



AXGFS: A hybrid alpha-investing and XGBoost framework for streaming feature selection

 Priyadarshini
Iyyanar Babu¹⁺
 Shirish Sane²

¹Department of Computer Engineering, MET Institute of Engineering,
Affiliated to Savitribai Phule Pune University Nashik, India.

²Department of Computer Engineering, K. K. Wagh Institute of Engineering
Education and Research, Nashik, India.

¹Email: ilppriyadarshini@gmail.com

²Department of Computer Engineering, Gokhale Education Society, R. H.
Sapat College of Engineering, Nashik, India.

²Email: shirish.sane@gmail.com



(+ Corresponding author)

ABSTRACT

Article History

Received: 2 June 2026

Revised: 11 August 2026

Accepted: 31 August 2026

Published: 13 October 2026

Keywords

Alpha-investing
Dynamic feature streams
Feature stream mining
Feature stream selection
Online feature selection
Streaming feature selection
XGBoost.

In real-time applications like predictive maintenance in industry, patient health monitoring, and video surveillance, a large volume of data is being generated, and new features are becoming available over time. Existing feature selection methods assume that all features are available at the same instance of time and that the complete feature space should be known in advance to perform feature selection. Therefore, they are not suitable for environments where new features arrive dynamically. To overcome this problem, the paper presents a feature stream selection framework that incrementally processes incoming features and identifies a subset of relevant features that has higher predictive power for classification. This is a hybrid framework that combines the strength of statistical feature selection using Alpha-Investing with model-based feature importance ranking using XGBoost to select a relevant and non-redundant feature set. The proposed framework was evaluated using eight benchmark datasets, and classification accuracy was evaluated using three different classifiers. The experimental results show that the proposed hybrid framework increases classification accuracy by up to 5–7% while selecting fewer features than existing feature selection methods. These results indicate that the proposed hybrid approach of integrating Alpha-Investing with XGBoost selects more relevant and non-redundant features, thereby improving the predictive accuracy of the classifier. Thus, the framework can be used in applications where features arrive dynamically, and feature stream selection needs to be performed using the features available up to the current time, such as healthcare monitoring and predictive maintenance.

Contribution/Originality: The study contributes to feature stream selection by providing a hybrid framework that integrates Alpha Investing with XGBoost-based feature importance analysis. The framework identifies a compact and informative feature subset, resulting in improved classification accuracy.

1. INTRODUCTION

Feature selection is one of the most important preprocessing tasks in machine learning that identifies the most relevant features from high-dimensional data by removing irrelevant, redundant, and non-informative features. Selecting an appropriate subset of features reduces computational cost, decreases model training time, and improves the predictive performance of classification and regression models [1–3]. Traditional methods assume that the entire feature space is available before feature selection begins. However, for many applications, such as monitoring patients, industry, predictive maintenance, recommendation systems, and video surveillance, features are generated sequentially over time.

Feature selection techniques can be applied to both static and streaming data, as shown in Figure 1. In static datasets, the entire feature space is available in advance, and filter, wrapper, and embedded methods are widely used to identify relevant features. In streaming environments, data are broadly categorized into data streams and feature streams [4]. In data streams, the number of features remains fixed while new instances continuously arrive. In feature streams, the number of instances remains fixed, and the features arrive continuously. Each incoming feature is evaluated to determine whether to retain or discard it.

Depending on the availability of class labels and the nature of feature arrival, feature selection methods can be classified into supervised and unsupervised approaches. Supervised methods use class information to evaluate the features, and unsupervised methods work by identifying the representative features based on feature similarity. Some approaches evaluate individual features as they arrive, while others consider group structures within the incoming feature stream.

The motivation for this study arises from the challenges of selecting relevant features in dynamic feature-stream environments, where features become available sequentially over time rather than all features at once. Although several online feature-selection methods have been proposed to process incoming features incrementally, the selected feature subsets do not always provide the best predictive performance. Many existing approaches rely primarily on online statistical evaluation during feature selection, or there is a need to set a relevancy threshold. Thus, there exists an opportunity to further refine the selected feature subset through an additional evaluation mechanism.

The main research problem addressed in this study is to select the most relevant and nonredundant feature set in dynamic feature streams, which will improve classification accuracy. To solve this problem, this paper proposes a hybrid feature stream selection framework that integrates Alpha Investing, a statistical filter for features, with XGBoost-based feature importance analysis. Initially, Alpha Investing evaluates each incoming feature and decides whether to retain or discard it. Then, XGBoost ranks all the selected features according to their predictive importance. The proposed hybrid framework is evaluated on eight benchmark datasets and compared against existing online feature selection methods using classification accuracy and the number of selected features as performance measures.

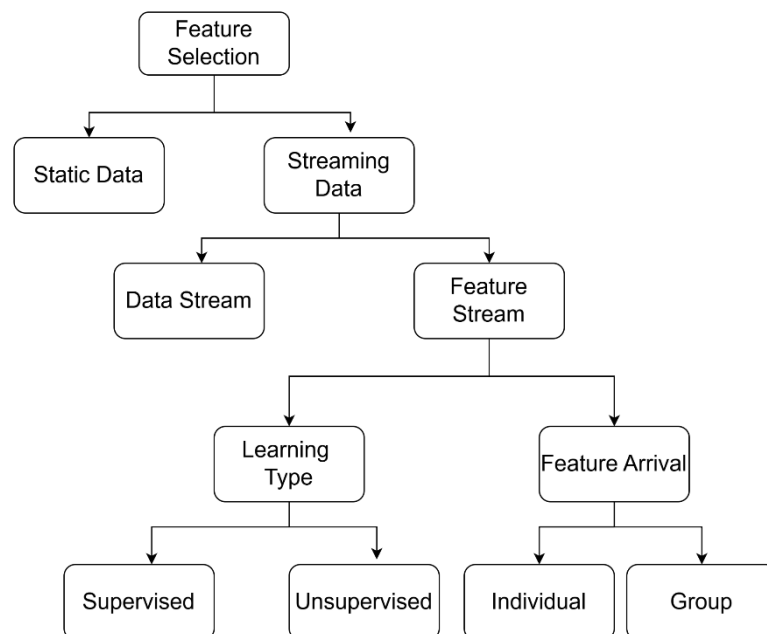


Figure 1. Classification of feature selection methods.

The rest of the paper is organized as follows. Section 2 reviews the existing work in feature stream selection along with the research gap. Section 3 discusses the identified research issues and challenges. The proposed

methodology is presented in Section 4. Section 5 discusses the experimental results and observations. Finally, Section 6 concludes the paper.

2. LITERATURE REVIEW

Feature stream selection is a critical challenge in data stream mining, especially in scenarios where features arrive dynamically over time rather than being available all at once. To effectively manage this continuous arrival of features, existing approaches generally rely on three core scoring mechanisms: statistical tests (such as p-values, t-tests, and chi-square tests) that use hypothesis testing to filter out noise [1, 2]; information theory-based metrics (like entropy, mutual information, and conditional independence) that quantify uncertainty reduction and excel at capturing complex, non-linear relationships without assuming a rigid model structure [4]; and model-based weights (such as linear regression coefficients, decision tree metrics, or SHAP values) that leverage a model's internal mechanisms to rank features with high interpretability [5]. Beyond their scoring mechanisms, these methodologies are fundamentally divided into supervised and unsupervised approaches, depending on their reliance on class labels. Supervised algorithms leverage class labels to assess discriminatory power, yielding high accuracy but often incurring steep computational costs and requiring predefined thresholds or heuristic functions [6]. Prominent examples include Grafting, which uses gradient descent for incremental selection [7]; Alpha Investing, which manages the false discovery rate using p-values and a dynamic wealth parameter [8]; OSFS and Fast-OSFS, which employ conditional independence tests to eliminate redundancy [4]; SAOLA and OFS-DD, which scale to large datasets using mutual information [9, 10]; and OFS-A3M, which analyzes local data structures through neighborhood distances [4, 11]. Conversely, when labels are sparse, delayed, or entirely unavailable in real-time streams, unsupervised methods step in. These techniques rely strictly on internal data characteristics, utilizing strategies like UFSSF, which pairs Pearson correlation with k-means clustering to isolate representative features [12], or OUFSDFC, which groups features dynamically using density-based clustering [13]. While the iterative nature of these clustering and correlation calculations can increase processing time, unsupervised methods offer the crucial flexibility and adaptability required for real-world, unlabeled streaming environments.

Table 1. Comparative analysis of the strengths, weaknesses, and research gaps of existing feature stream selection methods.

Sr. No.	Technique / Method (Reference)	Strengths	Weaknesses	Research Gap
1	Grafting [7]	First online supervised feature selection method; incrementally updates the model without complete retraining.	Requires prior knowledge of the global feature space and careful regularization parameter tuning.	Unsuitable for infinite feature streams where the complete feature space is unknown.
2	Alpha-Investing [8]	Computationally efficient online statistical feature selection with false discovery rate control.	Considers only feature relevance and ignores feature redundancy.	Requires redundancy-aware online feature selection.
3	OSFS (Conditional Independence Test) [4]	Selects strongly relevant and non-redundant features with good prediction accuracy.	Conditional independence testing is computationally expensive for high-dimensional data.	More scalable redundancy elimination techniques are required.
4	Fast-OSFS [4]	Improves execution speed over OSFS through efficient redundancy checking.	Computational complexity still increases with the number of selected features.	Scalability remains challenging for large-scale feature streams.
5	SAOLA (Mutual Information) [9]	Efficient redundancy removal and suitable for high-dimensional datasets.	Depends on a predefined relevance threshold and may retain unnecessary features.	Adaptive threshold selection is needed to improve robustness.
6	OFS-A3M	Utilizes neighborhood	Performance deteriorates	Improved redundancy

Sr. No.	Technique / Method (Reference)	Strengths	Weaknesses	Research Gap
	(Neighbourhood Rough Set) [4]	relationships to improve feature evaluation.	when highly redundant features are present.	handling is required for correlated feature streams.
7	OFS-Density [11]	Better redundancy reduction than earlier rough-set methods.	Higher execution time than SAOLA and OFS-A3M.	Faster online implementation is required.
8	OFS-DD (Normalized Mutual Information) [10]	Dynamically classifies arriving features and improves prediction accuracy.	Mutual information calculations remain computationally intensive.	Requires computationally efficient adaptive feature evaluation.
9	UFSSF (Correlation and Clustering) [12]	Does not require class labels and reduces storage using representative features.	Clustering operations introduce computational overhead and parameter sensitivity.	More scalable unsupervised feature selection is needed.
10	OUFSDFC (Dynamic Density-based Clustering) [13]	Adapts to evolving feature streams without supervision.	Density-based clustering is computationally expensive for large feature streams.	Efficient unsupervised methods with lower computational complexity are required.

Table 2. Comparison of feature stream selection methods based on scoring mechanisms.

Scoring Mechanism	Representative Methods (Reference)	Major Limitation
Regularization-based	Grafting [7]	Requires prior knowledge of the global feature space and careful parameter tuning.
Statistical Test-based	Alpha-Investing [8]	Evaluates feature relevance but does not adequately address feature redundancy.
Conditional Independence-based	OSFS [4] Fast-OSFS [4]	Computationally expensive due to repeated conditional independence tests, affecting scalability.
Information-Theoretic	SAOLA [9] OFS-DD [10]	Depends on mutual information computations and relevance thresholds, resulting in increased computational cost.
Neighbourhood Rough Set-based	OFS-A3M [4] OFS-Density [11]	Performance is affected by highly redundant data and increased execution time.
Correlation and Clustering-based	UFSSF [12]	Clustering operations introduce computational overhead and parameter sensitivity.
Density-based Clustering	OUFSDFC [13]	Computationally intensive and less scalable for large feature streams.

In summary, the literature presents a diverse range of feature stream selection strategies, each offering distinct advantages and trade-offs. Table 1 provides a comparative analysis of the strengths, weaknesses, and research gaps of representative feature stream selection methods, while Table 2 summarizes the major categories of feature stream selection methods based on their underlying scoring mechanisms, representative algorithms, and associated limitations. The comparison indicates that regularization- and statistical test-based methods are computationally efficient but provide limited redundancy handling, whereas conditional independence, information-theoretic, and neighborhood rough set-based methods improve feature relevance evaluation at the expense of increased computational complexity. In contrast, clustering-based unsupervised approaches eliminate the dependence on class labels but incur additional computational overhead. Recent survey studies [14, 15] further emphasize the importance of scalability, adaptability, and robustness in handling dynamically arriving features [16, 17]. These findings indicate that there is still a need for a feature stream selection framework capable of selecting a compact and informative feature subset while effectively balancing feature relevance and redundancy in dynamic, high-dimensional feature stream environments.

2.1. Research Gap

Even though many feature stream selection methods exist, there remains a research gap; some existing methods, such as Grafting [7], select relevant features but do not remove redundant features, whereas information-theoretic and rough-set-based approaches handle redundancy, but a predefined relevance threshold is needed, which is difficult to obtain [9]. Most existing feature stream selection methods use one of the scoring methods mentioned in Table 2 for feature selection [4, 14, 15]. Recently, the hybrid approach for feature stream selection has gained attention due to its increased predictive performance [18]. Therefore, a hybrid framework is proposed that uses the strengths of statistical relevance testing and model-based feature-importance ranking. This combination would help identify the most relevant and non-redundant features, which would improve the predictive power of the classifier.

3. ISSUES AND CHALLENGES IN FEATURE STREAM SELECTION

Based on the identified research gaps, the most important challenge exists when the entire feature space is not known in advance, and features arrive dynamically. A hybrid framework would be needed that combines the strengths of more than one scoring mechanism to select the best features. The proposed hybrid feature stream selection framework is presented in the next section.

4. PROPOSED METHODOLOGY

A hybrid framework combining Alpha Investing and XGBoost for feature selection (AXGFS) is proposed (Figure 2).

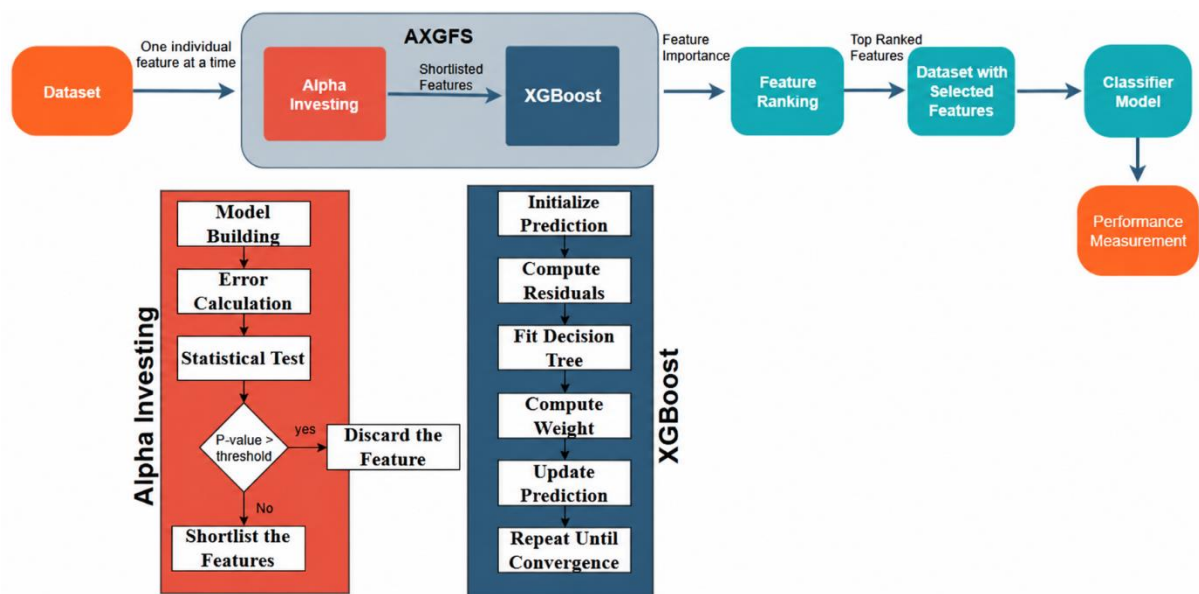


Figure 2. Architecture of the proposed AXGFS feature stream selection framework.

Alpha Investing [8] was chosen because it is more suitable for the selection of dynamic features, as well as controlling the false discovery rate. It is capable of evaluating features incrementally, one by one, as they arrive. It selects statistically significant features. It helps filter irrelevant features and select statistically significant features.

XGBoost, or Extreme Gradient Boosting [19], is an optimized gradient-boosting algorithm used for regression and classification. It is chosen because of its strong predictive power. It also has a metric to determine how each feature contributes to the prediction.

In this paper, a hybrid framework combining these two approaches is studied. It utilizes the strengths of both statistical testing and model-based feature-importance scoring.

The feature stream selection process is as follows:

Relevancy Checking: After the arrival of each feature, its relevance to the class label is evaluated using alpha investing [8]. The features that satisfy the description of steps 1 to 15 in the algorithm are retained, and others are discarded. This serves as initial-level filtering to remove the irrelevant features.

Feature Importance Ranking: After the relevance-checking phase, the selected feature streams are used to build an XGBoost model. The selected features are then ranked according to the model's gain-based feature-importance measure.

Candidate Feature Selection: After all the features have been ranked, the number of features is further reduced. Different feature subsets are generated by retaining the top 25%, 40%, 50%, and 75% based on the feature importance score. The effectiveness of the proposed framework in Figure 2 is evaluated in terms of the predictive accuracy achieved by the KNN, SVM, and CART classifiers using the selected features.

Table 3. Mathematical notations used in the AXGFS framework.

Symbol	Description
X	Dataset
Y	Target labels
n	Number of samples
X_i	Incoming feature
$S_{initial}$	Feature subset selected by Alpha Investing
$S_{refined}$	Final selected feature subset
W_i	Alpha wealth at iteration i
α_i	Significance level
α_Δ	Reward parameter
P_i	p-value
I_i	XGBoost importance score
T	Percentile threshold
K	Number of boosting rounds
M	Number of selected features

Algorithm 1: Pseudocode of the proposed AXGFS Feature Stream Selection Framework

Input:

Dataset $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$

Initial wealth W_0

Reward parameter α_Δ

Number of boosting rounds K .

Output:

Refined feature subset $S_{refined}$

```

1: Initialize  $W_i \leftarrow W_0$ 
2: Initialize  $S_{initial} \leftarrow \emptyset$ 
3: Initialize  $i \leftarrow 1$ 
4: while unprocessed features remain do
5:  $x_i \leftarrow \text{get\_new\_feature}(X)$ 
6:  $\alpha_i \leftarrow W_i / 2^i$ 
7:  $p_i \leftarrow \text{get\_p\_value}(x_i, S_{initial})$ 
8: if  $p_i < \alpha_i$  then
9:  $S_{initial} \leftarrow S_{initial} \cup \{x_i\}$ 

```

```

10:  $W_{i+1} \leftarrow W_i + \alpha_\Delta - \alpha_i$ 
11: else
12:  $W_{i+1} \leftarrow W_i - \alpha_i$ 
13: end if
14:  $i \leftarrow i + 1$ 
15: end while
16: Train XGBoost using  $S_{\text{initial}}$  and  $Y$  with  $K$  boosting rounds.
17: Compute feature importance scores  $I_i$  for each feature
18: Set percentile threshold  $T$ 
19: Initialize  $S_{\text{refined}} \leftarrow \emptyset$ 
20: for each feature  $X_i \in S_{\text{initial}}$  do
21: if  $I_i \geq T$  then
22:  $S_{\text{refined}} \leftarrow S_{\text{refined}} \cup \{X_i\}$ 
23: end if
24: end for

```

The mathematical notation used in the AXGFS framework is summarized in Table 3, and the step-by-step procedure of the feature stream selection framework is presented in Algorithm 1.

4.1. Complexity Derivation

Let:

- F = Total number of incoming features.
- S = Number of training samples.
- R = Number of features retained after Alpha-Investing ($R \leq F$).
- B = Number of boosting rounds in XGBoost.

The overall runtime of AXGFS is derived by examining each stage independently.

Stage 1: Alpha-Investing Feature Selection

Each incoming feature is processed sequentially. For every feature, the algorithm performs feature arrival, significance-level computation (α_i), regression-based statistical testing, and wealth updating. The statistical-testing step involves fitting the regression model using the available samples and selected features. Therefore, considering R as the maximum number of retained features, the approximate cost of this stage is $O(S \times R^2)$ per incoming feature.

Therefore, across F incoming features: $T_{\text{stage1}} = O(F \times S \times R^2)$

Stage 2: XGBoost Training

Following the screening stage, XGBoost is trained on the retained feature subset of size R . The computational cost of XGBoost training is:

$$T_{\text{stage2}} = O(B \times S \times \log(S) \times R)$$

where B = Number of boosting rounds; S = number of training samples; R = number of retained features.

Stage 3: Feature Refinement

The refinement stage performs a single scan over all R retained features. For each feature:

1. Retrieve importance score
2. Compare against threshold T

Each comparison runs in $O(1)$ time. Therefore:

$$T_{\text{stage3}} = R \times O(1) = O(R)$$

Overall Complexity

Combining all three stages:

$$T_{\text{AXGFS}} = T_{\text{stage1}} + T_{\text{stage2}} + T_{\text{stage3}}$$

Substituting the derived terms:

$$\text{TAXGFS} = O(\text{FSR}^2) + O(\text{BSlog}(S)R) + O(R)$$

Accordingly, the computational complexity of the proposed AXGFS framework is primarily governed by both the Alpha Investing screening stage and the XGBoost training stage.

5. EXPERIMENTAL RESULTS AND ANALYSIS

This section discusses the experimental procedure and results.

5.1. Dataset Description

The eight different datasets, along with their sources and descriptions, used for the study are described in Table 4. These datasets are widely used by many researchers [10] to carry out experiments for feature stream selection, enabling comparison with existing methods. The link to the dataset is available in the data availability section.

Table 4. Benchmark dataset.

Sr. No	Dataset Name	Instances	Features	No. of Classes	Description	Source
1	SRBCT	63	2308	4	Small-round-blue-cell tumor (SRBCT)	KDD08 Benchmark Dataset Repository (Binghamton University)
2	LYMPHOMA	62	4026	3	Cancer	KDD08 Benchmark Dataset Repository (Binghamton University)
3	PROSTATE	102	6033	2	Cancer	KDD08 Benchmark Dataset Repository (Binghamton University)
4	LEUKEMIA	72	7129	2	Blood cancer	KDD08 Benchmark Dataset Repository (Binghamton University)
5	ARCENE	200	10,000	2	Cancer	UCI Machine Learning Repository
6	MADELON	2600	500	2	Artificial Dataset	UCI Machine Learning Repository
7	GINA	3468	970	2	Global Initiative for Asthma	OpenML Repository
8	GISETTE	7000	5000	2	Handwritten Text Recognition	UCI Machine Learning Repository

5.1.1. Data Preprocessing: The missing values

The dataset is initially preprocessed for missing values. Missing values, if found, are handled using mean imputation, and no additional normalization is performed. The features present in the dataset are then shuffled randomly.

5.2. Experimental Setup

The experiments were conducted using Python 3.10 on Google Colab. Figure 3 describes the detailed experimental procedure. For Alpha-Investing, the parameters were set to $\alpha = 0.05$ and $\alpha\Delta = 0.5$, as suggested in Zhou, et al. [8]. The XGBoost classifier was configured with a learning rate of 0.1, a maximum tree depth of 8, and 300 estimators. To analyze the robustness of the proposed framework, a threshold sensitivity analysis was performed by varying the percentile cut-off values of XGBoost feature importance scores at 25%, 40%, 50%, and 75%.

Each experiment was repeated ten times with different random feature-arrival orderings, and the mean classification accuracy with standard deviation was reported to demonstrate statistical reliability. Since streaming

feature selection is inherently order-dependent, different feature-arrival sequences may produce different feature subsets, making repeated experiments essential for reliable performance evaluation.

To evaluate the selected feature subsets, three classifiers were employed: K-Nearest Neighbor (KNN, $k = 5$) [20], Support Vector Machine (SVM with a linear kernel) [21], and Classification and Regression Tree (CART with the Gini impurity criterion) [22]. For KNN, five-fold stratified cross-validation was used. For SVM and CART, the datasets were divided into training and testing sets using an 80:20 split. For datasets containing more than 2,000 instances (MADELON, GINA, and GISETTE), LinearSVC was adopted as a computationally efficient implementation of linear SVM.

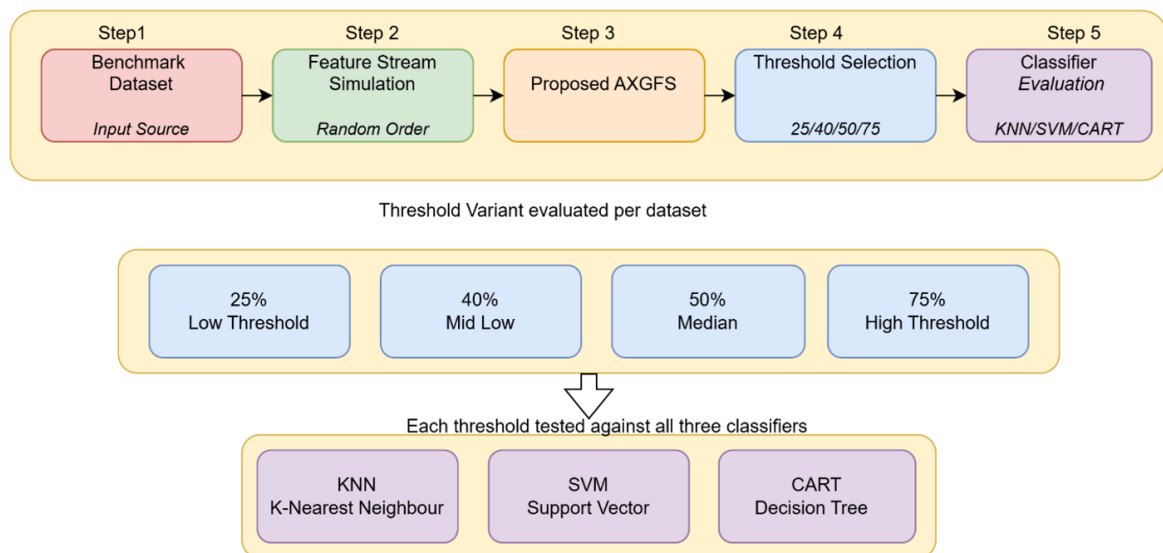


Figure 3. Experimental Procedure for the feature stream selection.

5.3. Classification Accuracy Comparison

5.3.1. KNN Classifier

Table 5 summarizes the classification accuracy obtained using the KNN classifier for different feature selection methods across eight benchmark datasets, according to the parameters in Sec. 5.2.

Table 5. Accuracy Comparison of Feature Selection Methods Using KNN (Values for alpha-Investing, Fast-OSFS, and SAOLA are reproduced from Zhou, et al. [10]).

Dataset	Alpha-Investing	Fast-OSFS	SAOLA	Proposed AXGFS with different thresholds (Mean accuracy $\pm$ standard deviation over 10 independent runs)			
				25%	40%	50%	75%
SRBCT	0.3406	0.8483	0.865	0.77 ± 0.09	0.80 ± 0.08	0.82 ± 0.09	0.84 ± 0.07
LYMPHOMA	0.625	0.9417	0.9167	0.95 ± 0.05	0.96 ± 0.04	0.97 ± 0.03	0.99 ± 0.02
PROSTATE	0.57	0.895	0.865	0.86 ± 0.06	0.89 ± 0.04	0.89 ± 0.02	0.91 ± 0.02
LEUKEMIA	0.7286	0.9357	0.9286	0.88 ± 0.07	0.91 ± 0.06	0.91 ± 0.05	0.92 ± 0.04
ARCENE	0.6625	0.6875	0.6588	0.68 ± 0.05	0.70 ± 0.05	0.71 ± 0.07	0.75 ± 0.08
MADELON	0.647	0.5626	0.5595	0.54 ± 0.02	0.57 ± 0.03	0.59 ± 0.03	0.65 ± 0.07
GINA	0.9227	0.8707	0.8067	0.88 ± 0.01	0.91 ± 0.01	0.92 ± 0.01	0.94 ± 0.01
GISETTE	0.8397	0.8665	0.8391	0.97 ± 0.00	0.97 ± 0.00	0.97 ± 0.00	0.97 ± 0.00
AVG Accuracy	0.67	0.83	0.80	0.82	0.84	0.85	0.87

AXGFS achieved better performance on most benchmark datasets. However, on SRBCT, SAOLA produced slightly higher accuracy, likely due to the dataset's limited sample size. These observations indicate that while AXGFS is generally robust, its performance may vary on a very small dataset.

5.3.2. SVM Classifier

Table 6 presents the accuracy comparison using the SVM classifier across different methods. AXGFS achieves the highest accuracy on 6 out of 8 datasets.

Table 6. Accuracy Comparison of Feature Selection Methods Using SVM (Values for alpha-Investing, Fast-OSFS, and SAOLA are reported in Zhou, et al. [10]).

Dataset	Alpha-Investing	Fast-OSFS	SAOLA	Proposed AXGFS with different thresholds (Mean accuracy $\pm$ standard deviation over 10 independent runs)			
				25%	40%	50%	75%
SRBCT	0.2608	0.888	0.9098	0.75 $\pm$ 0.15	0.81 $\pm$ 0.12	0.83 $\pm$ 0.12	0.85 $\pm$ 0.10
LYMPHOMA	0.6833	0.9083	0.9333	0.92 $\pm$ 0.06	0.92 $\pm$ 0.07	0.95 $\pm$ 0.06	0.97 $\pm$ 0.04
PROSTATE	0.59	0.885	0.83	0.82 $\pm$ 0.09	0.87 $\pm$ 0.03	0.88 $\pm$ 0.03	0.88 $\pm$ 0.02
LEUKEMIA	0.7679	0.9429	0.9321	0.87 $\pm$ 0.13	0.93 $\pm$ 0.08	0.95 $\pm$ 0.09	0.97 $\pm$ 0.06
ARCENE	0.687	0.6775	0.6413	0.66 $\pm$ 0.04	0.69 $\pm$ 0.06	0.71 $\pm$ 0.04	0.80 $\pm$ 0.05
MADOLON	0.6128	0.6117	0.6009	0.57 $\pm$ 0.01	0.57 $\pm$ 0.01	0.58 $\pm$ 0.01	0.58 $\pm$ 0.01
GINA	0.87	0.8441	0.8069	0.85 $\pm$ 0.01	0.86 $\pm$ 0.01	0.87 $\pm$ 0.01	0.87 $\pm$ 0.00
GISETTE	0.9277	0.8204	0.841	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00
AVG	0.67	0.82	0.81	0.80	0.83	0.84	0.86

The average classification accuracy shows that the proposed hybrid framework improves the performance of the SVM classifier when compared to other approaches. It is evident from the table that selecting the top 75% of the features from XGBOOST feature importance gives overall better performance. It is seen that even though AXGFS performs well in most of the datasets, SAOLA still achieves higher accuracy in the SRBCT dataset. This may be due to its small sample size. It is also observed that the low standard deviation across 10 runs indicates that AXGFS provides stable and consistent results despite variation in the feature arrival order.

5.3.3. CART Classifier

Table 7 presents the accuracy comparison using the CART classifier.

Table 7. Accuracy Comparison of Feature Selection Methods Using CART (Values for Alpha-Investing, Fast-OSFS, and SAOLA are reported in Zhou, et al. [10]).

Dataset	Alpha-Investing	Fast-OSFS	SAOLA	Proposed AXGFS with different thresholds (Mean accuracy $\pm$ standard deviation over 10 independent runs)			
				25%	40%	50%	75%
SRBCT	0.293	0.8531	0.8147	0.75 $\pm$ 0.16	0.78 $\pm$ 0.16	0.78 $\pm$ 0.19	0.76 $\pm$ 0.20
LYMPHOMA	0.6	0.8083	0.925	0.76 $\pm$ 0.14	0.78 $\pm$ 0.18	0.82 $\pm$ 0.12	0.81 $\pm$ 0.13
PROSTATE	0.59	0.89	0.855	0.79 $\pm$ 0.11	0.82 $\pm$ 0.06	0.80 $\pm$ 0.05	0.83 $\pm$ 0.06
LEUKEMIA	0.7286	0.9107	0.8786	0.87 $\pm$ 0.11	0.91 $\pm$ 0.07	0.89 $\pm$ 0.09	0.89 $\pm$ 0.08
ARCENE	0.6363	0.6575	0.6125	0.70 $\pm$ 0.07	0.69 $\pm$ 0.05	0.73 $\pm$ 0.10	0.75 $\pm$ 0.10
MADOLON	0.6276	0.5464	0.5343	0.54 $\pm$ 0.04	0.54 $\pm$ 0.03	0.56 $\pm$ 0.03	0.61 $\pm$ 0.08
GINA	0.867	0.8394	0.7858	0.86 $\pm$ 0.02	0.86 $\pm$ 0.01	0.87 $\pm$ 0.01	0.88 $\pm$ 0.00
GISETTE	0.934	0.9029	0.8862	0.95 $\pm$ 0.01	0.95 $\pm$ 0.01	0.95 $\pm$ 0.01	0.95 $\pm$ 0.01
AVG Acc.	0.66	0.80	0.79	0.78	0.79	0.80	0.81

AXGFS achieves the highest accuracy on 4 out of 8 datasets. However, its performance on datasets SRBCT, LYMPHOMA, PROSTATE, and MADOLON is lower when compared to the existing approaches. This may be due to the greedy splitting strategy of the CART classifier, which may be sensitive to variations in the selected feature subset. Consequently, feature subsets that perform well with KNN and SVM may not always yield the best classification performance with CART. Nevertheless, AXGFS achieves the highest average classification accuracy (0.81), demonstrating its overall effectiveness.

5.3.4. Cross-Classifer Analysis

Figure 4 illustrates the average accuracy across KNN, SVM, and CART classifiers across four feature selection methods. As shown in Table 8, the proposed AXGFS achieves higher average classification accuracy across the three classifiers than the existing approaches. The hybrid framework demonstrates the effectiveness of the two-stage feature selection approach, which combines Alpha-Investing streaming selection with XGBoost importance ranking.

Table 8. Accuracy comparison of cross-classifiers.

Method/Classifier	Average Accuracy across Classifiers		
	KNN	SVM	CART
Alpha-Investing	0.67	0.67	0.66
Fast-OSFS	0.83	0.82	0.80
SAOLA	0.80	0.81	0.79
Proposed AXGFS with best threshold	0.87	0.86	0.82

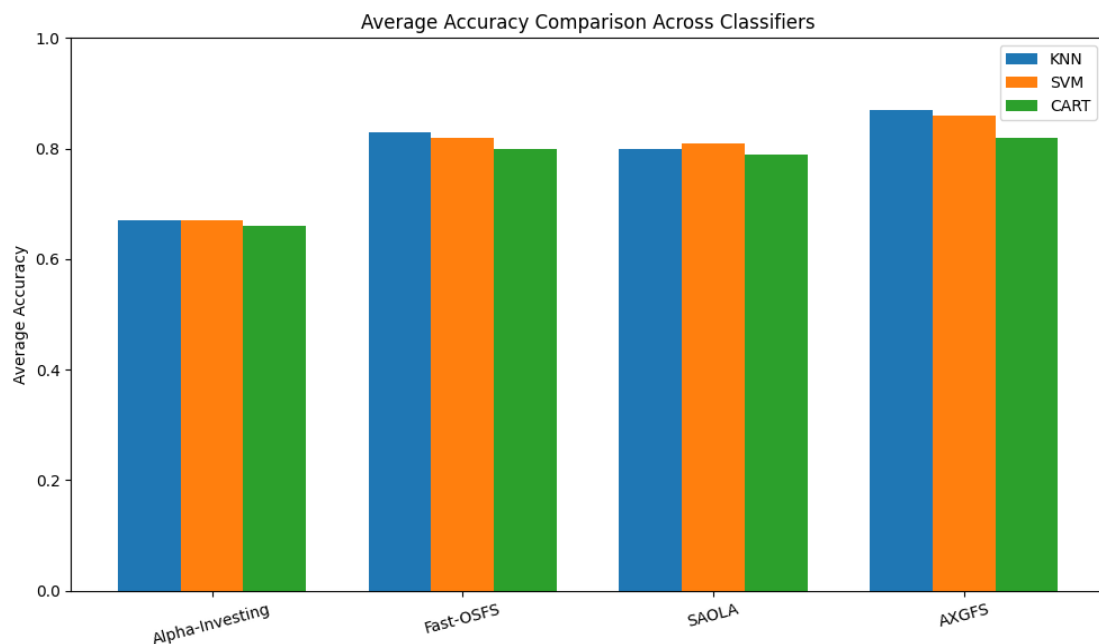


Figure 4. Average Accuracy comparison across classifiers.

5.4. Feature Selection Analysis

The threshold-wise number of features selected by the proposed AXGFS is compared with existing approaches, as presented in Table 9. As shown in Figure 5, most datasets achieve higher classification accuracy at the 75% threshold while still maintaining a significantly lower number of selected features compared to SAOLA and Alpha-Investing.

Table 9. Average No. of features selected using different feature selection methods (Values for Alpha-Investing, Fast-OSFS, and SAOLA are reported in Zhou, et al. [10]).

Dataset	Alpha-Investing	Fast-OSFS	SAOLA	Proposed AXGFS with Different Threshold			
				25%	40%	50%	75%
SRBCT	1	4	17	2	4	5	7
LYMPHOMA	1	5	22	4	7	9	14
PROSTATE	2	3	11	2	3	4	6
LEUKEMIA	3	5	21	2	3	4	6
ARCENE	6	6	17	2	3	4	7
MADOLON	6	5	7	1	2	2	3
GINA	93	22	10	21	34	43	64
GISETTE	365	13	17	101.0	161.0	202.0	299.0

For instance, on the PROSTATE dataset, AXGFS selects only 6 features compared to SAOLA's 11, while on the LEUKEMIA dataset, it selects only 6 features versus SAOLA's 21, demonstrating its effectiveness in dimensionality reduction. These results, shown in Figure 6, indicate that AXGFS not only improves classification accuracy but also selects a compact feature subset, making it suitable for high-dimensional data sets.

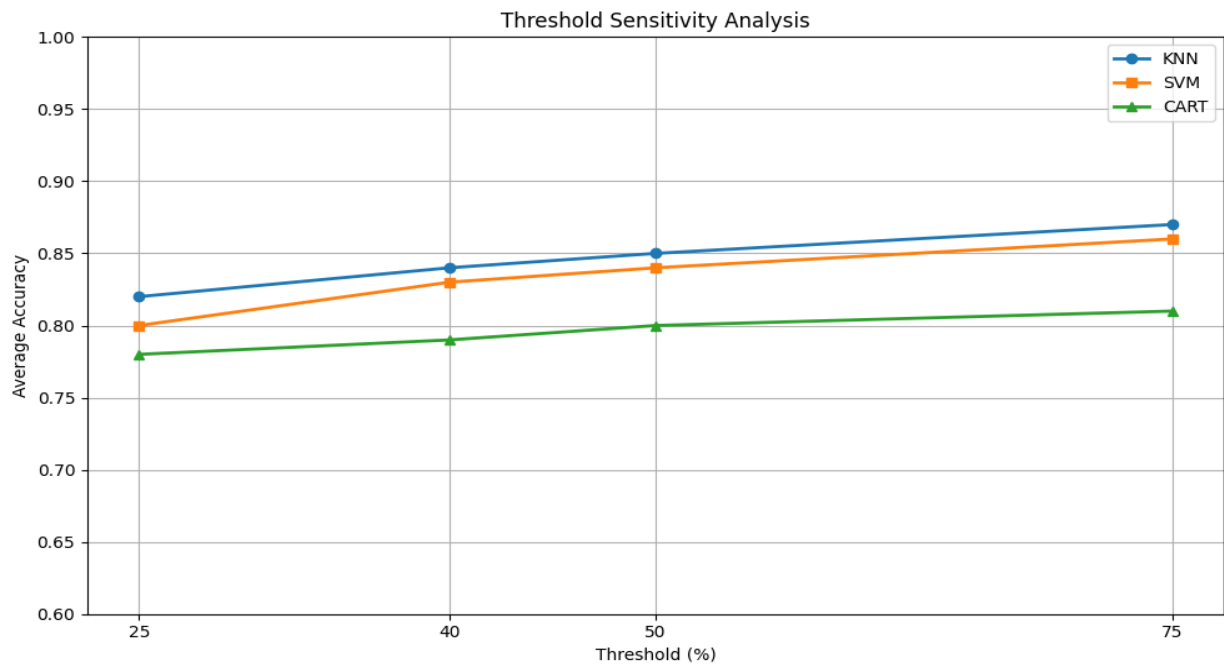


Figure 5. Threshold sensitivity analysis.

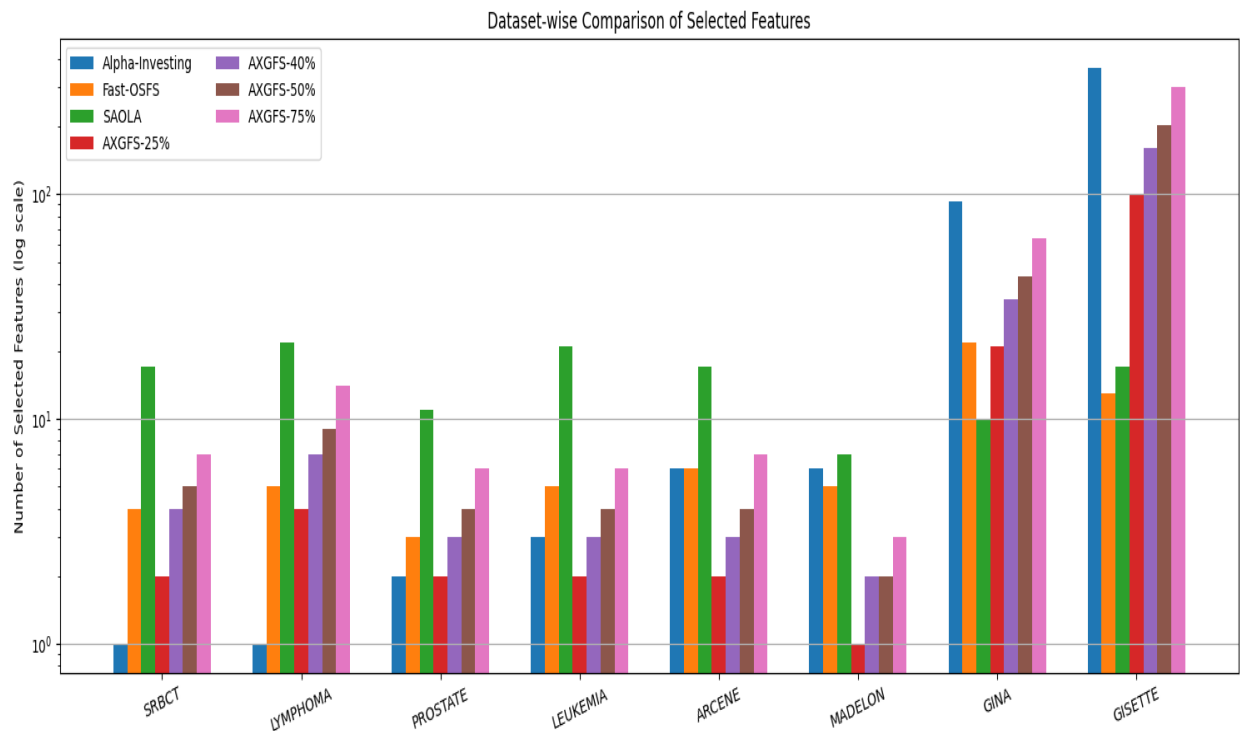


Figure 6. Dataset-wise comparison of the number of selected features.

5.5. Statistical Analysis

To perform the statistical comparison of different feature stream selection methods, the Friedman non-parametric test was performed, keeping the significance level of $\alpha = 0.05$ [23, 24]. In this study, four different algorithms are

compared, i.e., AXGFS, Fast-OSFS, SAOLA, and Alpha Investing ($K = 4$). These algorithms were evaluated on eight benchmark datasets ($N = 8$) using three different classifiers (KNN, SVM, and CART). The algorithms are ranked for each dataset according to classification accuracy. Rank 1 is assigned to the highest accuracy and rank 4 to the lowest accuracy. In cases of identical accuracies, average ranks are assigned. The Friedman test statistic is computed using the average rank of the algorithm across the eight datasets. The resulting average rank is shown in Table 10.

Table 10. Average ranks of competing methods (Lower rank = better performance. Bold indicates the best result).

Method	KNN	SVM	CART	Overall Avg.
AXGFS	1.5	1.81	1.88	1.73
Fast-OSFS	2.13	2.5	2.13	2.25
SAOLA	3.13	2.88	3	3
Alpha-Investing	3.25	2.81	3	3.02

The Friedman test results are shown in Table 11.

Table 11. Friedman test.

Classifier	N	k	df	χ^2F	p-value	Decision
KNN	8	4	3	10.05	0.018	Reject H_0
SVM	8	4	3	3.456	0.327	Fail to reject H_0
CART	8	4	3	4.95	0.175	Fail to reject H_0

The Friedman test was performed, and Table 11 summarizes the results for the KNN, SVM, and CART classifiers; it is observed that, for the KNN classifier, the Friedman test produced a test statistic of ($\chi^2F = 10.05$) with ($df = 3$) and a corresponding p-value of 0.018. It shows that statistically significant differences exist among the four feature-selection methods. The corresponding effect size, measured using Kendall's coefficient of concordance, was $W = 0.419$. However, for SVM and CART, it indicates that the observed differences in the rankings of the competing methods are not statistically significant at the 95% confidence level. Therefore, the Nemenyi [25] post-hoc test was subsequently performed only for KNN to identify pairwise differences between the algorithms [25].

The Critical Difference (CD) for the Nemenyi post-hoc test was computed using the following equation.

$$CD = q\alpha \times \sqrt{[k(k + 1) / (6N)]} \quad (1)$$

Where $q\alpha = 2.569$ for a significance level of $\alpha = 0.05$, $k = 4$ is the number of algorithms used for comparison, and $N = 8$ is the number of benchmark datasets. Substituting these values into the equation yields a Critical Difference of 1.6583.

The difference in average ranks between AXGFS and Alpha-Investing (1.75) exceeded the Critical Difference (1.6583), indicating a statistically significant difference. The remaining pairwise comparisons did not exceed the Critical Difference, indicating no statistically significant differences among those methods. Figure 7 illustrates these relationships using the Critical Difference diagram, where algorithms connected by a horizontal bar are not significantly different.

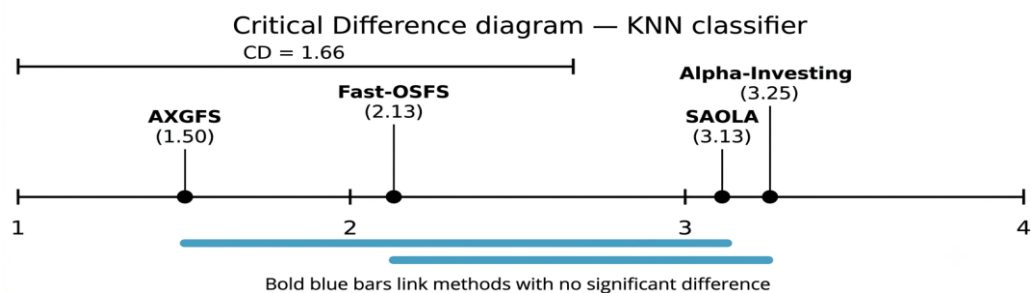


Figure 7. Critical Difference Diagram for the KNN classifier.

AXGFS achieved the best average rank among the competing methods. However, the Nemenyi post-hoc analysis indicates that its improvement is statistically significant only when compared with Alpha-Investing, whereas the differences between AXGFS and both Fast-OSFS and SAOLA were not statistically significant for the KNN classifier. These results suggest that, although AXGFS achieved the best overall average rank, Fast-OSFS and SAOLA remained competitive on specific benchmark datasets, as discussed in section 5.4.

5.6. Limitation

Although the hybrid framework gives encouraging results, it has computational overhead due to the integration of alpha investing and XGBoost. Furthermore, the experiments were conducted on benchmark datasets with simulated feature streams. Therefore, validation on real-world dynamic feature stream applications is an important direction for future work.

6. CONCLUSION

This study introduced AXGFS, a hybrid feature-selection framework designed for high-dimensional datasets and feature-stream environments. By integrating the statistical significance analysis of Alpha-Investing with the predictive capability of XGBoost, the proposed method effectively selects features that are both relevant and highly informative. Comprehensive experimental comparisons with established methods, including Alpha-Investing, Fast-OSFS, and SAOLA, demonstrate that AXGFS achieves improved classification accuracy while maintaining effective control over the number of selected features. The results further show that AXGFS provides a favorable balance between dimensionality reduction and predictive performance, making it suitable for dynamic environments where accuracy and interpretability are critical.

This study has certain limitations. First, it has higher computational time than some traditional filter-based approaches, particularly on large datasets. The framework assumes sequential arrival of features, but in real-world applications, such as health care and predictive maintenance, the features may arrive at a high rate, and the current framework has not been evaluated under such conditions.

Future work will focus on reducing execution time through parallel processing and extending the framework for real-time streaming feature selection applications, including scenarios where features arrive in groups.

Funding: This study received no specific financial support.

Institutional Review Board Statement: Not applicable.

Transparency: The authors state that the manuscript is honest, truthful, and transparent, that no key aspects of the investigation have been omitted, and that any differences from the study as planned have been clarified. This study followed all writing ethics.

Competing Interests: The authors declare that they have no competing interests.

Authors' Contributions: Both authors contributed equally to the conception and design of the study. Both authors have read and agreed to the published version of the manuscript.

Disclosure of AI Use: The author(s) used OpenAI's ChatGPT 5.5 to edit and refine the wording. All outputs were reviewed and verified by the authors.

REFERENCES

- [1] A. J. Miller, *Subset selection in regression*, 2nd ed. Boca Raton, FL, USA: Chapman & Hall/CRC, 2002.
- [2] P. Dhillon, D. Foster, and L. Ungar, "Feature selection using multiple streams," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, Chia Laguna Resort, Sardinia, Italy, vol. 9, 2010, pp. 153-160.
- [3] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238, 2005. <https://doi.org/10.1109/TPAMI.2005.159>
- [4] E. A. K. Zaman, A. Mohamed, and A. Ahmad, "Feature selection for online streaming high-dimensional data: A state-of-the-art review," *Applied Soft Computing*, vol. 127, p. 109355, 2022. <https://doi.org/10.1016/j.asoc.2022.109355>

- [5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," presented at the Advances in Neural Information Processing Systems 30 (NIPS 2017). Long Beach, California, United States, 2017.
- [6] S. Sandhiya and U. Palani, "A novel hosfs algorithm for online streaming feature selection," presented at the 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), IEEE, 2020.
- [7] S. Perkins and J. Theiler, "Online feature selection using grafting. In T. Fawcett & N. Mishra (Eds.)," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, Washington, DC, USA, 2003, pp. 592–599.
- [8] J. Zhou, D. Foster, R. Stine, and L. Ungar, "Streaming feature selection using alpha-investing," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, Chicago, IL, USA, 2005, pp. 384–393. <https://doi.org/10.1145/1081870.1081914>
- [9] 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*, vol. 11, no. 2, pp. 1–39, 2016. <https://doi.org/10.1145/2976744>
- [10] P. Zhou, S. Zhao, Y. Yan, and X. Wu, "Online scalable streaming feature selection via dynamic decision," *ACM Transactions on Knowledge Discovery from Data*, vol. 16, no. 5, pp. 1–20, 2022. <https://doi.org/10.1145/3502737>
- [11] S. Li, K. Zhang, Y. Li, S. Wang, and S. Zhang, "Online streaming feature selection based on neighborhood rough set," *Applied Soft Computing*, vol. 113, p. 108025, 2021. <https://doi.org/10.1016/j.asoc.2021.108025>
- [12] N. Almusallam, Z. Tari, J. Chan, A. Fahad, A. Alabdulatif, and M. Al-Naeem, "Towards an unsupervised feature selection method for effective dynamic features," *IEEE Access*, vol. 9, pp. 77149–77163, 2021. <https://doi.org/10.1109/ACCESS.2021.3082755>
- [13] X. Yan, A. Homaifar, M. Sarkar, B. Lartey, and K. D. Gupta, "An online unsupervised streaming features selection through dynamic feature clustering," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 5, pp. 1281–1292, 2023. <https://doi.org/10.1109/TAI.2022.3196637>
- [14] C. Villa-Blanco, C. Bielza, and P. Larrañaga, "Feature subset selection for data and feature streams: A review," *Artificial Intelligence Review*, vol. 56, no. Suppl. 1, pp. 1011–1062, 2023. <https://doi.org/10.1007/s10462-023-10546-9>
- [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, 2022. <https://doi.org/10.1016/j.aci.2019.01.001>
- [16] Y. Hochma and M. Last, "Fast online feature selection in streaming data," *Machine Learning*, vol. 114, no. 1, p. 1, 2025. <https://doi.org/10.1007/s10994-024-06712-x>
- [17] Y. Vivek, V. Ravi, and P. R. Krishna, "Online feature subset selection for mining feature streams in big data via incremental learning and evolutionary computation," *Swarm and Evolutionary Computation*, vol. 94, p. 101896, 2025. <https://doi.org/10.1016/j.swevo.2025.101896>
- [18] J. Ding, J. Du, H. Wang, and S. Xiao, "A novel two-stage feature selection method based on random forest and improved genetic algorithm for enhancing classification in machine learning," *Scientific Reports*, vol. 15, no. 1, p. 16828, 2025. <https://doi.org/10.1038/s41598-025-01761-1>
- [19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, San Francisco, CA, USA, 2016, pp. 785–794. <https://doi.org/10.1145/2939672.2939785>
- [20] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. <https://doi.org/10.1109/TIT.1967.1053964>
- [21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. <https://doi.org/10.1007/BF00994018>
- [22] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Belmont, California, United States: Wadsworth International Group, 1984.
- [23] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937. <https://doi.org/10.1080/01621459.1937.10503522>

- [24] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1-30, 2006.
- [25] P. B. Nemenyi, "Distribution-free multiple comparisons," Doctoral Dissertation, Princeton University, Princeton, New Jersey, 1963.

Views and opinions expressed in this article are the views and opinions of the author(s). Review of Computer Engineering Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.