

## Research Article

B. Venkatesh and J. Anuradha\*

# Fuzzy Rank Based Parallel Online Feature Selection Method using Multiple Sliding Windows

<https://doi.org/10.1515/comp-2020-0169>

Received Apr 30, 2020; accepted Jul 19, 2020

**Abstract:** Nowadays, in real-world applications, the dimensions of data are generated dynamically, and the traditional batch feature selection methods are not suitable for streaming data. So, online streaming feature selection methods gained more attention but the existing methods had demerits like low classification accuracy, fails to avoid redundant and irrelevant features, and a higher number of features selected. In this paper, we propose a parallel online feature selection method using multiple sliding-windows and fuzzy fast-mRMR feature selection analysis, which is used for selecting minimum redundant and maximum relevant features, and also overcomes the drawbacks of existing online streaming feature selection methods. To increase the performance speed of the proposed method parallel processing is used. To evaluate the performance of the proposed online feature selection method k-NN, SVM, and Decision Tree Classifiers are used and compared against the state-of-the-art online feature selection methods. Evaluation metrics like Accuracy, Precision, Recall, F1-Score are used on benchmark datasets for performance analysis. From the experimental analysis, it is proved that the proposed method has achieved more than 95% accuracy for most of the datasets and performs well over other existing online streaming feature selection methods and also, overcomes the drawbacks of the existing methods.

**Keywords:** Online Streaming Feature Selection (OSFS), Fuzzy Rank, Multiple Sliding-Window, Parallel Online Feature Selection

## 1 Introduction

In today's world the size of the datasets increase very rapidly, which leads to the presence of noisy, irrelevant,

and redundant features that degrade the performance of the learning algorithm. So, Feature Selection (FS) is applied to overcome these problems by discarding irrelevant and redundant features from the original feature set. Feature selection is the process of selecting a subset of features from the original feature set which are relevant to the target class. A variety of feature selection algorithms have been developed in the literature [4, 10, 19, 20], and it has been shown that FS improves the performance of the learning algorithm by increasing the prediction accuracy and thus reducing the burden for the learning algorithms. However, most of the existing feature selection algorithms are batch processing only, which means all the features of the datasets are available before the feature selection process is computed.

These traditional feature selection methods are not suitable when the features are arriving dynamically. Similarly in case of real-world applications [6, 25], all the features are not available before the feature selection process starts. Some of the examples are remote sensing images [18], email spam filtering [13, 29], texture-based image segmentation [23], network intrusion detection [32], and behavioral changes of hot topics in Weibo. In such cases, the features are generated dynamically, one after the other or as a group. It is also not feasible to wait until all the features have arrived before the learning process starts, and for the failure to implement traditional selection algorithms. So, online feature selection was introduced to cope with streaming features. In streaming data the features are arriving in two ways: 1) The number of features is fixed and the instances arrive dynamically, where patterns keep changing, and 2) different features may arrive dynamically at a given point in time. Some of the online feature selection methods available are Grafting [15], Alpha-Investing [34], Online Streaming Feature Selection (OSFS) [26], Online Group Feature Selection (OGFS) [24], Scalable and Accurate On Line Approach (SAOLA) [31], and Online Streaming Feature selection Using Sliding Window (OSFSW) [27]. There are some drawbacks to the above-mentioned online feature selection methods. Grafting [15] is good at discarding most of the redundant features and irrelevant features, but it requires the number of features in advance for calculating the regularization parameter ( $\lambda$ ). Alpha-Investing [34] is good at

\*Corresponding Author: J. Anuradha: SCOPE, Vellore Institute of Technology, Vellore, India; Email: januradha@vit.ac.in

B. Venkatesh: SCOPE, Vellore Institute of Technology, Vellore, India

handling large datasets, but it fails to remove the redundant features among the selected features. As a result, the error rate for this method is rather high. OSFSW uses a sliding window method for selecting the features from the online streaming data and it needs two stages for selecting the features. To overcome the drawbacks mentioned above, we proposed Online Streaming Multi Window Feature Selection method (OSMWFS) processes in parallel for selecting the features using the fast-mRMR feature selection method [17].

The proposed OSMWFS method uses the concept of multiple sliding windows being processed in parallel. In this method, instead of crisp values, we use fuzzy data for feature selection. The fuzzy features collected from the multiple sliding windows are passed to fuzzy fast-mRMR analysis for selecting the candidate features and rank the features. The fast-mRMR feature selection is used to reduce runtime by processing the redundancy of the candidate features only for the already selected relevant features, instead of the entire dataset. The process is continuous until some stopping criteria are met. OSMWFS method uses a fuzzy rank aggregation method for assigning the final ranks for the features present in the candidate feature sets. Then we select the top 'n' (Here 'n' is set to 3 in our proposed method) ranked features from the aggregate ranked features sets. Rank aggregation helps in discarding the redundant features present in other candidate feature sets. It also reduces the time needed to select the optimal features. To compare the performance of the proposed method, four state-of-the-art online streaming feature selection algorithms, namely Alpha-Investing, OSFS, SAOLA, and OSFSW are used over publicly available benchmark datasets. We use performance measures like classification Accuracy, Precision, Recall, F1-Score, time taken, and the number of features selected to validate the performance of the proposed OSMWFS method over the available state-of-the-art online streaming feature selection methods. The highlights of the proposed OSMWFS method are as follows:

- No chance of eliminating potential candidate points of time since we are using ranking aggregation,
- Parallel processing reduces the time required for FS, and
- Fast mRMR + fuzzy Ranking on multiple windows enhances performance.

The remainder of the paper is organized as follows. Section 2 explains the related work. Section 3 explains the proposed work, section 4 presents the dataset description and experimental setup, Section 5 contains result analysis, and finally, section 6 offers the conclusion and future work.

## 2 Related Work

For the past two decades, feature selection, has been used as an efficient method to handle large dimensionality and used for finding an optimal subset of features. In general, feature selection methods are divided into four methods, namely: filter, wrapper, embedded, and hybrid methods. These feature selection methods are suitable and perform well when all the candidate features are available before a learning process starts. However, they are not applicable for real-world applications where the features are arriving one after the other at a regular interval of time. In the following paragraphs, we are going to discuss the available online feature selection methods, their merits, and demerits from different researchers.

Perkins and Theiler 2003 proposed a streaming feature selection namely Grafting algorithm [15]. It uses a stage wise gradient descent approach. The proposed method can eliminate all the available redundant features. For determining the value of the tuning parameter ( $\lambda$ ), it requires all the candidate features in advance.

Zhou *et al.* 2005 proposed Alpha-Investing for streaming feature selection [34]. It overcomes the overfitting problem by dynamically adjusting the threshold for adding the feature to the model and also suitable for handling huge feature sets. The drawback of this method is that it evaluates the features only one time and discards to find the redundant features between the selected features. Therefore, the accuracy of this method is low.

Wu *et al.* 2010 proposed Online Streaming Feature Selection (OSFS) and Fast-OSFS for streaming feature selection based on feature relevance [26]. OSFS uses the Fisher Z-test and  $G^2$  test for removing the redundant and irrelevant features from the streaming features. OSFS attains high accuracy with lesser features than *Alpha-Investing* and *Grafting*. It is suitable for high redundant features but, its runtime increases with the increase of non-redundant features.

In [11] proposed GFSSF (Group Feature Selection with Streaming Features) based on group feature selection. It works at two levels, the group level, and the individual feature level. It uses the concept of mutual information and entropy for selecting the features. GFSSF simultaneously completes the feature selection both at the individual feature level and group level. When a new feature arrives first, it invokes a feature selection level for processing in the present group. The group-level feature selection is invoked when all the features of a group have arrived. GFSSF performs stream feature selection at the group level, individual level, or both.

In [23] proposed OGFS (Online Group Feature Selection) for streaming features based on the knowledge of group information. It consists of two stages inter-group and intra-group feature selection. In intra-group selection, features are selected based on spectral graph theory. In inter-group selection, the optimal feature subset is selected based on global group information by using the Lasso linear regression model. The disadvantage of OGFS is during intra-group selection; it requires a positive parameter number in advance and needs to select an optimal value.

In [22] uses sparsity regularization and truncation techniques for the online feature selection method. Jialei Wang *et al.* use two different types of inputs for learning; 1) learning with full input, and 2) learning with partial inputs. The proposed methods are not able to address the problem with online multiclass classification and regression problems.

In [7] proposed a new online streaming feature selection algorithm class OS-NRRSAR-SA using rough sets. Any feature selection require prior knowledge about the feature space but in case of rough set-based data mining any domain knowledge is required other than the available data. For controlling the unknown feature space in OSFS, the OS-NRRSAR-SA method uses classical significance analysis concept available in rough set theory and fails to discard redundant features.

In [31] proposed SAOLA (Scalable and Accurate On Line Approach) for online streaming feature selection. It uses a novel pairwise correlation among the features when features are arriving one by one. Yu *et al.* extend the SAOLA to group-SAOLA that deals with group features. It selects the feature groups that are sparse at the feature level and group level. For increasing the accuracy of the learning algorithm, it obtains a solution that is sparse at both intra and inter-groups simultaneously. It eliminates the redundant features from both the groups and are sparse at feature and group levels. The drawback of these methods is that it is complicated to obtain optimal value for the relevance threshold.

In [33] proposed online streaming feature selection using the sampling technique and the correlation between features (STCF). It improves the performance of OSFS for high-dimensional data by using sampling techniques. Sampling techniques are used for handling imbalanced data and increasing the accuracy of the model.

In [12] proposed two streaming feature selection namely multilabel feature selection with label correlation (MUCO) and multilabel streaming feature selection (MSFS), respectively based on fuzzy Mutual Information for Multilabel learning [12]. There are two steps in MSFS namely online redundancy analysis and online relevance analysis for finding optimal feature subsets. In addition to this,

the correlation between the label set are also used in this approach.

In [35] proposed online feature selection for high dimensional class imbalanced data. For real world applications like fraud detection and medical diagnosis, available data is mostly class imbalanced and has high dimensions. Existing online feature selection algorithms are discarding the small class data which has significant role in these applications. So to overcome these problem Zhou *et al.* proposed a new online feature selection algorithm called K-OFSD based on Dependency in K-nearest neighbors. For selecting relevant features, the algorithm uses information of nearest neighbors.

In [16] proposed filter based online feature selection method called Online Streaming Feature Selection based on Mutual Information (OSFSMI) and Online Stream Feature Selection Based on Mutual Information with fixed number of features (OSFSMI-k) using mutual information [16]. The proposed method uses MI in a streaming manner for finding correlation between features. Using this algorithm the author is able to find relevant features and can also discard redundant features.

In [27] proposed online streaming feature selection, namely OSFSW, for high redundant data using sliding window technique. In OSFSW, the features are added into the window that is highly relevant with the class label, then later, it uses this window to find the redundancy between the selected relevant features and already available candidate features. OSFSW uses a fixed-size sliding window size. In [28] proposed an extension to OSFSW, namely OSFASW that uses a self-adaption technique for selecting the size of the window.

In [36] proposed online streaming feature selection for high dimensional data using adapted neighborhood rough set. Adapted neighbours are detected using rough set neighborhood relation. Using this approach Zhou *et al.* is able to select maximal-relevant and maximal significant features.

In [3] proposed Dynamic Correlation-based Feature Selection (DCFS) for improving the performance of feature drifting in data streams. Proposed Correlation based Feature Selection (CFS) uses heuristic evaluation for scoring feature subsets. Relevant feature subsets are updated dynamically using CFS and also uses adaptive strategy for drift monitor.

In [7] proposed an extension to OS-NRRSAR-SA algorithm for eliminating redundant features [9]. The author uses online redundancy analysis for discarding redundant features and non-significant features are eliminated using online significance analysis.

### 3 Proposed Work

Consider a dataset  $F$  with  $n$  features  $F = \{f_1, f_2, \dots, f_n\}$ , the main objective of any feature selection method is to find an optimal feature subset say ( $F'$ ) with  $d$  feature (where  $d \ll n$ ). The objectives of the optimal feature subset ( $F'$ ) are; i)  $F'$  is an optimal solution, ii)  $f'_i \in F'$  are the most relevant features for target class ( $c$ ), and iii) The error rate for  $F'$  is very low than that of other feature subsets. So, to obtain an optimal feature set, the proposed OSMWFS method uses fuzzy fast-minimum Redundancy and Maximum Relevance feature selection method. It uses a greedy approach during the feature selection process for reducing more number of MI computations. Also to increase the computational speed of the feature selection process, we use parallel processing. To achieve parallel processing the proposed method uses CUDA programming model.

The primary goal of any feature selection is to select the features, which has high relevance with the target class  $c$ . So, to find the maximum relevant features the proposed fast-mRMR method uses Eqn. (1) to find the Max-Relevance features with the target class  $c$  [14].

$$\max R(F, c), \quad R = \frac{1}{|F|} \sum_{f_i \in F} I(f_i; c), \quad (1)$$

where  $I(f_i; c)$  is the mutual information between the feature and the target class.

The proposed method uses fuzzy features for feature selections. The fuzzy mutual information [8] between the fuzzy features are calculated by using the Eqn. 2.

$$FI(X, Y) = H(X) + H(Y) - H(X, Y) \quad (2)$$

where  $X$  and  $Y$  are two fuzzy variables,  $H(X)$ ,  $H(Y)$  are fuzzy entropy values for  $X$  and  $Y$ , respectively, and  $H(X, Y)$  fuzzy joint entropy for  $X$  and  $Y$ . The fuzzy entropy and fuzzy joint entropy on  $F$  can be calculated by using Eqns. (3), (4) and (5), respectively.

$$H(A) = -\frac{1}{n} \sum_{f_i \in F} \left[ \mu_A(f_i) \log \mu_A(f_i) + (1 - \mu_A(f_i)) \log(1 - \mu_A(f_i)) \right] \quad (3)$$

$$H(B) = -\frac{1}{n} \sum_{f_i \in F} \left[ \mu_B(f_i) \log \mu_B(f_i) + (1 - \mu_B(f_i)) \log(1 - \mu_B(f_i)) \right] \quad (4)$$

$$H(A \cup B) = -\frac{1}{n} \sum_{f_i \in F} [\mu_A(f_i) \quad (5)$$

$$\vee \mu_B(f_i)] \log \mu_A[\mu_A(f_i) \vee \mu_B(f_i)] + (1 - [\mu_A(f_i) \vee \mu_B(f_i)]) \log(1 - [\mu_A(f_i) \vee \mu_B(f_i)])$$

From the literature, it is a known fact that features selected according to maximum relevance may have more redundant features, which means the dependency among the selected features may be large. The class discriminate power will not be affected much when one of the two highly dependent features are removed. Therefore, Maximum-relevance is combined with minimum-Redundancy [5]. And the minimum redundancy between the selected features are calculated by using Eqn. (6).

$$\min mR(F), \quad mR = \frac{1}{|F|^2} \sum_{f_i, f_j \in F} FI(f_i, f_j). \quad (6)$$

The mRMR score for the feature set  $F$  is obtained by using the Eqn. (7)

$$\max \theta(R, mR), \quad \theta(R - mR). \quad (7)$$

The fuzzy ranks ( $Rank_f$ ) are assigned by using Eqn. (8), i.e., the ratio between  $R$  and  $mR$

$$Rank_f = \arg \max \left( \frac{R}{mR} \right) \quad (8)$$

#### 3.1 Fuzzification

Fuzzification is the process of converting the crisp values into fuzzy values with the help of a fuzzy membership function. The proposed work uses the Gaussian membership function for the fuzzification of given features into fuzzy features. The Gaussian membership function is defined as in Eqn. (9).

$$\mu_{f_i}(f_{ij}) = \exp \left( -\frac{(f_{ij} - \bar{f}_i)^2}{2\sigma_{f_i}^2} \right) \quad (9)$$

where  $f_i$  is  $i^{th}$  feature set in the dataset ( $D$ ),  $f_{ij}$  represents  $i^{th}$  feature set  $j^{th}$  element,  $\bar{f}_i$  represents the mean of the  $f_i$  feature set and  $\sigma_{f_i}^2$  represents the standard deviation of  $f_i$  feature set.

#### 3.2 fast-mRMR Analysis

In the proposed OSMWFS method for selecting the features, we use fast-mRMR feature selection method. These fast-mRMR feature selection method is an extension of mRMR feature selection method. Ramírez-Gallego *et al.* [17] first uses this fast-mRMR feature selection method to overcome the problems with mRMR feature selection method and



computational burden. In mRMR, the main drawback is related to more number of MI computation between a given feature with class label or a pair of input features. So, fast-mRMR uses greedy approach during feature selection and also limits the number of comparisons between the features.

In the fast-mRMR analysis, candidate features are selected from the set of available features. It uses Mutual Information (MI) for selecting strongly relevant and non-redundant features. First, it selects the features that are maximum relevance with the class label using the Eqn. (1). It finds the mutual information between the available feature set and the class label plus features that have maximum MI with the class label are selected. In our proposed work, we use the fuzzy feature as input to the maximum relevant calculation, and also, the method uses fuzzy mutual information for the selection. Then we calculate minimum redundancy using Eqn. (6) for the features that have maximum relevance with the class label.

From the literature, we came to know that maximum relevant features tend to have redundant features and the classification accuracy will not be affected more when one of the redundant features is removed. So, with this knowledge, we are selecting minimum redundant features from the available max relevant features. The final mRMR score is calculated by using Eqn. (7). It is the difference between minimum redundancy and maximum relevance score. The feature that is having maximum mRMR scores are selected as candidate features. Finally, it assigns a fuzzy ranking for the selected features using the Eqn. (8).

### 3.3 Parallel processing

In the proposed work to attain the parallelism, we use the GPU-CUDA (Compute unified device architecture) parallel version. The algorithms that run Parallel on GPUs (Graphics Processing Unit) achieve up to 100 times speed over the traditional CPU (Central Processing Unit) algorithms. It is recommended to use a parallel environment, especially when the size of data exceeds the computational power of the CPU. So, in our proposed OSMWFS method to achieve fast computation of feature selection, we adapted the parallel environment.

In our OSMWFS work, we use the parallel computing platform and programming model called CUDA, introduced by NVIDIA [1]. CUDA provides direct access to the parallel memory in GPUs and the virtual instruction set. For communication with other threads, the kernel uses shared memory, global memory, and registers. Accessing data from registers and shared memory is very fast, but their availability is very

low. Many processing units may keep idle during kernel execution if the program uses too many of these resources. So, care must be taken during the kernel design to balance the use of registers and shared memory. Generally, any CUDA program of stages like 1) GPU global memory allocation, 2) Swapping data from RAM (Random Access Memory) and global memory, 3) execution of GPU kernels, 4) Moving the results from Global memory to RAM, and 5) Deallocating the GPU global memory.

In our proposed method, all the features from the RAM are transferred to GPU global memory, then execute the kernel, then finally, the results are transferred back to RAM.

### 3.4 Framework of OSMWFS Method

Figure 1 explains the OSMWFS method framework for on-line streaming features using multiple sliding windows that are processed in parallel. In Figure 1  $W_1, W_2, W_3, \dots, W_N$  represents 'N' Sliding windows and  $CF_1, CF_2, CF_3, \dots, CF_N$  represents the candidate features obtained by using fast-mRMR feature selection. In this proposed method, the features are arriving one by one at regular intervals of time. Features are added to the sliding windows in parallel one after another upon arriving until the windows are full. When the windows are full, it stops adding the feature to the current sliding windows, and it starts adding the features into other windows. Now the features that are present in the full windows are processed in parallel into fuzzy features using the Gaussian membership function. These fuzzy features are taken as input for fast-mRMR analysis and final candidate features are selected and are assigned

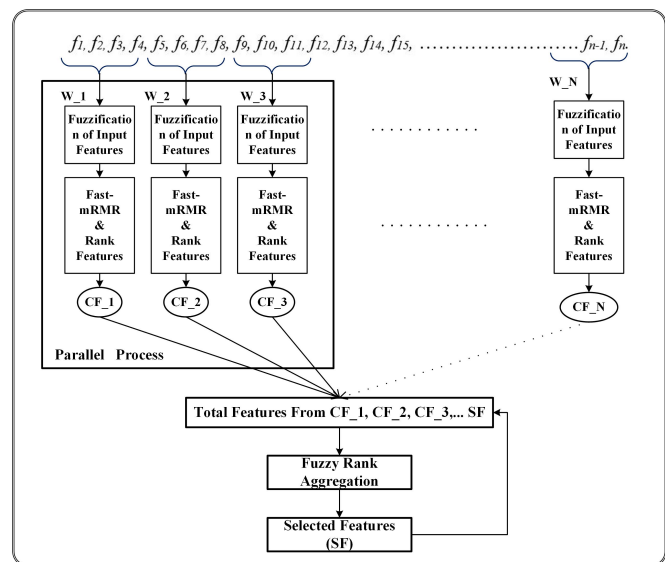


Figure 1: Framework of proposed OSMWFS Method.

with fuzzy rankings by using Eqn. (8) based upon the minimum redundancy and maximum relevance values of the features. By using fast-mRMR feature selection method we are able to remove the redundant and irrelevant features from the available features. Finally, the selected ranked features from the fast-mRMR feature selection method are added to the candidate feature sets (CF).

The relevant features that are selected into one candidate feature set may go to irrelevant with the features that are selected in another candidate feature set in due course and vice versa. So to overcome this, the ranked features presented in each candidate feature set are aggregated with previously selected feature sets (SF) using fuzzy Gaussian rank aggregation method [2, 21]. Top 'n' ranked features are selected from the aggregate ranked features. These top features are then assigned to the selected feature set. The process continues as new features are streaming, and the candidate feature sets continue to change. The process is not stopped until some stopping criteria are met. Finally, at the end the features that are available in the selected feature set are considered as the optimal feature subset.

### 3.4.1 Algorithm

Algorithm 1 explains how the OSMWFS method is works. Line no 5–8 the reading of streaming features into the sliding windows (Buffer<sub>i</sub>). Line no 9–13 represents the parallel execution of each sliding window for the selection of optimal features using fast-mRMR analysis. Algorithm 2 represents the fast\_mRMR analysis and returns candidate features as output. The ranked candidate features from each sliding window of fast-mRMR analysis are assigned with final ranking by aggregating the feature ranks present in the candidate feature sets and the ranked features that are selected from the previous step. Fuzzy rank aggregation is done by using Eqn. (10). Here fuzzy rank aggregation is used for selecting the top 'n' features. In algorithm 1, line 13 represents the rank aggregation process. The optimal features are selected from line 13 of algorithm 1.

$$\mu_{f_i}(x) = \frac{1}{\sqrt{2\pi\sigma_{f_i}^2}} \exp\left(-\frac{1}{2} \left[\frac{(x - \bar{x}_{f_i})^2}{\sigma_{f_i}^2}\right]\right) \quad (10)$$

where  $\bar{x}_{f_i}$  mean and  $\sigma_{f_i}^2$  variance position of features  $f_i$ .

---

#### Algorithm 1: OSMWFS Method Algorithm

---

**Input** : 'c' Target Class; 'sws' sliding-window size, 'N' Number of Sliding windows.  
**Output** : SF= {} selected features;  
1 Initialize:  
2 CF\_1, CF\_2, ..., CF\_N /\* candidate feature sets \*/  
3 Buffer\_1, Buffer\_2, ..., Buffer\_N: 'N' sliding windows each of size (sws)  
4 **repeat**  
5   **for**  $i=0; i < N; i++$  **do**  
6      $f \leftarrow$  new feature arrives at time  $t$   
7     Buffer<sub>i</sub>=read( $f$ )  
8   **end**  
9   /\* Parallelize lines 9 to 12 \*/  
10   **for**  $i=0; i < N; i++$  **do**  
11     CF<sub>i</sub> = fast\_mRMR (Buffer<sub>i</sub>)  
12     CF<sub>i</sub> = Rank (CF<sub>i</sub>) // Rank the Candidate Features Using Eqn. (8).  
13   **end**  
14   /\* Rank Aggregation \*/  
15   SF= merge\_rank{SF, CF\_1, CF\_2, ..., CF\_N}  
16 **until** stopping criteria are met;

---



---

#### Algorithm 2: fast\_mRMR (Buffer<sub>i</sub>)

---

**Input** : Buffer<sub>i</sub>  
**Output** : CF<sub>i</sub>= {} // Candidate feature sets;  
1 **for**  $\forall, x \in \text{Buffer}_i$  **do**  
2   Convert the crisp value (x) into fuzzy value using Gaussian membership function  
3   Find the max relevance features using Eqn. (1)  
4   Find the min redundant features using Eqn. (6)  
5   Final candidate features (CF<sub>i</sub>) are selected using Eqn. (7)  
6 **end**  
7 **return** CF<sub>i</sub>

---

## 4 Dataset Description

### 4.1 Dataset

In the proposed work, we use eight publicly available benchmark datasets. These datasets are collected from the websites LIBSVM<sup>1</sup> and UCI<sup>2</sup> machine learning repository.

<sup>1</sup> www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

<sup>2</sup> https://archive.ics.uci.edu/ml/index.php

**Table 1:** Dataset Description

Dataset	No. of Features	No. of Instances
Ionosphere	34	351
Madelon	500	4400
Lymphoma	4026	62
Gisette	5000	7000
Prostate	6033	102
Leukemia	7129	72
Real-sim	20958	30000
rcv1	47236	30000

Table 1 describes the datasets in terms of features and number of instances present in the dataset. All the datasets are 2 class classification datasets.

## 4.2 Experimental Setup

For the implementation of the proposed OSMWFS method, all the experiments are conducted on a computer with an Intel Core i7 7th Generation, NVIDIA GeForce RTX 2080 (8 GB) GPU and 32 GB RAM. For evaluating the performance of the proposed OSMWFS method we use performance measures like Accuracy, Recall, Precision, F1-Score, and ROC curve. The experimental results are compared with four state-of-the-art online streaming feature selection algorithms, namely Alpha-Investing, OSFS, SAOLA, and OSFSW for eight benchmark datasets as described in Table 1.

OSMWFS method is compared with the other state-of-the-art online streaming feature selection that are implemented using LOFS (Library of Online streaming Feature Selection) [30], an open-source library for streaming features implemented using MATLAB software. As evaluation measures, we use the number of features selected, Accuracy, Precision, Recall, F1-Score, and runtime. We use 10-fold cross-validation for training the model. To allow a fair comparison, all algorithms use the same experimental settings. All experiments were performed more than 15 times, with a random permutation of features for each dataset.

For evaluating the performance of the proposed method, we use three classification algorithms namely k-nn, SVM and Decision Tree. These learning algorithms are integrated into MATLAB 2016's App tool. We use classification measures obtained from each learning algorithm for comparing with other state-of-the-art online feature selection methods. For the statistical tests, we use  $G^2$  test and Fisher z-test for discrete and continuous data respectively at a significance level of ( $\lambda = 0.01$ ). Finally, we need to set the size of the sliding window. We compare the time

taken by these streaming feature selections for finding the optimal feature for different sliding window sizes (sws) that vary from 5, 10, 20, 30, 40, 50.

## 5 Result Analysis

The proposed OSMWFS method is compared with four state-of-the-art streaming feature selection methods, namely Alpha-Investing, OSFS, SAOLA, and OSFSW, over eight publicly available benchmark datasets. We use classification measures for comparing the performance of the proposed method with four state-of-the-art streaming feature selection methods. The measures are namely Accuracy, time taken, and number of features selected. First, we analyze the time taken for different window sizes (sws) to find the optimal number of features. Table 2 describes in detail the time taken and the number of features selected by the proposed algorithm for different sliding window sizes.

From Table 2, we observe that the time taken and the number of features selected is comparatively less and consistent for sws size 20. So, from this observation, the sliding window size is set to 20 in further experimental setups. When the window size increases, the running time and number of selected features also increases.

Table 3 represents the classification accuracy obtained by the state-of-the-art streaming feature selection methods like Alpha-Investing, OSFS, SAOLA, OSFSW, and proposed OSMWFS method using a k-NN classifier. Here the k value is taken as 5 in our experiments. The bolded values indicate the highest accuracy obtained by the OSMWFS method over other streaming feature selection methods. The italic values represent the second-highest accuracy obtained by the OSMWFS method. It is inferred from Table 3 that, for lymphoma and leukemia datasets proposed OSMWFS method gives nearly equivalent accuracy to that of the SAOLA method. We can observe that the prediction accuracy of the Madelon dataset is very low nearly 60 % for all the methods because it is a synthetic dataset so it includes more noisy and redundant features.

Table 4 represents the classification accuracy obtained by the state-of-the-art streaming feature selection methods like Alpha-Investing, OSFS, SAOLA, OSFSW, and proposed OSMWFS method using an SVM classifier. The bolded values indicate the highest accuracy obtained by the proposed method over other streaming feature selection methods. The italic values represent the second-highest accuracy obtained by the proposed method. However, for lymphoma and leukemia datasets proposed method gives nearly equivalent accuracy to that of the SAOLA method. We can observe

**Table 2:** Time taken and No. of features selected (#) for different sliding window

Dataset	Sliding Window Sizes											
	sws=5		sws=10		sws=20		sws=30		sws=40		sws=50	
	Time	#	Time	#	Time	#	Time	#	Time	#	Time	#
Ionosphere	0.5±0.03	5	0.4±0.05	5	<b>0.4±0.08</b>	4	0.3±0.05	4	0.3±0.03	3	0.5±0.001	3
Madelon	10±0.2	13	08±0.6	14	<b>07±0.3</b>	12	07±0.01	13	08±0.25	14	10±0.9	13
Lymphoma	65±0.2	10	70±0.35	8	<b>55±0.6</b>	5	75±0.4	6	85±0.1	5	78±0.25	5
Gisette	72±0.15	13	75±0.3	10	<b>66±0.55</b>	6	83±0.6	8	88±0.4	6	94±0.35	6
Prostate	70±0.05	4	73±0.5	5	<b>68±0.05</b>	4	75±0.35	4	80±0.55	5	85±0.5	4
Leukemia	73±0.7	8	77±0.15	6	<b>70±0.5</b>	4	70±0.15	5	84±0.2	5	90±0.55	4
real-sim	120±0.05	72	110±0.25	64	<b>105±0.05</b>	43	125±0.15	40	130±0.75	51	140±0.5	42
rcv1	150±0.05	86	160±0.5	65	<b>148±0.45</b>	48	165±0.85	45	170±0.03	45	170±0.15	47

**Time** : seconds; **#**: No. of Features Selected; **sws**: Sliding Window Size.

**Table 3:** Accuracy of algorithms with k-NN Classifier

Dataset	Alpha- Investing	OSFS	SAOLA	OSFSW	OSMWFS
Ionosphere	86.40	86.20	87.50	87.20	<b>87.65</b>
madelon	49.30	54.20	55.50	56.10	<b>58.40</b>
Lymphoma	96.30	95.50	<b>99.80</b>	98.30	<b>99.20</b>
gisette	76.40	80.50	87.90	88.10	<b>90.70</b>
Prostate	85.40	95.60	92.70	93.10	<b>96.20</b>
leukemia	96.80	95.60	<b>98.80</b>	96.00	<b>98.20</b>
real-sim	68.10	74.20	75.60	78.50	<b>82.50</b>
rcv1	72.20	75.60	78.50	80.50	<b>83.10</b>

**Table 4:** Accuracy of algorithms with SVM Classifier

Dataset	Alpha- Investing	OSFS	SAOLA	OSFSW	OSMWFS
Ionosphere	90.50	92.30	92.30	92.30	<b>92.80</b>
madelon	55.20	61.40	62.10	61.50	<b>62.50</b>
Lymphoma	96.80	95.20	<b>100.00</b>	96.50	<b>99.50</b>
gisette	80.50	83.40	85.40	86.30	<b>88.70</b>
Prostate	91.00	95.80	93.00	95.20	<b>96.40</b>
leukemia	97.00	97.10	<b>100.00</b>	97.20	<b>99.50</b>
real-sim	58.20	65.25	68.50	72.40	<b>78.50</b>
rcv1	74.20	76.40	78.50	82.25	<b>85.10</b>

**Table 5:** Accuracy of algorithms with Decision Tree Classifier

Dataset	Alpha- Investing	OSFS	SAOLA	OSFSW	OSMWFS
Ionosphere	90.10	89.50	86.50	90.20	<b>91.40</b>
madelon	50.20	58.10	57.20	58.20	<b>58.50</b>
Lymphoma	93.50	80.40	95.20	<b>96.60</b>	<b>96.20</b>
gisette	82.40	83.45	82.25	84.50	<b>85.75</b>
Prostate	88.00	90.50	87.10	91.30	<b>92.10</b>
leukemia	95.60	96.20	94.40	97.20	<b>98.10</b>
real-sim	60.50	68.50	71.50	75.00	<b>78.40</b>
rcv1	78.90	76.50	77.40	81.40	<b>82.60</b>



**Table 6:** Time taken and No. of features selected by different algorithms

Dataset	Algorithms									
	Alpha- Investing		OSFS		SAOLA		OSFSW		OSMWFS	
	Time	#	Time	#	Time	#	Time	#	Time	#
Ionosphere	<b>0.050±0.005</b>	10	0.15±0.005	4	0.98±0.05	4	0.9±0.05	4	0.4±0.05	<b>4</b>
madelon	<b>0.025±0.003</b>	13	10.25±0.15	16	0.15±0.05	18	4.5±0.4	14	3±0.40	<b>12</b>
Lymphoma	30.12±0.03	5	45.25±0.15	8	52.85±0.05	39	35.65±0.07	15	<b>15±0.1</b>	<b>11</b>
gisette	35.25±0.02	15	38.45±0.04	24	30.70±0.30	40	45.50±0.25	16	<b>25±0.15</b>	<b>10</b>
Prostate	55.35±0.55	5	55.5±0.2	12	110±6	15	70±3	14	<b>40±0.15</b>	<b>10</b>
leukemia	72.25±0.15	10	45.45±0.15	28	90±5	35	83±2	17	<b>62±0.55</b>	<b>12</b>
real-sim	130.25±0.03	80	130.00±10	54	210±15	60	140±5	52	<b>90±0.85</b>	<b>43</b>
rcv1	135.35±0.04	73	150.00±15	90	280±20	55	200±15	57	<b>100±0.5</b>	<b>48</b>

Time: seconds; #: No. of Features Selected;

that the prediction accuracy of the Madelon dataset is very low nearly 60 % for all the methods because it is a synthetic dataset so it includes more noisy and redundant features.

Table 5 represents the classification accuracy obtained by the state-of-the-art streaming features selection methods like Alpha-Investing, OSFS, SAOLA, OSFSW, and proposed OSMWFS method using Decision Tree classifier. The bolded values indicate the highest accuracy obtained by the OSMWFS method over other streaming feature selection methods. The italic values represent the second-highest accuracy obtained by the OSMWFS method. We can observe that the prediction accuracy of the Madelon dataset is very low nearly 60 % for all the methods because it is a synthetic dataset, so it includes more noisy and redundant features.

We further analyze the performance of the proposed OSMWFS method over other online streaming feature selection methods. Table 6 shows the run time and number of features selected by the state-of-the-art streaming feature selection methods and the proposed OSMWFS method. The run time for the datasets like the ionosphere, Madelon, is less for Alpha-Investing than the other methods and proposed method. This is because Alpha –Investing only considers the newly added features and the discarded features are never considered again. However, the number of features selected for the OSMWFS method is very low when compared with the Alpha-Investing method. Moreover, the prediction accuracy is low for Alpha-Investing when compared with OSMWFS method (as observed from Tables 3-5). From Table 6, we inferred that the OSMWFS method outperforms for most of the datasets over the other streaming feature selection methods.

Table 7 represents the precision, recall, and F1-Score for 'k-NN,' SVM, and Decision Tree classifier for eight benchmark datasets over four state-of-the-art online feature selection (OFS) methods and proposed OSFSWFS method. Preci-

sion measures the probability of data predicted as positive to be positive. It is the measure of the ratio of True Positive by true positive plus false positive (i.e.,  $TP/(TP+FP)$ ).

A recall is referred to as True Positive Rate or sensitivity, and it is the measure of the ratio of True Positive by True Positive plus False Negative (i.e.,  $TP/(TP+FN)$ ). The columns represent measures like Precision, Recall and F1-Score over the three classifiers, whereas the rows represent the datasets and online feature selection methods. From Table 7, we infer that the performance of the proposed method is good when compared to the other OFS methods.

Receiver Operating Characteristic Curve (ROC Curve) is used to show how good a given learning model. It uses sensitivity and Specificity values for the representation. The Area Under The ROC Curve (AUC) is used to measure how best a given model varies between the classes. If the area is high, then the model is best and vice versa.

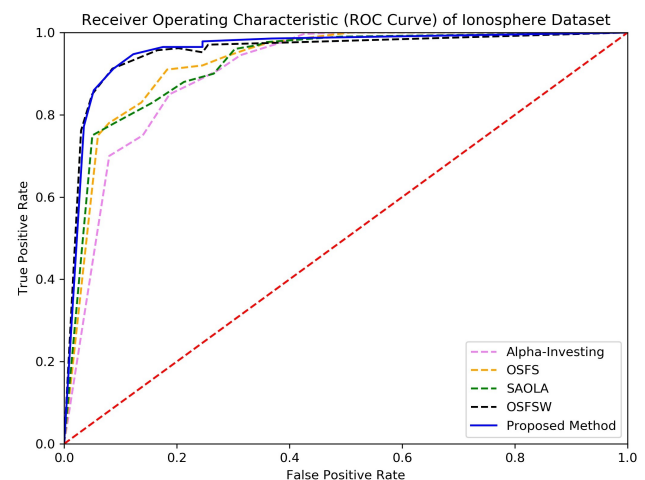
**Figure 2:** ROC Curve for Ionosphere Dataset

Table 7: Precision, Recall and F1-Score for Classifiers

Dataset	OFS	Classifiers								
		k-NN			SVM			Decision Tree		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Ionosphere	Alpha Investing	75.47	74.07	74.77	75.76	83.33	79.37	76.92	78.13	77.52
	OSFS	75.00	71.43	73.17	77.50	72.09	74.70	75.71	79.10	77.37
	SAOLA	76.36	75.00	75.68	79.52	81.48	80.49	80.26	81.33	80.79
	OSFSW	76.19	77.94	77.42	77.61	78.79	78.20	77.27	85.42	82.83
	OSMWFS	79.10	78.69	78.52	80.58	87.37	83.84	80.39	87.93	82.26
Madelon	Alpha Investing	79.17	79.17	79.17	78.95	77.69	76.67	81.89	85.54	72.73
	OSFS	81.48	83.54	82.50	80.00	81.36	80.67	83.65	87.00	85.29
	SAOLA	77.50	73.47	79.12	80.36	76.27	78.26	80.41	85.76	85.71
	OSFSW	76.47	82.54	79.39	85.71	76.67	75.00	81.25	89.04	84.97
	OSMWFS	85.71	84.93	81.05	88.87	82.35	81.58	88.82	89.51	88.26
Lymphoma	Alpha Investing	85.19	83.53	82.77	84.71	93.51	88.89	85.69	89.89	89.89
	OSFS	84.04	87.78	85.87	80.00	84.21	85.05	87.18	87.18	87.18
	SAOLA	81.65	88.12	84.76	86.36	86.48	88.37	83.78	83.78	83.78
	OSFSW	80.95	88.31	84.47	87.41	86.84	88.59	88.19	81.67	86.41
	OSMWFS	88.05	89.91	86.34	90.50	88.89	91.85	89.89	91.59	90.69
gisette	Alpha Investing	84.71	93.51	88.89	80.95	88.31	84.47	83.16	84.04	83.60
	OSFS	80.00	84.21	82.05	83.05	89.91	86.34	86.49	78.09	82.19
	SAOLA	86.36	90.48	88.37	75.34	83.33	79.14	82.52	85.86	84.16
	OSFSW	90.41	86.84	88.59	84.48	77.78	80.99	83.16	84.04	83.60
	OSMWFS	87.50	88.89	87.85	82.83	86.32	84.54	87.49	88.19	86.69
Prostate	Alpha Investing	77.78	80.77	79.25	77.78	80.77	79.25	77.27	76.44	81.60
	OSFS	83.12	86.49	84.77	83.12	86.49	84.77	80.46	79.74	84.85
	SAOLA	80.65	91.46	85.71	80.65	91.46	85.71	79.10	84.48	82.17
	OSFSW	82.91	87.39	85.09	82.91	87.39	85.09	80.95	79.39	74.73
	OSMWFS	84.21	88.07	86.10	77.27	86.44	81.60	89.55	83.20	84.51
leukemia	Alpha Investing	77.97	82.14	80.00	76.79	78.48	84.31	80.95	92.73	86.44
	OSFS	79.38	93.90	86.03	78.13	71.43	84.63	84.29	88.06	86.13
	SAOLA	81.33	83.56	82.43	76.62	80.77	83.10	84.31	97.73	90.53
	OSFSW	83.33	82.35	82.84	80.26	80.92	82.99	84.75	95.24	89.69
	OSMWFS	81.44	92.94	86.81	86.89	80.30	83.46	85.55	97.63	90.44
real-sim	Alpha Investing	92.98	98.15	95.50	94.12	97.96	96.00	96.43	93.10	94.74
	OSFS	94.29	97.06	95.65	95.24	98.77	96.97	97.98	97.98	97.98
	SAOLA	94.34	96.15	95.24	97.62	95.35	96.47	97.37	97.37	97.37
	OSFSW	95.65	98.88	97.24	97.30	94.74	96.00	97.44	98.70	98.06
	OSMWFS	95.65	95.65	95.65	95.83	95.83	95.83	97.10	98.53	97.81
rcv1	Alpha Investing	93.33	97.67	95.45	95.77	94.44	95.10	94.34	96.15	95.24
	OSFS	93.98	98.73	96.30	97.06	92.96	94.96	92.06	93.55	92.80
	SAOLA	86.67	96.30	91.23	94.34	96.15	95.24	93.75	94.94	94.34
	OSFSW	93.98	97.50	95.71	97.22	92.11	94.59	97.03	94.23	95.61
	OSMWFS	97.92	92.16	94.95	96.25	95.06	95.65	97.50	86.67	91.76

Figures 2 – 9 represent the Receiver Operating Characteristic (ROC) curve given by SVM classification over the proposed method and other state-of-the-art online feature selection methods for eight benchmark datasets. On X-axis False Positive Rate and Y-axis True Positive Rate values are represented.

From Figures 2 – 9, it is observed that the area under the ROC curve is low for most of the online feature selec-

tion method over the proposed OSMWFS method. From Figures 2 – 9, we observe that the AUC is more for datasets like Ionosphere, Madelon, leukemia, rc1and real-sim data. However, for other datasets, the AUC is not high for the proposed OSMWFS method but is nearby the top online feature selection method.

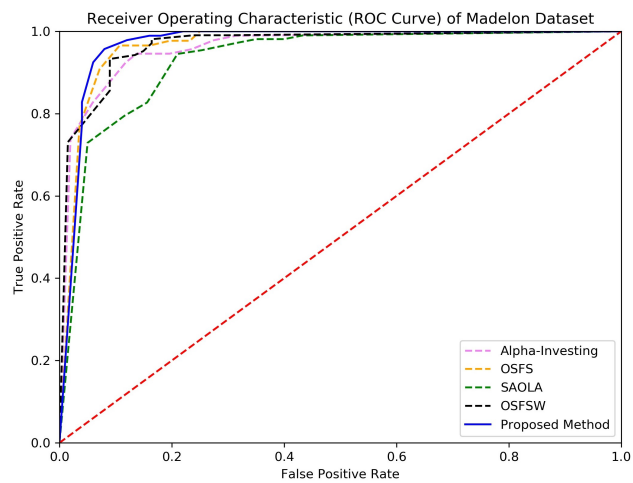


Figure 3: ROC Curve for Madelon Dataset

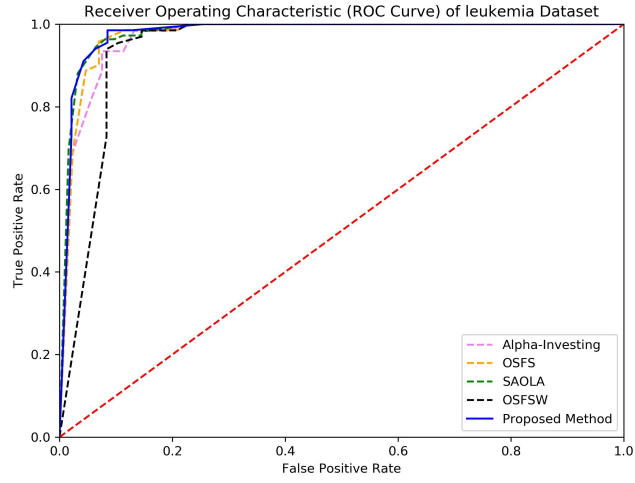


Figure 6: ROC Curve for Leukemia Dataset

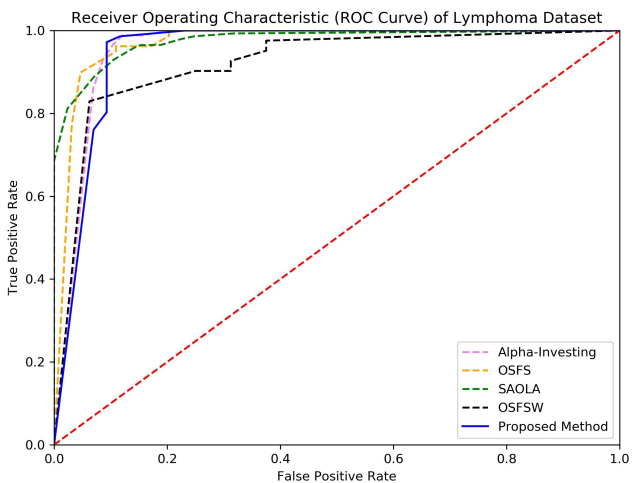


Figure 4: ROC Curve for Lymphoma Dataset

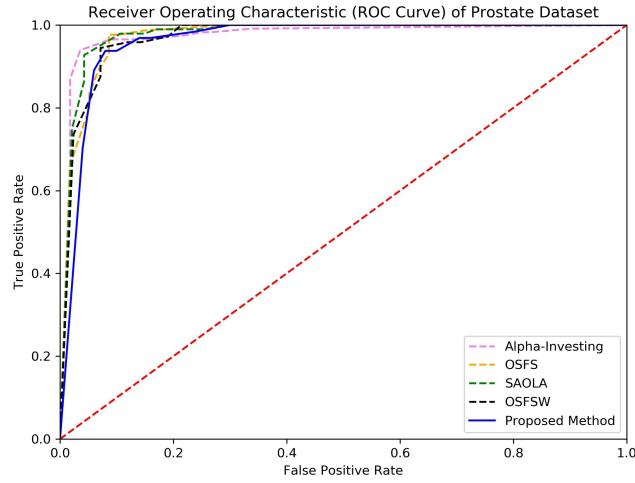


Figure 7: ROC Curve for Prostate Dataset

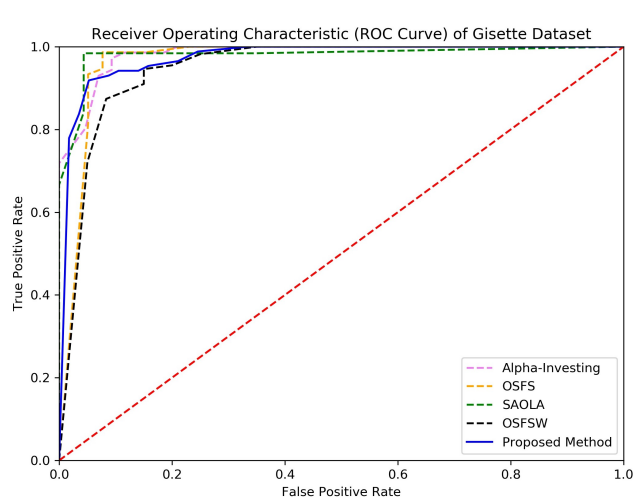


Figure 5: ROC Curve for Gisette Dataset

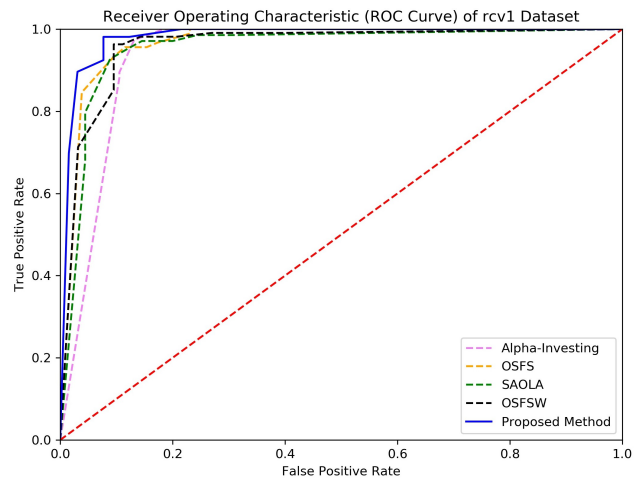


Figure 8: ROC Curve for rcv1 Dataset

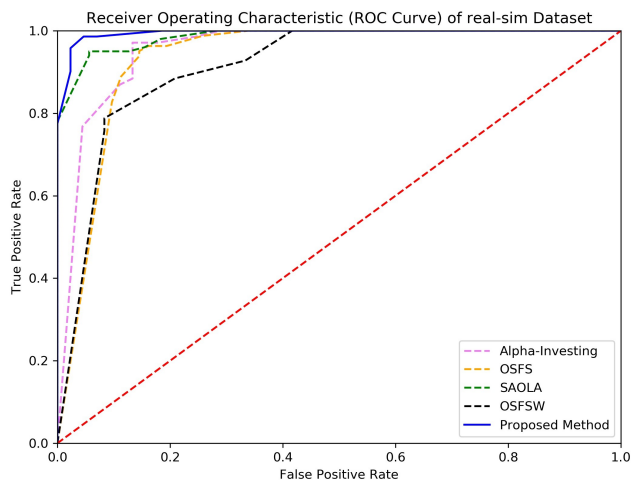


Figure 9: ROC Curve for real-sim Dataset

## 6 Conclusion and Future Scope

In this paper, we proposed the OSMWFS method using multiple sliding-windows that are processed in parallel, and the fuzzy fast-mRMR feature selection method is used for selecting features with minimum redundant and maximum relevant. The use of fuzzy fast-mRMR feature selection is to reduce the number of MI computations between the features and the class labels. The use of multiple sliding windows is to speedup the process of retrieving the streaming features. In this digital era, data keeps on changing rapidly and dynamically to cope up with this we need parallel processing. And finally the fuzzy rank aggregation is used to eliminate the redundant features between the previously selected feature and present selected feature set during the feature selection process. The selected features have minimum redundancy and maximum relevance with the class variable. The proposed method has a low number of features and high classification accuracy. The experimental results show that the OSMWFS method outperforms other online streaming feature selection methods. Our empirical study demonstrated that; 1) OSMWFS method is best suitable for high dimensional streaming data, 2) Parallel processing reduces the time required for FS, and 3) OSMWFS enhances the performance of the model.

Even though the OSMWFS method outperforms other online streaming features selection methods, the selection of sliding window size is static. In future, we will extend this work to fix the window size dynamically based on the data flow.

## References

- [1] CUDA Toolkit Documentation, <https://docs.nvidia.com/cuda/index.html>.
- [2] Beg M. S. Ahmad N., Soft computing techniques for rank aggregation on the world wide web, *World Wide Web*, 2003, 6(1), 5–22.
- [3] Chamby-Diaz J. C., Recamonde-Mendoza M., Bazzan A., Dynamic correlation-based feature selection for feature drifts in data streams, 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), IEEE, 2019, 198–203.
- [4] Dash M. Liu H., Feature selection for classification, *Intelligent data analysis*, 1997, 1(3), 131–156.
- [5] Ding C. Peng H., Minimum redundancy feature selection from microarray gene expression data, *Journal of bioinformatics and computational biology*, 2005, 3(02), 185–205.
- [6] Ding W., Stepinski T. F., Mu Y., Bandeira L., Ricardo R., Wu Y., Lu Z., Cao T., Wu X., Subkilometer crater discovery with boosting and transfer learning, *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011, 2(4), 1–22.
- [7] Eskandari S. Javidi M. M., Online streaming feature selection using rough sets, *International Journal of Approximate Reasoning*, 2016, 69, 35–57.
- [8] Hoque N., Ahmed H., Bhattacharyya D., Kalita J., A fuzzy mutual information-based feature selection method for classification, *Fuzzy Information and Engineering*, 2016, 8(3), 355–384.
- [9] Javidi M. M. Eskandari S., Online streaming feature selection: a minimum redundancy, maximum significance approach, *Pattern Analysis and Applications*, 2019, 22(3), 949–963.
- [10] Kwak N. Choi C.-H., Input feature selection for classification problems, *IEEE transactions on neural networks*, 2002, 13(1), 143–159.
- [11] Li H., Wu X., Li Z., Ding W., Group feature selection with streaming features, 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, 1109–1114.
- [12] Lin Y., Hu Q., Liu J., Li J., Wu X., Streaming feature selection for multilabel learning based on fuzzy mutual information, *IEEE Transactions on Fuzzy Systems*, 2017, 25(6), 1491–1507.
- [13] Liu W. Wang T., Online active multi-field learning for efficient email spam filtering, *Knowledge and Information Systems*, 2012, 33(1), 117–136.
- [14] Peng H., Long F., Ding C., Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2005, 8, 1226–1238.
- [15] Perkins S. Theiler J., Online feature selection using grafting, *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, 592–599.
- [16] Rahmaninia M. Moradi P., Osfsmi: online stream feature selection method based on mutual information, *Applied Soft Computing*, 2018, 68, 733–746.
- [17] Ramírez-Gallego S., Lastra I., Martínez-Rego D., Bolón-Canedo V., Benítez J. M., Herrera F., Alonso-Betanzos A., Fast-mrmr: Fast minimum redundancy maximum relevance algorithm for high-dimensional big data, *International Journal of Intelligent Systems*, 2017, 32(2), 134–152.
- [18] Sun X., Liu P., Ma Y., Liu D., Sun Y., Streaming remote sensing data processing for the future smart cities: State of the art and future challenges, *Environmental Information Systems: Concepts*,

- Methodologies, Tools, and Applications, 2019, 1711–1726, IGI Global.
- [19] Tang J., Alelyani S., Liu H., Feature selection for classification: A review, *Data classification: Algorithms and applications*, 2014, 37–64.
  - [20] Venkatesh B. Anuradha J., A Review of Feature Selection and Its Methods, *Cybernetics and Information Technologies*, 2019, 19 (1), 3–26, ISSN 1314-4081.
  - [21] Venkatesh B. Anuradha J., A fuzzy gaussian rank aggregation ensemble feature selection method for microarray data, *International Journal of Knowledge-based and Intelligent Engineering Systems*, 2020, 24(4), 289–301.
  - [22] Wang J., Zhao P., Hoi S. C., Jin R., Online feature selection and its applications, *IEEE Transactions on Knowledge and Data Engineering*, 2013, 26(3), 698–710.
  - [23] Wang J., Zhao Z.-Q., Hu X., Cheung Y.-M., Wang M., Wu X., Online group feature selection, *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013, 1757–1763.
  - [24] Wang J., Wang M., Li P., Liu L., Zhao Z., Hu X., Wu X., Online feature selection with group structure analysis, *IEEE Transactions on Knowledge and Data Engineering*, 2015, 27(11), 3029–3041.
  - [25] Wang M., Li H., Tao D., Lu K., Wu X., Multimodal graph-based reranking for web image search, *IEEE Transactions on Image Processing*, 2012, 21(11), 4649–4661.
  - [26] Wu X., Yu K., Wang H., Ding W., Online streaming feature selection, *Proceedings of the 27th international conference on machine learning (ICML-10)*, Citeseer, 2010, 1159–1166.
  - [27] You D., Wu X., Shen L., Chen Z., Ma C., Deng S., Online feature selection for streaming features with high redundancy using sliding-window sampling, *2018 IEEE International Conference on Big Knowledge (ICBK)*, IEEE, 2018, 205–212.
  - [28] You D., Wu X., Shen L., Deng S., Chen Z., Ma C., Lian Q., Online feature selection for streaming features using self-adaption sliding-window sampling, *IEEE Access*, 2019, 7, 16088–16100.
  - [29] Yu J. Xu W., Incremental knowledge discovering in interval-valued decision information system with the dynamic data, *International Journal of Machine Learning and Cybernetics*, 2017, 8(3), 849–864.
  - [30] Yu K., Ding W., Wu X., Lofs: a library of online streaming feature selection, *Knowledge-Based Systems*, 2016, 113, 1–3.
  - [31] Yu K., Wu X., Ding W., Pei J., Scalable and accurate online feature selection for big data, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2016, 11(2), 16–55.
  - [32] Zhang L., Zhao J., Li W., Online and unsupervised anomaly detection for streaming data using an array of sliding windows and pdds, *IEEE Transactions on Cybernetics*, 2019, 1–6.
  - [33] Zheng H.-T. Zhang H., Online streaming feature selection using sampling technique and correlations between features, *Asia-Pacific Web Conference*, Springer, 2016, 43–55.
  - [34] Zhou J., Foster D., Stine R., Ungar L., Streaming feature selection using alpha-investing, *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ACM, 2005, 384–393.
  - [35] Zhou P., Hu X., Li P., Wu X., Online feature selection for high-dimensional class-imbalanced data, *Knowledge-Based Systems*, 2017, 136, 187–199.
  - [36] Zhou P., Hu X., Li P., Wu X., Online streaming feature selection using adapted neighborhood rough set, *Information Sciences*, 2019, 481, 258–279.