
\end{aligned}
$$

Because $s_i \in \{0,1\}$, $s_i^2 = s_i$. Then,

$$
\begin{aligned}
&S_{t+1}^T Q_t S_{t+1} - S_t^T Q_t S_t \\
&= MIC(s_{t+1};C) - \beta \sum_{i=1}^{t} MIC(s_i;s_{t+1})
\end{aligned}
$$

In our proposed metric, the variable β is set to reciprocal of the number of selected features. Then,

$$S_{t+1}^T Q_t S_{t+1} - S_t^T Q_t S_t = MIC_{Gain}(f_{t+1}, S_t)$$

Thus, at timestamp $t+1$, if $MIC_{Gain}(f_{t+1}, S_t) > 0$, then $S_{t+1}^T Q_{t+1} S_{t+1} > S_t^T Q_t S_t$.

The value of MIC_{Gain} determines the importance of newly arrived feature f_t to the currently selected subset S_{t-1} at timestamp t. If MIC_{Gain} is greater than 0, the newly arrived feature is positive for the complete information of the selected subset; otherwise, the value of MIC_{Gain} is less than 0.

For streaming feature selection, the speed of the algorithm is critical. Because MIC needs to divide the variables into multiple grids, the time complexity of MIC is a bit high. Besides, in practical applications, there are always many irrelevant or low correlation features.

Definition 3. To speed up the online streaming feature selection, we propose a new metric MIC_{Cor} to discard these irrelevant and low correlation features directly as follows:

$$MIC_{Cor}(S, C) = \frac{1}{|S|} \sum_{f_i \in S} MIC(f_i; C), \tag{11}$$

where MIC_{Cor} is the mean correlation of each features in the currently selected feature subset.

In other words, MIC_{Cor} aims to filter out low correlation features and maximize the correlation of the selected subset

$$max\{MIC_{Cor}(S_t, C)\}. \tag{12}$$

Corollary 1. For a new arriving feature f_t, if $MIC(f_t;C)$ is samller than $MIC_{Cor}(S_{t-1}, C)$, then it can be discarded directly.

Proof 2. Suppose at timestamp t, the new arriving feature is f_t and the selected feature set is S_{t-1}. If $MIC(f_t;C) \leq MIC_{Cor}(S_{t-1}, C)$, the selection of f_t will decrease $MIC_{Cor}(S_t, C)$.

Let $|S_{t-1}| = N_{t-1}$ and $MIC_{Cor}(S_{t-1}, C) = Cor_{t-1}$. Then $\sum_{f_i \in S_{t-1}} MIC(f_i;C) = N_{t-1} \times Cor_{t-1}$. If we add f_t into S, $S_t = S_{t-1} \cup f_t$ and $|S_t| = N_{t-1} + 1$.

$$
\begin{aligned}
MIC_{Cor}(S_t, C) &= \frac{1}{|S_t|} \sum_{f_i \in S_t} MIC(f_i;C) \\
&= \frac{1}{N_{t-1}+1}(N_{t-1} \times Cor_{t-1} + MIC(f_t;C)) \\
&= Cor_{t-1} + \frac{1}{N_{t-1}+1}(MIC(f_t;C) - Cor_{t-1})
\end{aligned}
$$

Because $MIC(f_t;C) \leq MIC_{Cor}(S_{t-1}, C)$, then $MIC(f_t;C) - MIC_{Cor}(S_{t-1}, C) \leq 0$. Thus, $MIC_{Cor}(S_t, C) \leq MIC_{Cor}(S_{t-1}, C)$.

Therefore, to maximize the correlation of the selected feature subset, we can discard the low correlation streaming features safely and directly in terms of MIC_{Cor}.

3.4 The Proposed Algorithm

To sum up, in terms of (10) and (11), we propose a new online heterogeneous streaming feature selection algorithm for unknown type streaming features as Algorithm 1.

More specifically, if a new feature f_t arrives at timestamp t, Steps 5-8 calculates the correlation values between f_t and C, then compares $MIC(f_t;C)$ to $Mean_S$, and selects the features with high correlation for the further evaluation processes. Steps 9-12 decide whether the newly arrived feature f_t is important for the candidate feature subset. If $MIC_{Gain}(f_t, S) > 0$, which mean the new feature f_t can increase the information of selected feature subset, we add f_t into subset S. With this new online streaming feature selection algorithm, we can select features with high correlation and high significance while ignoring the feature type of each streaming feature. Besides, it is worth mentioning that our algorithm does not need to set any parameters in advance.

Algorithm 1 Online Heterogeneous Streaming Feature Selection (OHSFS)

Input:
  F: the condition feature set;
  C: the class attributes;
Output:
  S: the selected feature set;
1: Initialization:$S = \{\}$;
2: $MIC_{Cor}(S, C)$:the mean correlation of features in S, initialized to 0;
3: Repeat
4: Get a new arriving feature f_t at time stamp t;
5: IF $MIC(f_t;C) \leq MIC_{Cor}(S, C)$
6: Discard feature f_t;
7: Continue;
8: End IF
9: IF $MIC_{Gain}(f_t, S) > 0$
10: $S = S \cup \{f_t\}$;
11: End IF
12: Until no more features are available;
13: Output selected features contained in S.

3.5 Time Complexity

Here is an estimation of the time complexity of the algorithm OHSFS. Let m and n be the numbers of features and samples for the target dataset, respectively. When calculating $MI(D,k,l)$, in order to avoid grid exhaustive cutting, perform traversal optimization, and reduce the computational complexity of the MIC method, the literature [29] proposed a dynamic programming algorithm to approximate the solution of the MIC. The number of cycles of equal depth division is $B/2$, i is the number of segments for the $y-axis$. so its overall complexity is $O(i^2klB) = O(k^2B^2)$. Therefore, we assume that the time complexity of MIC is constant $O(\Omega)$.

At time stamp t, suppose that the number of selected features is $|S_t|$. Steps 5-8 calculate the correlation between the new streaming feature and the class label, then compare $MIC(f_t;C)$ to $Mean_S$ and select the features with high correlation. The time complexity of these steps is $O(\Omega)$. Steps 9-12 calculate the MIC between the new arriving feature and each selected feature in S. If $MIC_{Gain}(f_t,S_t) > 0$, we add f_t into S_t. Therefore, the time complexity of steps 9-12 is $O(m*|S|*\Omega)$. In sum, the worst time complexity of OHSFS is $O(m^2\Omega)$ when we select all the streaming features. However, there are always many low correlation features for real-world datasets, and it is impossible for all features to increase the information of the selected feature subset. Thus, the time complexity of OHSFS will be much smaller than $O(m^2\Omega)$.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

This section applies the proposed online streaming feature selection method (OHSFS) and competing algorithms on twenty-one real-world datasets. The details of these datasets are shown in Table 2 [1].

We compare OHSFS with four state-of-the-art traditional mixed feature selection methods on the first four small datasets in Table 2. Meanwhile, we compare OHSFS with five state-of-the-art online streaming feature selection algorithms on the last sixteen datasets in Table 2. There are three main reasons for this: 1) The time complexity of these four traditional mixed feature selection methods is too high to be applied to high-dimensional datasets. 2) The dimensions of the first four datasets are too small to match the online streaming feature selection scenarios. Therefore, we do not conduct the experimental competition of online streaming feature selection methods on the first four small datasets. 3) The first four datasets are mixed, while most of the last sixteen are single feature types. Our new method can handle mixed datasets, but the five competing online streaming feature selection algorithms cannot handle mixed datasets directly. This is another reason we do not conduct the experimental competition of online streaming feature selection methods on the first four small datasets.

4.1.2 Evaluation Metrics

Because our new proposed method and all these competing feature selection algorithms are implemented in MATLAB, we use five built-in classifiers, KNN (k = 3), SVM (with the linear kernel), CART, Ensemble Learning, and Neural Network, in MATLAB r2023b to conduct the experiments to require the predictive accuracy and running time for the fairness. We perform a 5-fold cross-validation on each dataset. Feature selection is to train on 4/5 of the data samples and test on the remaining 1/5 of the samples. All competing algorithms use the same training and test sets. For each dataset, the order of stream features is random. We run each dataset ten times and recorded the average prediction accuracy, running time, and the mean number of features selected on each classifier.

To verify whether the prediction accuracy of OHSFS and its competitors on different classifiers is significantly different, we performed the Friedman test at 95% significance level under the null hypothesis [55]. If the null hypothesis is rejected, there is a significant difference in the performance of OHSFS and its competitors. When the null hypothesis of the Friedman test was rejected, we proceeded to the Nemenyi test as a post-hoc test [55].

All experimental results are conducted on a PC with AMD 5800X, 3.8 GHz CPU, and 16 GB memory.

TABLE 2: Real-world Datasets

Dataset	instances	Features	Classes	Feature Type
German	1000	20	2	mixed
Heart	303	13	2	mixed
Australian	690	14	2	mixed
FLags	358	29	7	mixed
Dermatology	358	34	6	real
Arrarrhythmia	452	279	16	mixed
LYMPHOMA	62	4026	3	real
SRBCT	63	2308	4	real
DLBCL	77	6285	2	real
CAR	174	9182	11	real
OVARIAN	253	15154	2	real
LEU	72	7129	2	real
PROSTATE	102	6033	2	real
ARCENE	200	10000	2	real
LUNG2	203	3312	5	real
LUNG	181	12533	2	real
SYLVA	216	14394	2	mixed
GISETTE	7000	5000	2	integer
DEXTER	600	20000	2	integer
HIVA	4229	1617	2	categorical
NOVA	1929	16969	2	categorical

4.1.3 Parameter Setting of Competing Algorithms

In this section, we summarize the parameter settings of all these compared algorithms in our experiments.

We compare OHSFS with four state-of-the-art traditional mixed feature selection methods, including ε-approximate reduct [27], IFSM [28], EUIAR [26], and FRUAR [25]. ε-approximate reduct is a fuzzy rough set-based filter-wrapper model feature selection method. We set $\varepsilon = 0.5$ according to the source code of this algorithm. IFSM is a neighborhood rough set-based method, and the parameter values δ are considered from $[0.1, 0.4]$. EUIAR is an unsupervised hybrid feature selection method based on fuzzy complementary entropy. The parameter value of EUIAR is set to 1 as the default value from the paper [26]. FRUAR is an unsupervised hybrid feature selection method based on fuzzy rough sets that find the optimal subset of features through a heuristic search algorithm. We set the parameter value of FRUAR to 0.1 as the default value [25].

[1] Public available at https://archive.ics.uci.edu/, and http://www.cs.binghamton.edu/ lyu/KDD08/data/.

Besides, we compare OHSFS with five state-of-the-art online streaming feature selection algorithms, including α-investing [18], Fast-OSFS [9], SAOLA [23], OSFSMI [47], and SFS-FI [14]. α-investing is a streamwise feature selection algorithm, and we set the two parameters $WEALTH$ and $DELTA\_ALPHA$ to 0.5 as the default value from the paper [18]. Fast-OSFS is an online streaming feature selection algorithm based on Markov blankets to discard redundant features quickly, and we set the significance level of α to 0.01 [9]. SAOLA is a scalable and accurate online feature selection method for ultra-high dimensional datasets, and we set its parameter to 0.05 [23]. OSFSMI is based on mutual information theory that does not need specific parameter values. SFS-FI is a streaming feature selection algorithm that considers the interaction between features, and we set its parameter to 0.05 as the default value [14].

4.2 The effectiveness of MIC_{Cor}

We present MIC_{Cor} to directly discard low correlation features for speeding up the efficiency of OHSFS. To validate the effectiveness of MIC_{Cor}, we perform experiments between OHSFS(with MIC_{Cor}) and $\text{OHSFS}_{\text{noCor}}$(without MIC_{Cor}) on six datasets, including three low-dimensional datasets(German, Heart, Australian) and three high-dimensional datasets (SRBCT, DLBCL,PROSTATE). We run each dataset ten times and record the average prediction accuracy, running time, and the mean number of features selected on each classifier.

TABLE 3: Predictive Accurary Using Different Classifiers

Dataset	KNN		SVM		CART	
	OHSFS	$\text{OHSFS}_{\text{noCor}}$	OHSFS	$\text{OHSFS}_{\text{noCor}}$	OHSFS	$\text{OHSFS}_{\text{noCor}}$
German	0.7132	0.7084	0.6987	0.7272	0.7155	0.7117
Heart	0.7759	0.8037	0.8130	0.8344	0.7448	0.7763
Australian	0.8399	0.8365	0.8551	0.8551	0.8243	0.8242
SRBCT	0.9607	0.9310	0.9564	0.9730	0.8836	0.8448
DLBCL	0.9107	0.9280	0.9547	0.9640	0.8267	0.8293
PROSTATE	0.8967	0.8833	0.8767	0.8967	0.85	0.82

TABLE 4: Running Time (seconds) and the Mean Number of Selected Features

Dataset	Running Time		# Seleted Features	
	OHSFS	$\text{OHSFS}_{\text{noCor}}$	OHSFS	$\text{OHSFS}_{\text{noCor}}$
German	0.0873	0.2353	2.02	3.98
Heart	0.0730	0.1128	6.64	8.12
Australian	0.4444	0.7088	7.28	8.38
SRBCT	0.8209	857.036	18.4	791.88
DLBCL	14.7649	2172.253	144.06	815.68
PROSTATE	6.7207	2208.629	46	335.87

On predictive accuracy, the algorithm without MIC_{Cor} selects more features and is slightly better than the original algorithm. However, there is a dramatic increase in the running time for the algorithm without MIC_{Cor}. For example, the running time of $\text{OHSFS}_{\text{noCor}}$ is more than 1,000 times than OHSFS on dataset SRBCT, with a loss of around 0.03% in predictive accuracy. Therefore, we apply MIC_{Cor} into our algorithm to speed up the efficiency, with the expense of some loss in predictive accuracy.

4.3 OHSFS vs. Traditional Mixed Feature Selection Methods

In this section, we compare OHSFS with four state-of-the-art traditional mixed feature selection methods including ε-approximate reduct [27], IFSM [28], EUIAR [26], and FRUAR [25]. All algorithms are implemented in MATLAB. Since the extremely long running time of these four algorithms on high-dimensional datasets, we only conduct the experiments on the first five small datasets as shown in Table 2.

Tables 5-9 summarize the predictive accuracy on different classifiers, the running time, and the mean number of selected features of these competing algorithms. The p-values of Friedman test on KNN, SVM, CART, running time and the mean number of selected features are 0.221e-05, 0.366e-05, 0.0038, 0.113e-09 and 0.0271 respectively. Thus, there is a significant difference between OHSFS and the other four competing algorithms on predictive accuracy, running time, and the mean number of selected features. According to the Nemenyi test, the value of CD is 2.7294. Fig. 3 shows the statistical test of these competing algorithms in cases of KNN, SVM, and CART.

TABLE 5: Predictive Accuracy Using KNN as the Classifier

Dataset	IFSM	ε-approximate	EUIAR	FRUAR	OHSFS
German	0.6436(3)	0.6981(2)	0.613(4)	0.5083(5)	0.7009(1)
Heart	0.7519(1)	0.747(2)	0.5478(4)	0.5341(5)	0.7241(3)
Australian	0.7625(3)	0.8308(1)	0.4449(5)	0.6194(4)	0.8287(2)
Flags	0.4098(2)	0.3726(4)	0.3742(3)	0.3516(5)	0.5649(1)
Dermatology	0.8411(3)	0.9632(1)	0.3617(4)	0.3475(5)	0.9466(2)
AVG.	0.6818	0.7222	0.4683	0.4722	0.753
AVG. RANKS	2.4	2	4	4.8	1.8

TABLE 6: Predictive Accuracy Using SVM as the Classifier

Dataset	IFSM	ε-approximate	EUIAR	FRUAR	OHSFS
German	0.7(3)	0.7344(1)	0.6996(4)	0.3897(5)	0.7035(2)
Heart	0.7837(2)	0.8107(1)	0.7056(4)	0.4822(5)	0.7563(3)
Australian	0.7897(3)	0.8551(1)	0.4449(5)	0.8191(4)	0.8551(1)
FLags	0.4005(1)	0.3711(2)	0.3366(3)	0.2892(5)	0.302(4)
Dermatology	0.8651(3)	0.9595(1)	0.4539(4)	0.2978(5)	0.9407(2)
AVG.	0.7078	0.7462	0.5281	0.4556	0.7115
AVG. RANKS	2.6	1.3	4	4.6	2.5

TABLE 7: Predictive Accuracy Using CART as the Classifier

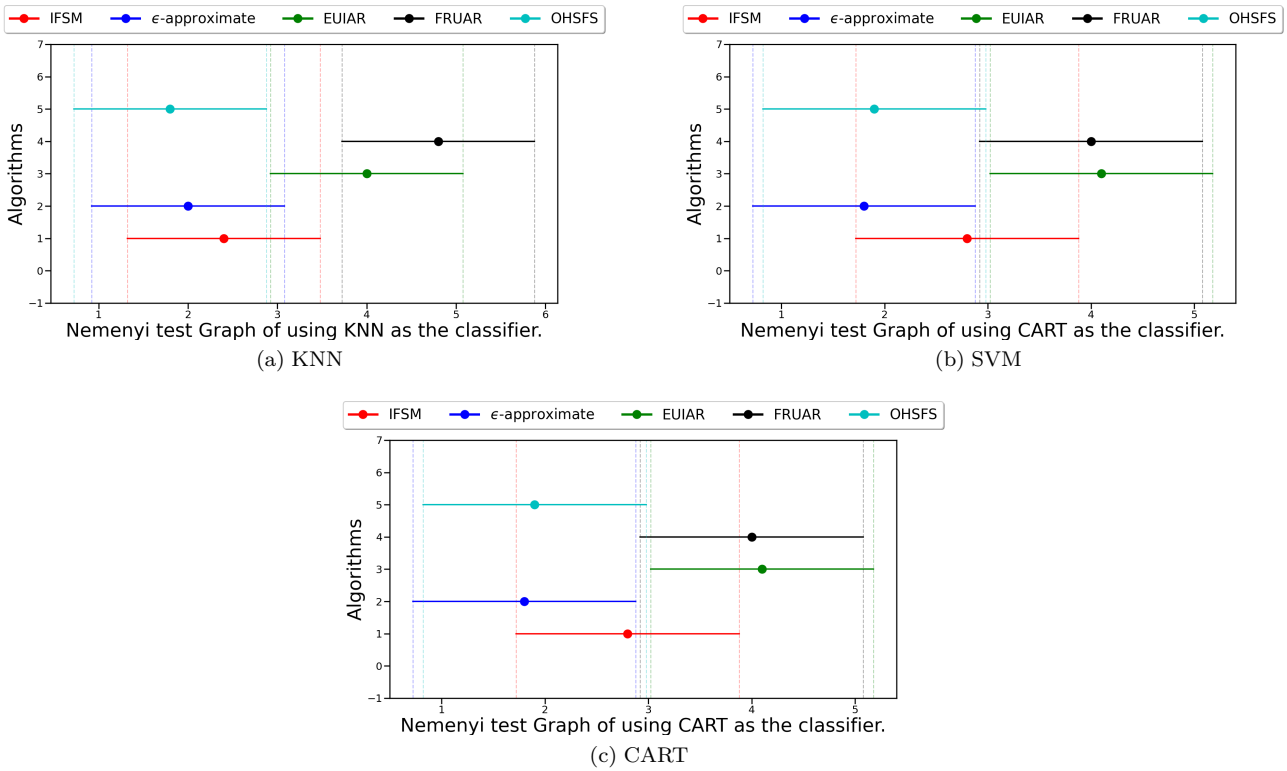
Dataset	IFSM	ε-approximate	EUIAR	FRUAR	OHSFS
German	0.6277(4)	0.6854(3)	0.6922(2)	0.5794(5)	0.7046(1)
Heart	0.747(2)	0.7848(1)	0.6974(3)	0.6004(5)	0.6974(3)
Australian	0.761(3)	0.8475(1)	0.4464(5)	0.7129(4)	0.832(2)
FLags	0.5007(2)	0.4484(3)	0.339(5)	0.4346(4)	0.5428(1)
Dermatology	0.8612(3)	0.9316(1)	0.4419(5)	0.8084(4)	0.9111(2)
AVG.	0.6995	0.7395	0.5234	0.6271	0.7376
AVG. RANKS	2.8	1.8	4.1	4.4	1.9

TABLE 8: Running time(seconds)

Dataset	IFSM	ε-approximate	EUIAR	FRUAR	OHSFS
German	0.1102(1)	2.7083(3)	7.2998(4)	342.2607(5)	0.2638(2)
Heart	0.0041(1)	0.0574(2)	0.1413(4)	1.9774(5)	0.1096(3)
Australian	0.0179(1)	0.6145(2)	1.4561(4)	105.0164(5)	0.7406(3)
FLags	0.0127(2)	0.1685(3)	1.8246(4)	3.6457(5)	0.0122(1)
Dermatology	0.0422(1)	0.5662(3)	3.4394(4)	21.4266(5)	0.0747(2)
AVG.	0.03742	0.823	2.8322	94.8654	0.2402
AVG. RANKS	1.2	2.6	4	5	2.2

From Tables 5-9, we can observe that:

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3350630



(a) KNN

(b) SVM

(c) CART

Fig. 3: The statistical test graph of OHSFS vs. traditional mixed feature selection algorithms

TABLE 9: The mean number of selected features

Dataset	IFSM	ε-approximate	EUIAR	FRUAR	OHSFS
German	9.04(3)	11.58(4)	3(2)	16.2(5)	2(1)
Heart	4.46(2)	6(4)	3(1)	11.66(5)	5.22(3)
Australian	6.62(3)	6(2)	3(1)	12.76(5)	7(4)
FLags	7.88(4)	9.28(5)	3(2)	6.86(3)	1(1)
Dermatology	8.2(2)	17.96(4)	3(1)	13.66(3)	20.52(5)
AVG.	7.24	10.164	3	12.228	7.148
AVG. RANKS	2.8	3.8	1.4	4.2	2.8

- OHSFS *vs*. IFSM: OHSFS gets higher average predictive accuracy and lower average ranks than IFSM in cases of KNN, SVM, and CART. IFSM is faster than OHSFS in running time and selects almost the same average number of features. IFSM is a neighborhood rough set-based incremental feature selection method to handle the dynamics of an object set that involves the change of a single object and multiple objects. Since the time complexity of the rough set model is square to the number of instances, IFSM is not capable of handling large datasets. Besides, IFSM needs to know the corresponding feature types before learning and can only handle static datasets.
- OHSFS *vs*. ε-approximate: There is no significant difference between OHSFS and ε-approximate on predictive accuracy. The predictive accuracy of ε-approximate is slightly better than that of OHSFS in cases of SVM and CART but worse in the case of KNN. ε-approximate is a supervised mixed feature selection algorithm based on fuzzy rough sets. ε-approximate can define corresponding fuzzy relationships for different features, which requires knowing the feature types before learning. Meanwhile, the time complexity of the ε-approximate is very high and unsuitable for processing high-dimensional datasets.
- OHSFS *vs*. EUIAR: OHSFS performs better than EUIAR on predictive accuracy in cases of these three classifiers. Meanwhile, OHSFS is faster than EUIAR in running time. EUIAR is an unsupervised mixed feature selection algorithm based on fuzzy rough sets and selects the fewest features that may lead to the loss of some critical information. Besides, EUIAR requires two thresholds to be given before feature selection to control the radius and the number of selected features. On the contrary, it is challenging to specify parameter values for streaming feature selection before learning.
- OHSFS *vs*. FRUAR: FRUAR performs the worst on predictive accuracy among all these competing algorithms. Meanwhile, there is a significant difference between OHSFS and FRUAR in the case of KNN. FRUAR uses fuzzy rough sets to define the importance of individual features. The time complexity and space complexity of fuzzy rough sets based algorithms are very high. Therefore, the running time of FRUAR is much higher than other comparison algorithms.

In sum, OHSFS is competing on the predictive accuracy compared to the four traditional feature selection methods on six mixed datasets. Three comments must be explained: 1) Traditional feature selection methods can often require the datasets to select the optimal features according to different measurements and strategies. However, OHSFS is an online streaming feature selection method that can require the datasets only once and decide whether to retain or discard

the streaming features on the fly. 2) Traditional feature selection methods spend much more time on running time than OHSFS. Due to exceptionally long running time, these competing traditional mixed feature selection methods cannot handle high-dimensional datasets. 3) Traditional feature selection methods need information on each feature in the datasets before learning. Nevertheless, for streaming features in real-world applications, the information on feature type may not be required. Therefore, compared to traditional feature selection methods, OHSFS is better in line with practical application needs.

4.4 OHSFS vs. Online Streaming Feature Selection Methods

In this section, we compare OHSFS with five state-of-the-art online streaming feature selection algorithms including α-investing [18], Fast-OSFS [9], SAOLA [23], OSFSMI [47], and SFS-FI [14] [2]. We conduct the experiments on sixteen high-dimensional datasets as shown in Table 2. Since most of these datasets are numerical features, we randomly selected 50% of the features and discretized these features into ten equal parts. Thus, all experimental datasets are mixed feature types for our new method. Meanwhile, because these five competing algorithms cannot handle mixed features, we use their categorical version algorithms in experimental, and the datasets are equidistantly discretized into two intervals. All algorithms are implemented in MATLAB.

Fig. 4 shows graphically the predictive accuracy of these competing algorithms on five different classifiers. Table 10-16 summarize the predictive accuracy, the running time and the mean number of selected features of these competing algorithms. The p-values of Friedman test on KNN, SVM, CART, Ensemble Learning, Neural Network, running time, and the mean number of selected features are 0.3502e-05, 0.5997e-05, 0.0101, 9.6416e-08, 0.0074, 0.2735e-14, and 0.485e-07 respectively. Thus, there is a significant difference between these competing algorithms on predictive accuracy, running time and number of selected features. According to the Nemenyi test, the value of CD is 1.8848. Fig. 5 shows the statistical test of these competing algorithms in cases of KNN, SVM, CART, Ensemble Learning, and Neural Network.

TABLE 10: Predictive Accuracy Using KNN as the Classifier

Dataset	α-investing	Fast-OSFS	SAOLA	OSFSMI	SFS-FI	OHSFS
Arrarrhythmia	0.5527	0.1958	0.608	0.5696	0.6133	0.6309
LYMPHOMA	0.74	0.7483	0.9767	0.9017	0.9867	0.98
SRBCT	0.7762	0.6183	0.9631	0.8176	0.9011	0.9641
DLBCL	0.7853	0.756	0.9267	0.7173	0.92	0.9187
CAR	0.6117	0.2764	0.9068	0.6041	0.8225	0.8741
OVARIAN	0.9723	0.9653	0.9774	0.9519	0.9253	0.9822
LEU	0.8057	0.8714	0.9643	0.9543	0.9514	0.97
PROSTATE	0.801	0.837	0.891	0.875	0.87	0.9
ARCENE	0.709	0.671	0.6415	0.6805	0.656	0.7785
LUNG2	0.8678	0.7881	0.9541	0.8376	0.9565	0.9564
LUNG	0.9672	0.9383	0.9933	0.9806	0.965	0.99
SYLVA	0.9858	0.9881	0.9821	0.9741	0.9355	0.9879
GISETTE	0.9542	0.8892	0.9006	0.6883	0.9341	0.9109
DEXTER	0.8218	0.6338	0.818	0.5002	0.6352	0.8068
HIVA	0.9656	0.9655	0.9658	0.9639	0.9647	0.9657
NOVA	0.6685	0.6976	0.6025	0.7151	0.5834	0.7561
AVG.	0.8115	0.74	0.8794	0.7957	0.8513	0.8983
AVG. RANKS	3.8125	4.75	2.625	4.375	3.6875	1.75

From Figs. 4-5 and Tables 11-16, we can indicate that:

[2] Public available at https://github.com/kuiy/LOFS, and https://github.com/doodzhou/OSFS.

TABLE 11: Predictive Accuracy Using SVM as the Classifier

Dataset	α-investing	Fast-OSFS	SAOLA	OSFSMI	SFS-FI	OHSFS
Arrarrhythmia	0.6022	0.5416	0.6613	0.6153	0.6422	0.6696
LYMPHOMA	0.7583	0.7333	0.9817	0.8867	0.9883	0.98
SRBCT	0.8029	0.6091	0.9659	0.8414	0.9625	0.9515
DLBCL	0.7987	0.7733	0.932	0.8453	0.9133	0.9547
CAR	0.6233	0.3144	0.9115	0.6335	0.8822	0.8713
OVARIAN	0.977	0.9633	0.9806	0.9727	0.9652	0.9822
LEU	0.8414	0.8543	0.9614	0.9571	0.94	0.9657
PROSTATE	0.819	0.854	0.866	0.896	0.867	0.893
ARCENE	0.7015	0.657	0.645	0.676	0.6395	0.7215
LUNG2	0.8776	0.8023	0.9448	0.8498	0.9472	0.9433
LUNG	0.9767	0.9472	0.9933	0.9806	0.9806	0.9917
SYLVA	0.9914	0.9898	0.9835	0.9845	0.9403	0.9903
GISETTE	0.9602	0.862	0.9002	0.8988	0.8925	0.9035
DEXTER	0.8602	0.6407	0.8542	0.6108	0.643	0.8697
HIVA	0.9639	0.965	0.9646	0.9282	0.9648	0.9649
NOVA	0.7656	0.7657	0.7579	0.706	0.7211	0.7754
AVG.	0.8325	0.7671	0.894	0.8302	0.8681	0.9018
AVG. RANKS	3.9375	4.9375	2.6875	4.0938	3.5313	1.8125

TABLE 12: Predictive Accuracy Using CART as the Classifier

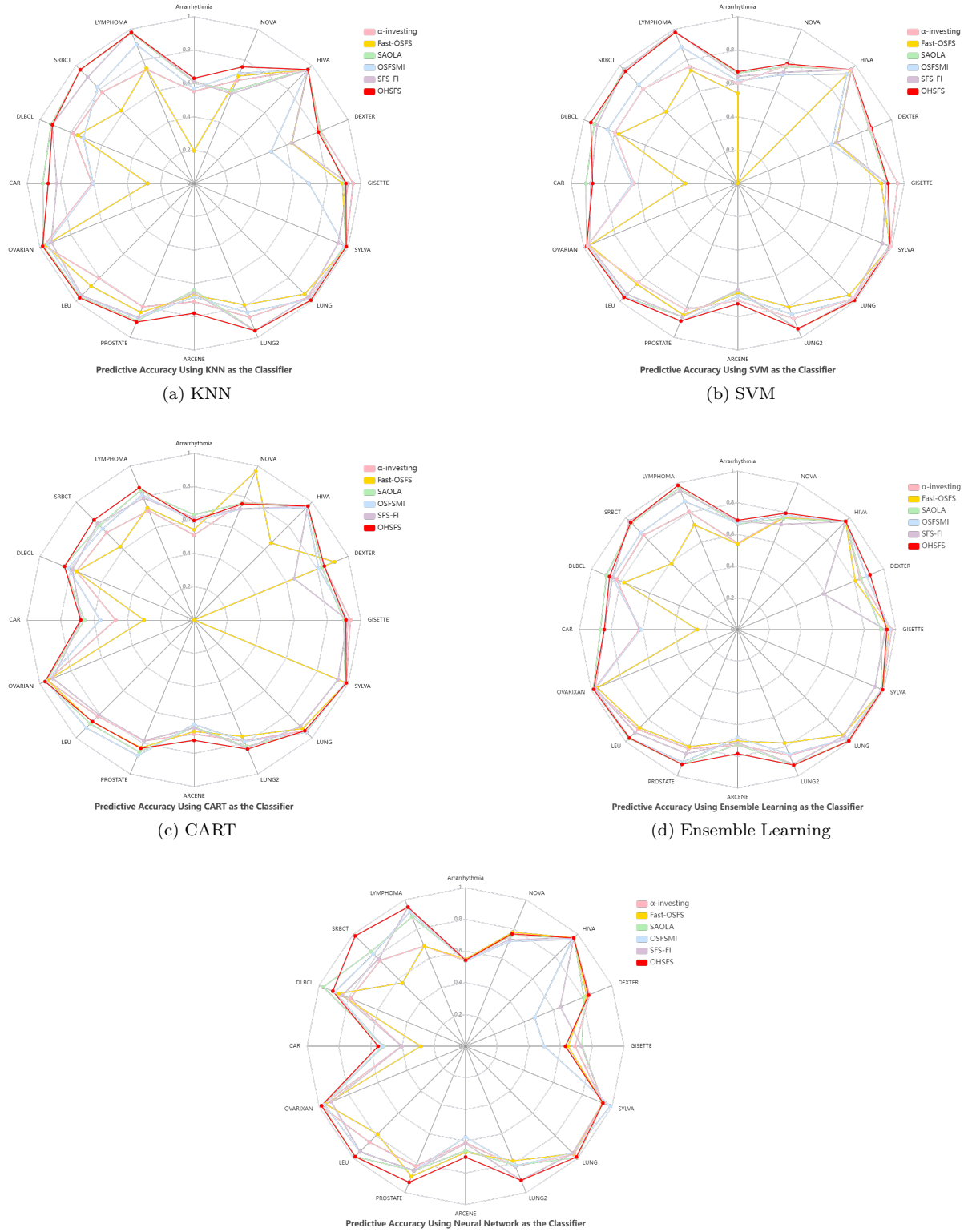
Dataset	α-investing	Fast-OSFS	SAOLA	OSFSMI	SFS-FI	OHSFS
Arrarrhythmia	0.5091	0.5416	0.6307	0.5924	0.6129	0.5962
LYMPHOMA	0.71	0.7283	0.84	0.8117	0.7917	0.8583
SRBCT	0.7408	0.6238	0.8021	0.7744	0.8159	0.8481
DLBCL	0.764	0.7613	0.844	0.816	0.7893	0.84
CAR	0.4704	0.3	0.657	0.5626	0.6753	0.6792
OVARIAN	0.9498	0.949	0.9619	0.958	0.9163	0.9663
LEU	0.8157	0.8629	0.88	0.9143	0.8043	0.8614
PROSTATE	0.784	0.829	0.857	0.88	0.783	0.833
ARCENE	0.6835	0.6685	0.642	0.6255	0.6465	0.7215
LUNG2	0.7853	0.7551	0.8214	0.7883	0.8295	0.8381
LUNG	0.94	0.9206	0.9483	0.9517	0.9033	0.9394
SYLVA	0.9877	0.9873	0.9818	0.9822	0.9351	0.9877
GISETTE	0.9376	0.9124	0.8984	0.9209	0.9201	0.9106
DEXTER	0.8435	0.653	0.829	0.8088	0.6497	0.8457
HIVA	0.9656	0.9655	0.9665	0.9531	0.9647	0.9654
NOVA	0.7628	0.7637	0.7576	0.7188	0.7207	0.7532
AVG.	0.7906	0.7639	0.8324	0.8162	0.7974	0.8403
AVG. RANKS	3.7188	4.375	2.75	3.5	4.3125	2.3437

TABLE 13: Predictive Accuracy Using Ensemble Learning as the Classifier

Dataset	α-investing	Fast-OSFS	SAOLA	OSFSMI	SFS-FI	OHSFS
Arrarrhythmia	0.5449	0.5407	0.6662	0.6727	0.6836	0.6898
LYMPHOMA	0.805	0.715	0.9667	0.875	0.9483	0.985
SRBCT	0.8408	0.5898	0.9473	0.8635	0.9666	0.9558
DLBCL	0.8293	0.7773	0.8973	0.8573	0.856	0.876
CAR	0.6197	0.2547	0.8674	0.6114	0.8427	0.8427
OVARIAN	0.9806	0.9596	0.9869	0.9747	0.9727	0.9853
LEU	0.89	0.8757	0.9714	0.9657	0.9171	0.9671
PROSTATE	0.813	0.8	0.919	0.906	0.847	0.92
ARCENE	0.73	0.7035	0.7245	.6815	0.713	0.7835
LUNG2	0.8577	0.7743	0.9236	0.8518	0.9186	0.926
LUNG	0.9756	0.9411	0.9922	0.9728	0.9489	0.9939
SYLVA	0.9916	0.9905	0.9849	0.985	0.9404	0.9909
GISETTE	0.9733	0.9589	0.9059	0.9637	0.934	0.9434
DEXTER	0.8375	0.8048	0.8452	0.8817	0.5885	0.9062
HIVA	0.9667	0.9658	0.9668	0.9676	0.9647	0.9663
NOVA	0.77	0.7632	0.7592	0.7798	0.718	0.795
AVG.	0.8391	0.7759	0.8953	0.8631	0.8600	0.9079
AVG. RANKS	3.63	5.31	2.69	3.44	4.16	1.78

- OHSFS *vs.* α-investing: α-investing is an adaptive complexity penalty method that can dynamically adjust the threshold on the error reduction required for adding a new feature. Based on the penalized likelihood ratio, α-investing has an advantage in not requiring the multiple retraining of the model. According to the statistical test results, OHSFS performs significantly better than α-investing in predictive accuracy in cases of KNN and SVM. On the classifiers of CART, Ensemble Learning, and Neural Network, OHSFS gets much lower average

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3350630



(a) KNN

(b) SVM

(c) CART

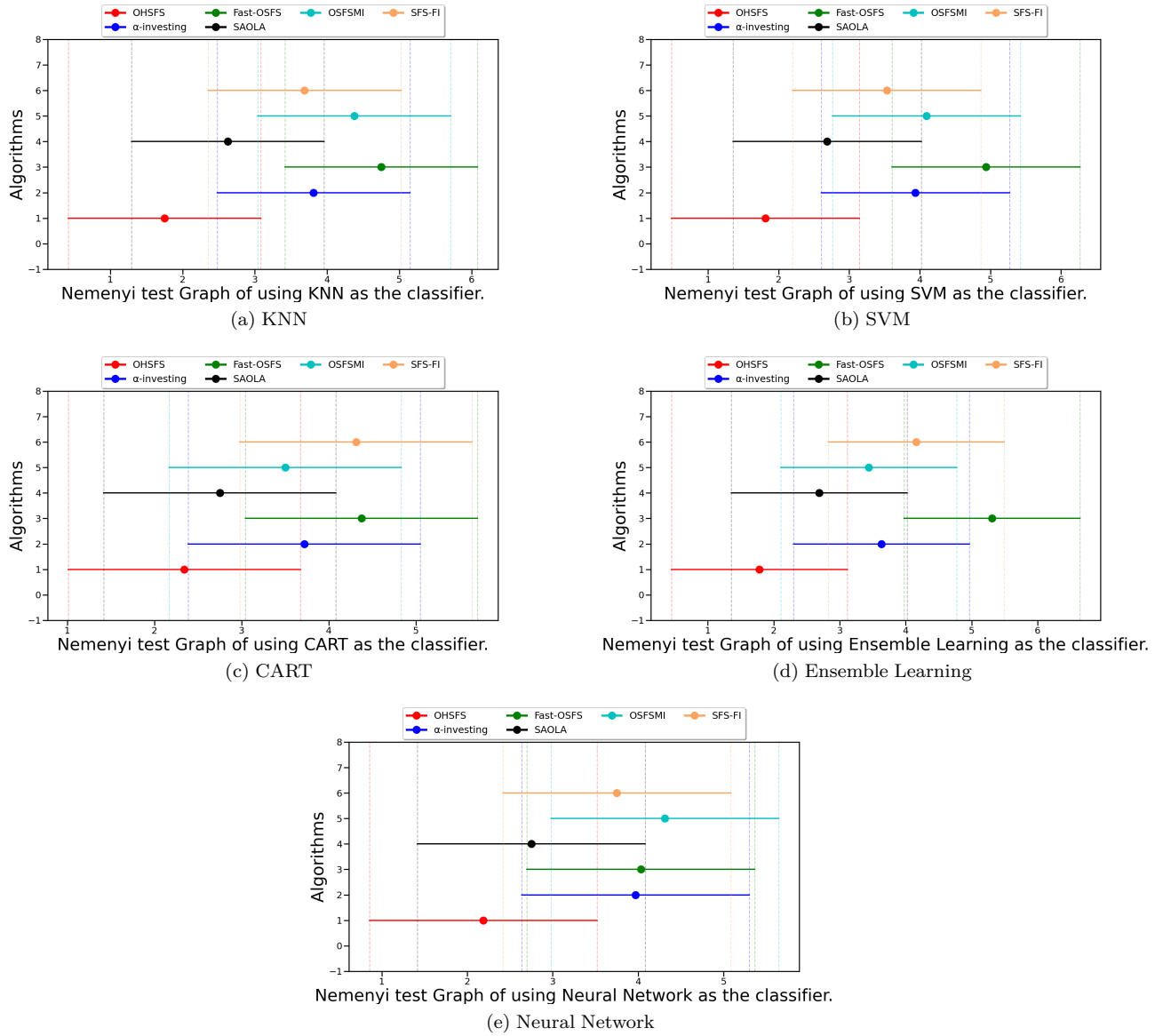
(d) Ensemble Learning

(e) Neural Network

Fig. 4: Predictive accuracy of these competing algorithms with five classifiers

ranks on predictive accuracy. Besides, OHSFS gets much higher average predictive accuracy than α-investing with all these five classifiers. α-investing does not handle redundancy between features and selects fewer features on sparse datasets. Therefore, OHSFS performs better than α-investing in predictive accuracy. On running time, α-investing is the shortest among these competing algorithms. α-investing dynamically decides which features to generate and add to the feature stream, which provides potentially significant savings in computation. In

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3350630



(a) KNN (b) SVM (c) CART (d) Ensemble Learning (e) Neural Network

Fig. 5: The statistical test graph of OHSFS vs. online streaming feature selection algorithms

TABLE 14: Predictive Accuracy Using Neural Network as the Classifier

Dataset	α-investing	Fast-OSFS	SAOLA	OSFSMI	SFS-FI	OHSFS
Arrarrhythmia	0.54	0.5489	0.5444	0.5378	0.5444	0.5422
LYMPHOMA	0.6833	0.6833	0.8833	0.9167	0.95	0.95
SRBCT	0.7636	0.5636	0.8434	0.8224	0.7762	0.9846
DLBCL	0.7867	0.8667	0.9733	0.8933	0.84	0.9067
CAR	0.4025	0.2822	0.5173	0.523	0.4029	0.5516
OVARIAN	0.9881	0.9569	0.9802	0.9645	0.9176	0.9842
LEU	0.8571	0.7857	0.9857	0.9429	0.9429	0.9857
PROSTATE	0.82	0.89	0.85	0.85	0.85	0.93
ARCENE	0.61	0.67	0.66	0.575	0.615	0.7
LUNG2	0.8219	0.7826	0.8131	0.8175	0.9156	0.9166
LUNG	0.9667	0.9611	0.9944	0.9556	0.9556	0.9889
SYLVA	0.9385	0.9357	0.9385	0.989	0.9387	0.9385
GISETTE	0.6907	0.6464	0.7353	0.4961	0.7314	0.6304
DEXTER	0.8467	0.8283	0.8083	0.4717	0.6467	0.84
HIVA	0.9631	0.9674	0.9676	0.9544	0.9648	0.9659
NOVA	0.7787	0.7777	0.7564	0.7148	0.7268	0.7668
AVG.	0.7786	0.7592	0.8282	0.7765	0.7949	0.8489
AVG. RANKS	3.97	4.03	2.75	4.31	3.75	2.19

TABLE 15: Running time(seconds)

Dataset	α-investing	Fast-OSFS	SAOLA	OSFSMI	SFS-FI	OHSFS
Arrarrhythmia	0.0077	0.2624	0.0246	0.2613	0.1044	8.3131
LYMPHOMA	0.0651	2.3755	0.897	0.461	3.294	40.1677
SRBCT	0.0286	1.1187	0.2155	0.2059	4.3432	0.8469
DLBCL	0.1387	2.9935	0.2323	1.0585	1.2055	13.4415
CAR	0.5461	7.1761	6.3329	1.2339	241.2094	46.9425
OVARIAN	1.3392	14.6532	0.9837	12.7006	13.8304	44.3455
LEU	0.195	3.4568	0.2511	0.7164	21.8869	16.0849
PROSTATE	0.1318	2.9869	0.172	0.6658	0.6187	6.9797
ARCENE	0.4637	5.4661	0.2295	124.8731	1.2078	54.5846
LUNG2	0.1436	3.2084	2.2243	0.4246	7.8362	132.6171
LUNG	1.0266	8.4292	1.7554	1.7922	3.4119	43.809
SYLVA	0.2746	132.0478	0.0735	3.4589	0.1433	114.0287
GISETTE	58.9935	386.4437	1.1407	778.7647	16.4794	849.6236
DEXTER	2.4291	8.7344	0.3374	2092.413	1.3195	20.5311
HIVA	0.3153	3.9962	0.1456	63.2055	0.2396	14.6994
NOVA	2.5189	20.0557	0.5917	3550.695	1.1994	72.2485
AVG.	4.2886	37.7128	0.9755	414.5581	18.6459	92.4521
AVG. RANKS	1.75	4.5625	1.8125	3.8125	3.5625	5.5

contrast, OHSFS uses MIC to calculate the information between two arbitrary types of stream features, which is

TABLE 16: The mean number of selected features

Dataset	α-investing	Fast-OSFS	SAOLA	OSFSMI	SFS-FI	OHSFS
Arrarrhythmia	5.56	3	21.32	100.14	80.22	21.78
LYMPHOMA	6.04	2	166.52	17.08	240.64	319.46
SRBCT	6.62	2	56.4	9.98	664.98	20.8
DLBCL	11.24	2.06	60.7	25.12	51.52	142.54
CAR	24.16	2	308.06	9.4	6042.7	109.8
OVARIAN	32.92	2.96	32.82	73.48	207.68	45.52
LEU	16	2	43.82	7.18	77.72	164.6
PROSTATE	10	2.14	22.74	8.3	21.42	47.96
ARCENE	10.08	3.02	27.08	2232.44	22.64	35.88
LUNG2	20.12	3	322.42	12.1	432.22	170.1
LUNG	34.38	3.2	283.38	9.22	52.78	96.46
SYLVA	37.48	14.44	9.64	95.72	2.64	16.9
GISETTE	297.98	10.14	20.58	1882.28	48.94	70.76
DEXTER	12.74	2.1	32.2	15024.46	22.24	87.7
HIVA	25.76	4.94	5.6	809.5	1.66	8.6
NOVA	15.46	6.3	11.04	11005.92	2.38	163.4
AVG.	35.4212	4.0813	89.02	1957.645	520.7738	95.1413
AVG. RANKS	3.0625	1.25	3.875	4.125	3.9375	4.75

very time-consuming.

- OHSFS *vs*. Fast-OSFS: Fast-OSFS aims to select strongly relevant and non-redundant features from streaming features. Two key components (online relevance analysis and online redundancy analysis) ensure the effectiveness of the OSFS framework. Fast-OSFS is an enhanced method based on OSFS that can significantly improve selection efficiency during redundancy analysis. There is a significant difference between OHSFS and Fast-OSFS on predictive accuracy in cases of KNN, SVM, CART, and Ensemble Learning. Meanwhile, OHSFS gets more than 10% higher average predictive accuracy than Fast-OSFS with all these five classifiers. Fast-OSFS selects the fewest features among all these competing algorithms that may lead to the loss of important information and lower prediction accuracy. Therefore, OHSFS performs much better than Fast-OSFS in predictive accuracy. On running time, Fast-OSFS is faster than OHSFS due to the time savings during redundancy analysis.
- OHSFS *vs*. SAOLA: SAOLA is designed for extremely high-dimensional data. By employing novel pairwise comparison techniques and maintaining a parsimonious model over time online, SAOLA is scalable on datasets and can run very fast. SAOLA has two versions (based on Fisher's Z-test and Mutual Information respectively) for continuous and discrete streaming features. On predictive accuracy, SAOLA performs better than our algorithm OHSFS in cases of KNN and SVM according to the average ranks of each classifier. OHSFS performs better than SAOLA in CART, Ensemble Learning, and Neural Network cases. Meanwhile, OHSFS achieves higher average predictive accuracy than SAOLA in all these five classifiers. SAOLA and OHSFS select almost the same number of features. Thus, these experimental results indicate the superiority of the selected features by OHSFS in predictive accuracy. On running time, SAOLA is much faster than OHSFS. To deal with the heterogeneous new streaming features, we use MIC to calculate the information between different streaming features and do not need to know the feature type information. However, MIC is very time-consuming to get the maximum value by the exhaustive search.
- OHSFS *vs*. OSFSMI: OSFSMI employs the mutual information concept in a streaming manner to evaluate the correlation between features. OSFSMI aims to select informative features by removing redundant and irrelevant features and does not employ any adjustable user-defined parameters. OHSFS performs significantly better than OSFSMI in cases of KNN and SVM. On CART, Ensemble Learning, and Neural Network, OHSFS gets higher predictive accuracy on average and lower average ranks than OSFSMI. On running time, OSFSMI is speedy on some datasets. Meanwhile, OSFSMI spends the most time on other datasets, such as ARCENE, DEXTER, and NOVA. OSFSMI selects the most features on average among these competing algorithms. Thus, the performance of OSFSMI varies widely on different datasets, which indicates its poor adaptability.
- OHSFS *vs*. SFS-FI: SFS-FI aims to account for feature interaction during streaming feature selection. Based on the metric of interaction gain, SFS-FI can select relevant and interactive streaming features on the fly. SFS-FI applies Fisher's Z-test and Mutual Information for continuous and discrete streaming features to compute the relationship between features, respectively. OHSFS performs significantly better than SFS-FI on KNN and CART. In the case of SVM, Ensemble Learning, and Neural Network, OHSFS achieves higher average prediction accuracy and lower average ranks than SFS-FI. Since the interaction between features is considered, SFS-FI selects, on average, many more features than OHSFS. Therefore, the features selected by OHSFS are superior to those chosen by SFS-FI. In terms of running time, SFS-FI is faster than OHSFS. SFS-FI also uses mutual information to select features but cannot handle mixed features and features of unknown type.

In sum, OHSFS achieves the highest predictive accuracy and lowest average ranks among these competing algorithms on these datasets. The superiority of OHSFS lines in two aspects: 1) In terms of MIC, OHSFS can treat mixed features in a uniformed framework that decreases the information loss during online feature selection; 2) The two metrics MIC_{Gain} and MIC_{Cor} ensure the effectiveness and efficiency of OHSFS. Meanwhile, most existing streaming feature selection methods cannot handle heterogeneous streaming features directly. In real-world applications, the streaming features can arrive in a random order. Therefore, it is unrealistic to know the feature type for the next arriving streaming feature before learning. Since OHSFS is nonparametric and does not need to know the feature type of each streaming feature in advance, it is better in line with practical application needs.

5 CONCLUSION

This paper proposes a novel online adaptive feature selection method to address heterogeneous streaming features without the feature type information in advane, which is more in line with practical applications. First, we model the issue of online heterogeneous streaming feature selection as a minimax problem. Then, in terms of MIC which can measure the correlation between arbitrary types of streaming features, we derive two new metrics that aim to select informative and compact features. Meanwhile, we proof the effectiveness of efficiency of these two metrics. Finally, extensive experiments demonstrate the effectiveness of our new proposed method compared to

four traditional mixed feature selection algorithms and five online streaming feature selection methods. However, the time complexity of OHSFS is high due to the calculation of MIC. In future work, we will focus on an online distributed streaming feature selection method that can process multiple streaming features concurrently in a distributed manner.

References

[1] H. Liu and H. Motoda, Computational methods of feature selection. CRC press, 2007.

[2] Z. Ling, B. Li, Y. Zhang, Q. Wang, K. Yu, and X. Wu, "Causal feature selection with efficient spouses discovery," IEEE Transactions on Big Data, vol. 9, no. 2, pp. 555–568, 2023.

[3] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," ACM Computing Surveys, vol. 50, no. 6, pp. 1–45, 2018.

[4] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," Artificial Intelligence Review, vol. 53, no. 2, pp. 907–948, 2020.

[5] E. Hancer, B. Xue, and M. Zhang, "A survey on feature selection approaches for clustering," Artificial Intelligence Review, vol. 53, no. 6, pp. 4519–4545, 2020.

[6] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," Neurocomputing, vol. 300, pp. 70–79, 2018.

[7] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for web image search," IEEE Transactions on Image Processing, vol. 21, no. 11, pp. 4649–4661, 2012.

[8] W. Ding, T. F. Stepinski, Y. Mu, L. Bandeira, R. Ricardo, Y. Wu, Z. Lu, T. Cao, and X. Wu, "Subkilometer crater discovery with boosting and transfer learning," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 4, pp. 1–22, 2011.

[9] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 5, pp. 1178–1192, 2013.

[10] J. Wang, P. Zhao, S. C. Hoi, and R. Jing, "Online feature selection and its applications," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 3, pp. 698–710, 2013.

[11] M. H. u. Rehman, E. Ahmed, I. Yaqoob, I. A. T. Hashem, M. Imran, and S. Ahmad, "Big data analytics in industrial iot using a concentric computing model," IEEE Communications Magazine, vol. 56, no. 2, pp. 37–43, 2018.

[12] X. Hu, P. Zhou, P. Li, J. Wang, and X. Wu, "A survey on online feature selection with streaming features," Frontiers of Computer Science, vol. 12, no. 3, pp. 479–493, 2018.

[13] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of machine learning research, vol. 3, no. Mar, pp. 1157–1182, 2003.

[14] P. Zhou, P. Li, S. Zhao, and X. Wu, "Feature interaction for streaming feature selection," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 10, pp. 4691–4702, 2021.

[15] N. AlNuaimi, M. M. Masud, M. A. Serhani, and N. Zaki, "Streaming feature selection algorithms for big data: A survey," Applied Computing and Informatics, vol. 18, no. 1/2, pp. 113–135, 2020.

[16] B. Sang, H. Chen, T. Li, W. Xu, and H. Yu, "Incremental approaches for heterogeneous feature selection in dynamic ordered data," Information Sciences, vol. 541, pp. 475–501, 2020.

[17] P. Zhang, T. Li, Z. Yuan, C. Luo, K. Liu, and X. Yang, "Heterogeneous feature selection based on neighborhood combination entropy," IEEE Transactions on Neural Networks and Learning Systems, pp. 1–14, 2022.

[18] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar, "Streamwise feature selection," Journal of Machine Learning Research, vol. 3, no. 2, pp. 1532–4435, 2006.

[19] H. G. Li, X. D. Wu, Z. Li, and W. Ding, "Group feature selection with streaming features," in IEEE 13th International Conference on Data Mining, 2013, pp. 1109–1114.

[20] S. Eskandari and M. Javidi, "Online streaming feature selection using rough sets," International Journal of Approximate Reasoning, vol. 69, no. C, pp. 35–57, 2016.

[21] P. Zhou, X. Hu, P. Li, and X. Wu, "Online feature selection for high-dimensional class-imbalanced data," Knowledge-Based Systems, vol. 136, pp. 187–199, 2017.

[22] P. Zhou, X. Hu, P. Li, and X. Wu, "Online streaming feature selection using adapted neighborhood rough set," Information Sciences, vol. 481, pp. 258–279, 2019.

[23] K. Yu, X. Wu, W. Ding, and J. Pei, "Scalable and accurate online feature selection for big data," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 11, no. 2, pp. 1–39, 2016.

[24] P. Zhou, S. Zhao, Y. Yan, and X. Wu, "Online scalable streaming feature selection via dynamic decision," ACM Trans. Knowl. Discov. Data, vol. 16, no. 5, pp. 1–20, 2022.

[25] Z. Yuan, H. Chen, T. Li, Z. Yu, B. Sang, and C. Luo, "Unsupervised attribute reduction for mixed data based on fuzzy rough sets," Information Sciences, vol. 572, pp. 67–87, 2021.

[26] Z. Yuan, H. Chen, and T. Li, "Exploring interactive attribute reduction via fuzzy complementary entropy for unlabeled mixed data," Pattern Recognition, vol. 127, p. 108651, 2022.

[27] X. Zhang, C. Mei, D. Chen, and J. Li, "Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy," Pattern Recognition, vol. 56, pp. 1–15, 2016.

[28] W. Shu, W. Qian, and Y. Xie, "Incremental feature selection for dynamic hybrid data using neighborhood rough set," Knowledge-Based Systems, vol. 194, p. 105516, 2020.

[29] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," science, vol. 334, no. 6062, pp. 1518–1524, 2011.

[30] P. Zhou, Y. Zhang, Y. Yan, and S. Zhao, "Unknown type streaming feature selection via maximal information

coefficient," in 2022 IEEE International Conference on Data Mining Workshops (ICDMW), 2022, pp. 650–657.
[31] L. Zhao, Z. Chen, Y. Hu, G. Min, and Z. Jiang, "Distributed feature selection for efficient economic big data analysis," IEEE Transactions on Big Data, vol. 4, no. 2, pp. 164–176, 2016.
[32] X. Wang, L. T. Yang, H. Liu, and M. J. Deen, "A big data-as-a-service framework: State-of-the-art and perspectives," IEEE Transactions on Big Data, vol. 4, no. 3, pp. 325–340, 2017.
[33] J. Paul, P. Dupont et al., "Kernel methods for heterogeneous feature selection," Neurocomputing, vol. 169, pp. 187–195, 2015.
[34] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," Neural computing and applications, vol. 24, no. 1, pp. 175–186, 2014.
[35] D. Lianjie, C. Degang, W. Ningling, and L. Zhanhui, "Key energy-consumption feature selection of thermal power systems based on robust attribute reduction with rough sets," Information Sciences, vol. 532, pp. 61–71, 2020.
[36] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in Advances in neural information processing systems, 2005, pp. 517–514.
[37] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," in Conference on Uai, 2011, pp. 1–8.
[38] K.-J. Kim and C.-H. Jun, "Rough set model based feature selection for mixed-type data with feature space decomposition," Expert Systems with Applications, vol. 103, pp. 196–205, 2018.
[39] F. Wang and J. Liang, "An efficient feature selection algorithm for hybrid data," Neurocomputing, vol. 193, pp. 33–41, 2016.
[40] S. Solorio-Fernández, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, "A new unsupervised spectral feature selection method for mixed data: a filter approach," Pattern Recognition, vol. 72, pp. 314–326, 2017.
[41] M. Labani, P. Moradi, and M. Jalili, "A multi-objective genetic algorithm for text feature selection using the relative discriminative criterion," Expert Systems with Applications, vol. 149, p. 113276, 2020.
[42] H. Hamedmoghadam, M. Jalili, and X. Yu, "An opinion formation based binary optimization approach for feature selection," Physica A: Statistical Mechanics and its Applications, vol. 491, pp. 142–152, 2018.
[43] R. Sharkawy, K. Ibrahim, M. Salama, and R. Bartnikas, "Particle swarm optimization feature selection for the classification of conducting particles in transformer oil," IEEE Transactions on Dielectrics and Electrical Insulation, vol. 18, no. 6, pp. 1897–1907, 2011.
[44] X. Song, Y. Zhang, Y. Guo, X. Sun, and Y. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," IEEE Transactions on Evolutionary Computation, vol. 24, no. 5, pp. 882–895, 2020.
[45] X. Song, Y. Zhang, D. Gong, H. Liu, and W. Zhang, "Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data," IEEE Transactions on Evolutionary Computation, vol. 27, no. 93, pp. 595–609, 2023.
[46] J. Liu, Y. Lin, W. Ding, H. Zhang, C. Wang, and J. Du, "Multi-label feature selection based on label distribution and neighborhood rough set," Neurocomputing, vol. 524, pp. 142–157, 2023.
[47] M. Rahmaninia and P. Moradi, "Osfsmi: online stream feature selection method based on mutual information," Applied Soft Computing, vol. 68, pp. 733–746, 2018.
[48] W. Gao, P. Hao, Y. Wu, and P. Zhang, "A unified low-order information-theoretic feature selection framework for multi-label learning," Pattern Recognition, vol. 134, p. 109111, 2023.
[49] A. Rafie, P. Moradi, and A. Ghaderzadeh, "A multi-objective online streaming multi-label feature selection using mutual information," Expert Systems with Applications, vol. 216, p. 119428, 2023.
[50] T. Naghibi, S. Hoffmann, and B. Pfister, "A semidefinite programming based search strategy for feature selection with mutual information measure," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 8, pp. 1529–1541, 2014.
[51] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in Noise reduction in speech processing. Springer, 2009, pp. 1–4.
[52] A. Kumar and S. Abirami, "Aspect-based opinion ranking framework for product reviews using a spearman's rank correlation coefficient method," Information Sciences, vol. 460, pp. 23–41, 2018.
[53] Y. W. Lee, "Statistical theory of communication," American Journal of Physics, vol. 29, no. 4, pp. 276–278, 1961.
[54] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1226–1238, 2005.
[55] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," The Journal of Machine Learning Research, vol. 7, pp. 1–30, 2006.

Peng Zhou received the Ph.D. degree from the Hefei University of Technology, Hefei, China, in 2018. He is currently an associate professor with Anhui University, Hefei. His research interests include data mining, machine learning, stream learning, and feature selection.

Yunyun Zhang received the M.Sc. degree from the School of Computer Science and Technology, Anhui University, Hefei. Her current research interests include feature selection and stream learning.

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3350630

Zhaolong Ling received the the PhD degree in the School of of Computer and Information at the Hefei University of Technology, China, in 2020. He is a Lecturer with the School of Computer Science and Technology, Anhui University, China. His research interests include feature selection, casual discovery, and data mining.

Yuanting Yan received his PhD degree in computer science from Anhui University, China, in 2016. He is currently an associate professor in the School of Computer Science and Technology at Anhui University. His current research interests include machine learning, granular computing and bioinformatics.

Shu Zhao received her Ph.D. degree in Computer Science from Anhui University in 2007. She is currently a professor in 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 PhD degree in artificial intelligence from the University of Edinburgh, Edinburgh, Scotland, U.K, 1993, and he is a Chang Jiang Scholar in the School of Computer Science and Information Engineering at the Hefei University of Technology, China. His current research interests include data mining, big data analytics, and knowledge-based systems.

Prof. Wu is a Steering Committee Chair of the IEEE International Conference on Data Mining, an Editor-in-Chief of the Knowledge and Information Systems (Springer), and a Series Editor-in-Chief of the Springer Book Series on Advanced Information and Knowledge Processing. He was an Editor-in-Chief of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEEING from 2005 to 2008. He served as a Program Committee Chair/Co-Chair of the 2003 IEEE International Conference on Data Mining (ICDM '03), the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-07), and the 19th ACM Conference on Information and Knowledge Management (CIKM2010). He is Fellow of IEEE and AAAS (American Association for the Advancement of Science).