

Adaptive Group Feature Stream Selection for Dynamic Feature Stream

Priyadarshini I.^{1*}, Shirish Sane²

¹Research Scholar, Department of Computer Engineering, MET Institute of Engineering, Affiliated to Savitribai Phule Pune University (SPPU), Nashik, India.

²Principal, Department of Computer Engineering, Gokhale Education Society, R. H. Sapat College of Engineering, Savitribai Phule Pune University (SPPU), Nashik, India.

E-mail: ^{1*} iyyanarb.phdcomp_ioe@bkc.met.edu, ² principal@ges-coengg.org

Orcid ID: ^{1*} 0009-0007-6087-3929, ² 0000-0002-4232-1051

Abstract

Feature selection is a dimensionality reduction technique that identifies relevant features while discarding irrelevant and redundant ones. Traditional feature selection methods require the entire feature space to be known in advance, making them suitable for static datasets. However, in real-world applications such as healthcare monitoring, sensor networks, and image analysis, new features are generated at different time intervals. In dynamic feature streams, the entire feature space is not known beforehand. To address this challenge, this paper proposes Adaptive Group Feature Stream Selection (AGFS), an online feature selection framework for dynamic feature streams. The proposed framework extends the Alpha-Investing algorithm from processing features individually to processing features that arrive in randomly sized groups. It employs a multi-stage processing approach. First, variance filtering removes constant features. Next, Alpha-Investing-based statistical testing selects relevant feature groups. A group-size capping mechanism is introduced to prevent larger feature groups from being discarded during statistical testing. Within each selected group, features are ranked using feature-class correlation, and redundant features are removed using correlation-based redundancy removal. A data-driven threshold, computed from feature-class correlation observed so far, is introduced to help prevent unnecessary elimination of complementary features. An adaptive wealth mechanism rewards the selection of informative features. Experiments conducted on six benchmark datasets using a K-Nearest Neighbours classifier with 5-fold cross-validation, demonstrate that the proposed framework achieves higher average classification accuracy (87.36%) while maintaining a competitive number of selected features compared to existing feature selection methods. The proposed approach is also validated across multiple classifiers and its consistency across classifiers is reported.

Keywords: Alpha-Investing, Feature Stream Selection, Dynamic Feature Stream, Adaptive Group Feature stream, Multi Stage Filtering.

1. Introduction

Feature selection is an important dimensionality reduction technique that identifies informative features while eliminating irrelevant and redundant ones. Unlike feature extraction, feature selection retains the original representation of the selected features while reducing

dimensionality. Feature selection reduces model building time and increases predictive accuracy [1]-[2].

A data stream refers to a continuous arrival of data instances with fixed features. In contrast, in a feature stream, the number of instances remains fixed and features arrive dynamically over time. The feature stream selection algorithm must decide whether to retain or discard the features as they arrive [3],[4],[5]. This work addresses feature stream selection, where features arrive dynamically in randomly sized groups.

Traditional feature selection methods such as filter, wrapper, and embedded approaches assume that the entire feature space is available before learning begins. This assumption does not hold in the feature stream environment described above, where features arrive dynamically. In such cases, traditional static feature stream selection methods are not suitable for feature stream selection.

Feature stream selection introduces two major challenges. First, the entire feature space is not known in advance. A further challenge is balancing the risk of prematurely discarding informative features when the incoming feature group size is larger than the available samples. This scenario is addressed in this work through group size capping. Existing online feature stream selection methods evaluate features individually or in groups. They typically rely on statistical testing, information theory measures, or feature weights for selecting relevant features. Some approaches also use a fixed threshold to remove irrelevant and redundant features.

The research problem addressed in this paper is how to select the optimal number of features from features arriving in different sized groups. The algorithm should identify relevant and non-redundant features and discard irrelevant and redundant features from the incoming feature group. Only the informative features are to be selected, which will improve the predictive accuracy of the classifier. To address this problem, this paper proposes Adaptive Group feature Stream Selection for features arriving in randomly sized groups. It employs an adaptive data-driven threshold to overcome the limitation of fixed relevancy threshold used by some existing approaches. In addition, a group-size capping mechanism is introduced to avoid large, randomly sized groups from being discarded during statistical testing on high-dimensional data.

1.1 Contribution

- It extends the Alpha-Investing framework's wealth-based statistical testing from single-feature streams to group-wise feature streams.
- It introduces a group-size capping mechanism that prevents large, randomly-formed feature groups from being discarded during statistical testing validated against uncapped and adaptively-sized alternatives
- It introduces a correlation-driven adaptive redundancy threshold together with an Inter-First redundancy handling strategy, for effective redundancy control.
- A variance filtering stage is incorporated prior to statistical testing to eliminate non-informative features and reduce computational overhead. The proposed framework is evaluated on six benchmark datasets and compared with existing feature selection methods in terms of classification accuracy and the number of selected features.

2. Related Work

Feature stream selection plays an important role in dimensionality reduction of high dimensional datasets. It has gained greater attention due to its capability in reducing machine learning model building time.

2.1 Supervised Online Feature Stream Selection

In supervised online feature selection, the class labels are available during learning. The Grafting technique [6] was one of the earliest online feature selection methods. It incrementally added new features into an existing model using gradient information and regularization. Even though it is effective, its requirement for knowledge of the global feature space makes it unsuitable for features arriving in continuous streams

The Alpha-Investing technique [7] introduced a statistically driven framework for feature stream selection. It processes the features one by one. Statistically significant features are retained, and insignificant features are discarded. However, it removes irrelevant features but does not remove redundant features.

To overcome this limitation, the Online Streaming Feature Selection (OSFS) method [8], combines both relevancy analysis and redundancy removal. It uses a conditional Independence test, but it has a large computational overhead for redundancy removal.

The Fast-OSFS approach [9], reduces the number of conditional independence tests, thereby accelerating redundancy checking while preserving prediction performance. Although it is computationally faster than OSFS, scalability still remains a challenge in high-dimensional datasets.

SAOLA (Scalable and Accurate Online Approach) [10] uses pairwise mutual information analysis for handling scalability issues. The issue with this approach is that it uses a predefined relevance threshold, and performance may change according to the selected threshold. Recent methods such as OFS-DD [11] use normalized mutual information to classify incoming features into selected, discarded, or delayed categories.

2.2 Unsupervised Online Feature Stream Selection

In many practical applications, the non-availability of class labels motivates the development of unsupervised online feature stream selection methods. A feature selection method [12] selects relevant features based on Pearson correlation, least square regression error, and the maximal information compression index. This method maintains a cluster of centroids instead of all features. Similarly, dynamic density-based feature clustering methods [13] were also used to select representative features.

2.3 Group Feature Stream Selection

Most early studies assume that features arrive individually, but in many real-world applications, groups of features arrive simultaneously. For instance, in medical diagnoses, multiple laboratory measurement results become available together, which has motivated the development of group feature stream selection algorithms.

The sparse group lasso method [14], originally proposed for regularized regression, has since been used as a comparison baseline in group feature stream selection studies. However, it requires all features to be known before the selection process can begin. It is difficult for the

standard batch method to operate on high-dimensional data.

Group Feature Selection with Streaming Features (GFSSF) [15], uses mutual information to identify relevant features and conditional mutual information to discard redundant features. Even though it was effective in terms of accuracy, it increased computational overhead.

Online Group Feature Selection (OGFS) [16], uses spectral analysis for intra group feature evaluation and lasso regression for inter-group redundancy removal. The approach improves feature subset quality but requires computationally intensive optimization procedures.

Group-SAOLA extends SAOLA [10], to group-arriving features by employing pairwise mutual information analysis. It has improved scalability but needs a predefined threshold and its performance depends on that threshold.

2.4 Comparative Analysis of Existing Methods

Existing single-stream feature selection algorithms, as mentioned in Table 1, have progressively improved the feature selection process. Alpha Investing provides an efficient statistical framework but lacks redundancy control, whereas OSFS and Fast OSFS incorporated relevancy analysis and redundancy control, but repeated conditional independence testing increases computational complexity. SAOLA enhances scalability but needs a predefined relevancy threshold for the feature selection process.

Group-based approaches such as GFSSF, OGFS, and Group-SAOLA are capable of feature selection in dynamic feature streams arriving in different group sizes and perform intergroup and intragroup redundancy removal. However, these methods generally involve higher computation overhead or require a predefined threshold, and their performance is affected by the threshold selection. SGLasso, in contrast, uses batch processing which needs the entire feature space to be known in advance. More recently, in the year 2026, the OS²FSG technique [17] addressed sparse streaming features through Gaussian copula-based missing-value imputation combined with online relevance and redundancy analysis, achieving strong performance across multiple domains, although its EM-based estimation introduces additional computational overhead.

2.5 Identified Research Gaps

The literature review reveals several important research gaps. Most of the existing algorithms process features individually as a single feature stream; that is, the features are processed one by one [6]-[11]. Even though some work exists for group feature stream selection, where features arrive in random-sized groups [15], [16], it has limitations. Many of the approaches rely on a predefined threshold, which was difficult to optimize for different datasets [10]. Recent survey studies further emphasize the importance of scalability, adaptability, and robustness in handling dynamically arriving features [18], [19], [20].

These research gaps motivated the development of the proposed Adaptive Group Feature Stream Selection (AGFS) framework. The proposed approach extends the existing Alpha-Investing to group-based feature streams for relevancy analysis, incorporating a group-size capping mechanism to keep significance testing statistically valid on large feature groups it integrates variance-based pre-filtering to eliminate constant features, adaptive wealth updating, and a data-driven redundancy control mechanism to carefully eliminate

non-informative features. It aims to achieve higher classification accuracy while maintaining a competitive number of selected features compared to existing feature selection methods for dynamic feature stream environments in which features arrive in randomly sized groups.

Table 1. Comparison summary of existing feature stream selection methods

Method	Relevance	Redundancy	Groups	Strength	Limitation
Grafting [6]	✓	×	×	Online model update	Requires global feature space
Alpha-Investing [7]	✓	×	×	Statistical relevance test	No redundancy control
OSFS [8]	✓	✓	×	High subset quality	Computationally expensive
Fast-OSFS [9]	✓	✓	×	Faster redundancy check	Scalability issues
SAOLA [10]	✓	✓	×	Highly scalable	Fixed relevance threshold
Group-SAOLA [10]	✓	✓	✓	Extends SAOLA to groups	Fixed relevance threshold
SGLasso [14]	✓	✓	✓	Group + within-group sparsity	High cost, high-dim data
OFS-DD [11]	✓	✓	×	Dynamic decision making	Depends on global stats
GFSSF [15]	✓	✓	✓	Group-wise feature analysis	High computational overhead
OGFS [16]	✓	✓	✓	Intra- + inter-group analysis	Computationally expensive
OS ² FSG [17]	✓	✓	×	Missing-data imputation via Gaussian copula	computational overhead from EM-based imputation

These research gaps motivated the development of the proposed Adaptive Group Feature Stream Selection (AGFS) framework. The proposed approach extends the existing Alpha-Investing to group-based feature streams for relevancy analysis, incorporating a group-size capping mechanism to keep significance testing statistically valid on large feature groups. It integrates variance-based pre-filtering to eliminate constant features, adaptive wealth updating, and a data-driven redundancy control mechanism to carefully eliminate non-informative features. It aims to achieve higher classification accuracy while maintaining a competitive number of selected features compared to existing feature selection methods for dynamic feature stream environments in which features arrive in randomly sized groups.

3. Proposed Work

Figure 1 illustrates the proposed AGFS (Adaptive Group Feature Stream Selection) methodology for selecting relevant and non-redundant features from a feature stream. It shows a multi-stage process involving variance filtering, statistical relevance testing, redundancy removal, and final validation using a classification model.

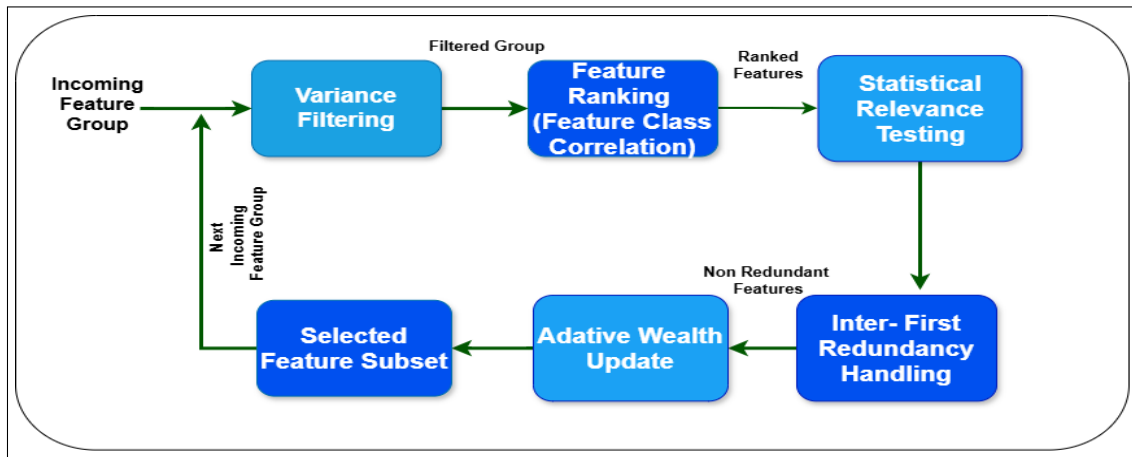


Figure 1. Adaptive Group Feature Stream Selection (AGFS)

3.1 Variance Filtering

The first stage in the proposed approach is variance filtering. The variance of the incoming features in a group is computed. If the variance is less than 0.00001, the corresponding features are eliminated. It is used to remove constant or near constant features. This stage aims for early removal of computation overhead. The variance of a feature X_j is calculated using Eq. (1)

$$var(x_j) = \frac{1}{n} \left(\sum_{i=1}^n x_{ij} - \bar{x}_j \right) \quad (1)$$

where

- x_j = the j^{th} feature
- x_{ij} = value of j^{th} feature for the i^{th} sample
- $\bar{x}_j$ = mean of the j^{th} feature
- n = Total number of samples

3.2 Statistical Relevance Testing

After variance filtering, each accepted feature in the feature group is ranked in descending order of feature-class Pearson correlation. The ranked features are then processed for statistical relevance testing. The feature groups are evaluated using the Alpha-Investing framework. Initially, the Wealth W , Reward Rate ω are set to 0.05 and 0.01. The adaptive significance threshold (α_g) is computed for each incoming feature group using Eq. (2).

$$\alpha_g = \frac{W}{(2 * group_count)} \quad (2)$$

Where $group_count$ is the number of feature groups processed so far. The statistical significance of feature group is computed using partial F-Statistic, calculated using Eq. (3)

$$F = \frac{(RSS_{old} - RSS_{new})/K}{RSS_{new}/(n - p - k - 1)} \quad (3)$$

Where

- Rss_{old} = Residual Error before adding new feature
- Rss_{new} = Residual Error after adding new feature
- n = Number of Samples
- p = Number of selected Features
- k = Group Size

The corresponding p-value is calculated. If $p\text{-value} < \alpha_g$ The feature group is considered to be statistically significant and selected for further processing.

A practical constraint affects this test: Since the F-test requires $n - P - K - 1 > 0$, oversized groups (common with high-dimensional data) render the test degenerate and are discarded outright, regardless of feature relevance. We address this by capping significance testing to the top- k_{cap} most correlated features per group, fixed at $k_{cap} = 20$ across all experiments, based on samples-per-parameter guidance from the regression literature [21][22], and validated via sensitivity analysis (Section 4.6). Redundancy removal continues to operate on the full, uncapped group. We validated this choice against two alternatives — no capping, and a dataset-adaptive cap ($k_{cap} = \max(5, n/10)$) comparing both against the fixed $k_{cap} = 20$ used throughout this study, on all datasets shown in Table 11 (Section 4.6)

3.3 Inter -First Redundancy Removal

After statistical relevance testing, the proposed framework then employs inter-first redundancy analysis, where each feature is compared with features in the previously selected subset. Consequently, intra-group redundancy is also handled automatically as every accepted feature is immediately added to the selected subset.

Pearson correlation coefficient(r) between two features X and Y is computed using Eq. (4).

$$r = \frac{\Sigma(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\Sigma(X_i - \bar{X})^2 \Sigma(Y_i - \bar{Y})^2}} \quad (4)$$

Where $\bar{X}$ and $\bar{Y}$ denote the mean of the features X and Y

To prevent aggressive feature replacement, the proposed framework introduces a correlation-driven adaptive redundancy threshold (δ). The threshold is computed online as the standard deviation of the feature-class correlations of the selected feature subset by using Eq. (5). A feature is replaced only when the improvement in feature relevance exceeds δ , thereby preserving complementary features.

$$\delta = \text{std}(\{|r(F, C)| : F \in S\}) \quad (5)$$

The redundancy removal decision is based on the following rule as per Eq. (6) - Eq. (9)

Case 1:

Discard the incoming feature F_i

$$|r(F_j, C)| > |r(F_i, C)| \quad (6)$$

and

$$|r(F_i, F_j)| \geq |r(F_i, C)| \quad (7)$$

Where F_j is an already selected feature.

Case 2:

Replace the already feature F_j with the incoming Feature F_i

$$\text{If } |r(F_i, C)| > |r(F_j, C)| + \delta \quad (8)$$

and

$$|r(F_i, F_j)| \geq |r(F_j, C)| \quad (9)$$

If neither condition is satisfied, both features are retained.

The proposed method introduces an inter-first redundancy handling strategy in feature stream selection, where incoming features are evaluated against the globally selected set before intra-group comparison. This design prevents the premature elimination of potentially useful features based solely on local group characteristics, maintains global consistency throughout the feature stream, and enables implicit intra-group redundancy removal without requiring a separate stage.

3.4 Adaptive Wealth Update

After redundancy control, the Alpha-Investing wealth is updated based on the number of features retained from the current feature group. The available wealth is updated in proportion to the incoming group size and the number of features retained. The reward rate, ω , determines how much wealth is replenished when a feature group is successfully accepted, computed as:

$$\text{reward} = \omega * \frac{\text{Number of Features Retained}}{\text{Number of Features in the full group}} \quad (10)$$

The wealth Updation after selecting the useful features

$$W_{\text{new}} = W_{\text{old}} - \alpha_g + \text{reward} \quad (11)$$

where α_g is the significance threshold spent on testing the current group, and reward is the reward earned based on the group's retention rate. Groups that fail the significance test are penalized directly by α_g without a corresponding reward. The group count is then incremented, and the significance threshold is adaptively updated for the next group as:

$$\alpha_g = \frac{W}{(2 * \text{group_count})} \quad (12)$$

This process is repeated for each incoming feature group until no more feature streams arrive, enabling adaptive learning by rewarding informative groups.

3.5 Classification-Based Validation

Feature selection was performed once per run on the full dataset. Missing values were imputed using the column median, and features standardized to zero mean and unit variance. Both imputation and standardization were applied to the full dataset prior to feature selection and classifier evaluation. Classifier performance was subsequently evaluated using standard 5-fold stratified cross-validation. KNN, SVM, Logistic Regression, Random Forest and XGBoost were employed to test performance across classifiers. Furthermore, the proposed method is compared against existing feature stream selection algorithms in terms of classification accuracy, number of features selected, and feature selection time. The algorithm and its associated notation, with parameter values summarized in Table 5, are provided below.

Algorithm :Feature Stream Selection using AGFS

Input: Feature groups $G = \{G_1, \dots, G_m\}$, labels y , W_0 , ω , τ_{var} , k_{cap}

Output: Selected feature subset S

```

1:  $W \leftarrow W_0$ ;  $S \leftarrow \emptyset$ ;  $t \leftarrow 0$ ;  $\text{RSS}_{\text{old}} \leftarrow \sum (y_i - \bar{y})^2$ 
2: for each  $G_i \in G$  do
3:    $t \leftarrow t+1$ ; if  $W \leq 0$  then break
4:    $\alpha_g \leftarrow W/(2t)$ 
5:    $A \leftarrow \{F_j \in G_i : \text{Var}(F_j) > \tau_{\text{var}}\}$ ; if  $A = \emptyset$  then continue
6:    $A \leftarrow \text{Sort } A \text{ by } |r(F_j, y)| \text{ descending}$ 
7:    $A_{\text{test}} \leftarrow \text{top-}k_{\text{cap}}(A)$  if  $|A| > k_{\text{cap}}$  else  $A$  ▷ capped, testing only
8:    $\text{RSS}_{\text{old}} \leftarrow \text{RSS}(S, y)$ ;  $\text{RSS}_{\text{new}} \leftarrow \text{RSS}(S \cup A_{\text{test}}, y)$ 
9:    $k \leftarrow |A_{\text{test}}|$ ;  $p \leftarrow |S|$ 
10:  if  $(n-p-k-1) \leq 0$  or  $\text{RSS}_{\text{new}} \geq \text{RSS}_{\text{old}}$  then  $W \leftarrow W - \alpha_g$ ; continue
11:   $F \leftarrow [(\text{RSS}_{\text{old}} - \text{RSS}_{\text{new}})/k] / [\text{RSS}_{\text{new}}/(n-p-k-1)]$ ; compute p-value
12:  if p-value  $> \alpha_g$  then  $W \leftarrow W - \alpha_g$ ; continue
13:   $n_{\text{old}} \leftarrow |S|$ 
14:   $\delta \leftarrow \text{std}(\{r(F_j, y) : F_j \in S\})$  if  $|S| > 1$  else  $\delta 0$  ▷  $\delta 0 = 0$ 
15:  for each  $F_i \in A$  (ranked order, FULL uncapped set) do ▷ inter-first
16:    discard  $\leftarrow \text{false}$ ;  $R \leftarrow \emptyset$ 
17:    for each  $F_j \in S$  do
18:      if  $r(F_j, y) > r(F_i, y)$  and  $r(F_i, F_j) \geq r(F_i, y)$  then discard  $\leftarrow \text{true}$ ; break
19:      if  $r(F_i, y) > r(F_j, y) + \delta$  and  $r(F_i, F_j) \geq r(F_j, y)$  then  $R \leftarrow R \cup \{F_j\}$ 
20:    end for
21:    if not discard then  $S \leftarrow (S \setminus R) \cup \{F_i\}$ 
22:  end for
23:   $\Delta \leftarrow |S| - n_{\text{old}}$ ; reward  $\leftarrow \omega \times (\Delta/|A|)$ ;  $W \leftarrow W - \alpha_g + \text{reward}$ 
24: end for
25: return  $S$ 

```

3.6 Computational Complexity

With the capping correction restricting significance testing to at most $k_{\text{cap}} = 20$ features per group, each partial F-test requires $O(n(p + k_{\text{cap}})^2) = O(np^2)$, since k_{cap} is a fixed constant. The inter-first redundancy analysis, however, operates on the full, uncapped group of size K , requiring $O(nKp)$ per group to compare each candidate feature against the selected subset. As the feature stream progresses, p may grow up to d , and K may also

approach d in the worst case. The overall worst-case computational complexity of AGFS is therefore $O(nd^2)$, driven primarily by the uncapped redundancy comparison rather than the significance test, where n is the number of samples and d is the total number of input features.

Table 2. Computational scalability analysis of redundancy checking

Dataset	Features (d)	Total Time for feature selection (s)	Redundancy checking Time (s)	Redundancy Checking (% of Total Time)
Madelon	500	0.164	0.007	4.3
Hiva	1617	1.908	0.515	27.0
Leukemia	7129	0.182	0.068	37.4
Lung Cancer	12533	1.875	1.144	61.0
Dexter	20000	4.652	2.407	51.7
Dorothea	100000	125.58	117.312	93.4

An additional 10-run runtime analysis was conducted to quantify the computational cost of the redundancy-checking stage. As shown in Table 2, redundancy checking accounts for a substantial proportion of the total runtime for high-dimensional datasets, reaching 61% for Lung Cancer (12,533 features), 51.7% for Dexter (20,000 features), and 93.4% for Dorothea (100,000 features). These results empirically demonstrate that redundancy checking can become a computational bottleneck when applied to full, uncapped feature groups, highlighting a scalability limitation of the current implementation and motivating the development of more efficient redundancy-checking strategies.

4. Results and Discussion

The experiments were conducted using Python 3.10 on Google Colab. The experimental parameters used are shown in Table 5. 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.

Feature selection using the proposed AGFS was performed once per run on the full dataset. Missing values were imputed using the column median, and features standardized to zero mean and unit variance, both applied to the full dataset prior to feature selection and classifier evaluation. Classifier performance was subsequently evaluated using 5-fold stratified cross-validation across five classifiers: K-Nearest Neighbour (KNN, $k = 5$), SVM, Logistic Regression, Random Forest and XGBoost .

4.1 Dataset

The benchmark dataset used for simulation of group feature stream selection is as shown in Table 3, drawn from the NIPS 2003 Feature Selection Challenge [23] and OpenML [24]. Table 4. has the additional dataset used for comparison with OS²FSG [17].

Table 3. Benchmark datasets

Sr. No	Data set	Number of Features	Number of Instances	Description
1	Madelon	500	2000	Synthetic dataset from NIPS 2003 Feature Selection Challenge).
2	Hiva	1617	4229	Drug discovery dataset for predicting compounds active against HIV.
3	Leukemia	7129	72	Microarray gene expression dataset for leukemia classification.
4	Lung-Cancer	12,533	181	Gene expression dataset for lung cancer classification.
5	Dexter	20,000	300	Sparse bag-of-words text classification dataset from the NIPS 2003 Feature Selection Challenge.
6	Dorothea	100,000	619	Drug discovery dataset for classifying compounds as active or inactive against thrombin.

Table 4. Additional dataset used for comparison with OS²FSG [17]

Sr. No	Data set	Number of Features	Number of Instances	Description
1	GLI_85	22,283	85	Gene expression dataset for glioma classification
2	RELATHE	4,322	1,427	Text categorization dataset

Table 5. Experimental parameter for the proposed AGFS

Module	Parameter	Value	Remark
Feature Grouping	No. of groups	100	Min (100, d)
	Group size	Variable	Random partitioning of d features
Variance Check	τ_{var}	1.00E-05	Removes near-constant features
Statistical Relevance	W0	0.05	Analogous to $\alpha=0.05$; fixed across datasets
	ω (Reward Rate)	0.01	Fixed across datasets
	k_cap	20	F-test only; redundancy uses full group

4.2 Accuracy Comparison

Table 6 compares accuracy across four feature stream selection methods. AGFS achieves the highest average accuracy (0.8736 ± 0.0169), outperforming group-SAOLA on 5 of 6 datasets, OGFS on all 6, and SGLasso on 4 of 6. AGFS trails on Leukemia and Madelon, with the gap is larger on Leukemia (3.7–5.4 pts), consistent with Leukemia's small sample size ($n=72$). Standard deviations (0.0011–0.0433) indicate reasonably consistent performance across 10 runs with different random seeds, with 95% confidence intervals calculated as $CI = \text{mean} \pm t(0.975, df=9) \times (SD/\sqrt{10})$.

Table 6. Accuracy comparison with other methods using the KNN classifier. (Values for Group-SAOLA, OGFS, and SGLasso are taken from [10]. Accuracy is reported on a 0–1 scale; W/T/L denotes AGFS's wins, ties, and losses against each baseline across the six datasets AGFS results are reported as mean $\pm$ SD over 10 runs.

Dataset	Group-SAOLA	OGFS	SGLasso	AGFS (Proposed)	AGFS 95% CI
Dexter	0.7947	0.5487	0.7000	0.8117 ± 0.0182	0.7987–0.8247

Lung Cancer	0.9584	0.9167	0.9300	0.9912 ± 0.0044	0.9881–0.9943
Hiva	0.9630	0.9471	0.9400	0.9671 ± 0.0011	0.9663–0.9679
Leukemia	0.9833	0.7792	1.0000	0.9463 ± 0.0247	0.9286–0.9640
Madelon	0.5317	0.4922	0.5900	0.5822 ± 0.0433	0.5512–0.6132
Dorothea	0.9183	0.8691	0.9100	0.9433 ± 0.0097	0.9364–0.9502
W/T/L	5/0/1	6/0/0	4/0/2	--	--
Average	0.8582	0.7588	0.8450	0.8736 ± 0.0169	--

4.2.1 Accuracy Comparison with Recent Method (OS²FSG)

Table 7 illustrates the accuracy comparison of the proposed AGFS with the recent method OS²FSG [17].

Table 7. Accuracy comparison with OS²FSG on shared datasets. OS²FSG results are from Yan et al. [17] (best across tested β values); AGFS: mean $\pm$ SD over 10 runs.

Dataset	OS ² FSG (Best β)[17]	AGFS(Proposed)
GLI_85	0.8824 ± 0.0644	0.9047 ± 0.0454
RELATHE	0.8424 ± 0.0106	0.8332 ± 0.0052
ALLAML	0.9581 ± 0.0343	0.9463 ± 0.0247
Madelon	0.8900 ± 0.0216	0.5822 ± 0.0433

The result demonstrates that the proposed AGFS performs better on GLI_85 and closely matches (OS²FSG) on RELATHE and ALLAML. On Madelon, the accuracy is considerably lower, due to its non-linear feature interaction, which AGFS linear relevance measure is not designed to capture.

4.3 Evaluation of Pre-processing and Feature Selection within Cross-Validation

To address concerns about pre-processing and feature selection within cross-validation, an additional nested cross-validation analysis was conducted. Unlike the standard protocol, where AGFS is applied once to the full dataset before classifier evaluation, nested CV performs pre-processing and feature selection independently within each training fold, with the resulting subset evaluated on the corresponding validation fold. Table 8 compares the two protocols to quantify the effect of fold-wise pre-processing and feature selection on reported performance. Differences between protocols were small for Madelon, Hiva, and Lung Cancer, moderate for Dorothea but larger for Leukemia and Dexter. This suggests the standard protocol can yield optimistic performance estimates on high-dimensional, small-sample datasets, whereas nested CV offers a more conservative assessment.

Table 8. Accuracy comparison between standard protocol and nested CV

Dataset	Standard Protocol	Nested CV	Percentage Point Difference
Madelon	58.22%	57.70%	0.52%
Leukemia	94.63%	87.52%	7.11%
Hiva	96.71%	96.59%	0.12%
Lung Cancer	99.12%	98.35%	0.77%
Dexter	81.17%	71.33%	9.84%
Dorothea	94.33%	92.08%	2.25%

4.4 Statistical Analysis

Table 9 shows the statistical analysis performed. One sample t-test was conducted. Comparing the AGFS 10 run distribution against each reported baseline mean. The results indicate that the proposed AGFS is significantly higher on most of the datasets. A Friedman test across all four methods confirmed a significant overall difference ($\chi^2=10.40$, $p=0.015$), and Nemenyi post-hoc analysis found a significant pairwise difference only against OGFS ($p=0.0095$). Table 9 shows the statistical analysis performed. One sample t-test was conducted. Comparing the AGFS 10 run distribution against each reported baseline mean. The results indicate that the proposed AGFS is significantly higher on most of the datasets. A Friedman test across all four methods confirmed a significant overall difference ($\chi^2=10.40$, $p=0.015$), and Nemenyi post-hoc analysis found a significant pairwise difference only against OGFS ($p=0.0095$).

Table 9. Statistical analysis

Comparison	Test	Statistic	Result Summary
AGFS vs. group-SAOLA	One-sample t-test (6 datasets)	—	5/6 significant (AGFS higher); 1/6 significant (AGFS lower, Leukemia)
AGFS vs. OGFS	One-sample t-test (6 datasets)	—	6/6 significant (AGFS higher)
AGFS vs. SGLasso	One-sample t-test (6 datasets)	—	5/6 significant (AGFS higher); 1/6 not significant (Madelon)
All 4 methods (overall)	Friedman test	$\chi^2=10.40$, $p=0.015$	Significant difference among methods
AGFS vs. OGFS (post-hoc)	Nemenyi	$p=0.0095$	Significant
AGFS vs. group-SAOLA (post-hoc)	Nemenyi	$p=0.808$	Not significant
AGFS vs. SGLasso (post-hoc)	Nemenyi	$p=0.536$	Not significant

4.5 Number of Features Selected

Table 10. Number of Features Selected -- Comparison with other methods , Values of group-SAOLA, OGFS, and SGLasso are taken from [10]. AGFS values averaged over 10 independent runs

Dataset	group-SAOLA	OGFS	SGLasso	AGFS
Madelon	2	15	5	6
Hiva	5	47	55	24
Leukemia	16	66	61	14
Lung Cancer	23	91	384	45
Dexter	21	72	56	116
Dorothea	41	126	135	120

Table 10 shows the number of features selected by the proposed AGFS framework across all six datasets. Comparing this with the accuracy results shown earlier in Table 6 reveals that on some datasets where it selects more features such as Dexter (116) and Dorothea (120), it also achieves a larger accuracy gain. Conversely, on datasets where it selects fewer features, such as Leukemia(14) and Madelon(6), the accuracy gain is smaller or slightly negative. This suggests that the additional features selected by AGFS on high dimensional dataset are contributing to relevant features rather than adding noise.

4.6 Parameter Sensitivity Results

As discussed in the methodological section, Table 11 compares accuracy under uncapped, adaptive-cap, and fixed-cap configurations. Capping significantly improves accuracy on four datasets (Dexter, Dorothea, Leukemia, Lung Cancer), with the largest gains on high-dimensional datasets (Dexter, Dorothea), though at an added computational cost, since previously-ignored large groups are now evaluated. Hiva and Madelon, having smaller average group sizes, show no considerable improvement.

Table 11. (Uncapped/adaptive/fixed, all 6 datasets) (10-Run Mean $\pm$ SD)

Dataset (sample size, cap size)	Metric	Uncapped	Adaptive Cap	Fixed Cap (k_cap=20)
Dexter (n $\approx$ 300, adaptive k_cap $\approx$ 30)	Accuracy	0.67 $\pm$ 0.05	0.79 $\pm$ 0.03	0.81 $\pm$ 0.02
	Time (s)	0.67	2.23	4.65
Dorothea (n $\approx$ 619, adaptive k_cap $\approx$ 61)	Accuracy	0.89 $\pm$ 0.01	0.94 $\pm$ 0.01	0.94 $\pm$ 0.01
	Time (s)	11.53	108.15	125.58
Leukemia (n=72, adaptive k_cap=7)	Accuracy	0.87 $\pm$ 0.03	0.95 $\pm$ 0.02	0.95 $\pm$ 0.02
	Time (s)	0.28	0.17	0.18
Lung Cancer (n=181, adaptive k_cap=18)	Accuracy	0.98 $\pm$ 0.01	0.99 $\pm$ 0.01	0.99 $\pm$ 0.00
	Time (s)	1.15	2.01	1.87
Hiva (n=4229, adaptive k_cap=422)	Accuracy	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00
	Time (s)	1.71	1.82	1.91
Madelon (n=2000, adaptive k_cap=200)	Accuracy	0.58 $\pm$ 0.04	0.58 $\pm$ 0.04	0.58 $\pm$ 0.04
	Time (s)	0.15	0.16	0.16

4.6.1 Sensitivity Analysis of k_cap, W₀, and ω and Variance Threshold:

Parameter sensitivity was evaluated for k_cap, W₀, ω , and variance threshold across accuracy, feature count, and computational cost (Figure 2, Figure 3 and Figure 4). Accuracy remained stable for W₀ and ω , with defaults (0.05, 0.01) performing at or near-best, while feature count and time rose mildly on higher-dimensional datasets. For k_cap, Leukemia's accuracy declined as the cap increased, consistent with its small sample size, whereas Hiva and Madelon showed no sensitivity. The variance threshold had no measurable effect across five orders of magnitude (1e-6 to 1e-2), confirming that near-constant and informative features are separated by a margin wider than this range. These results confirm the chosen defaults generalize well across datasets of varying scale.

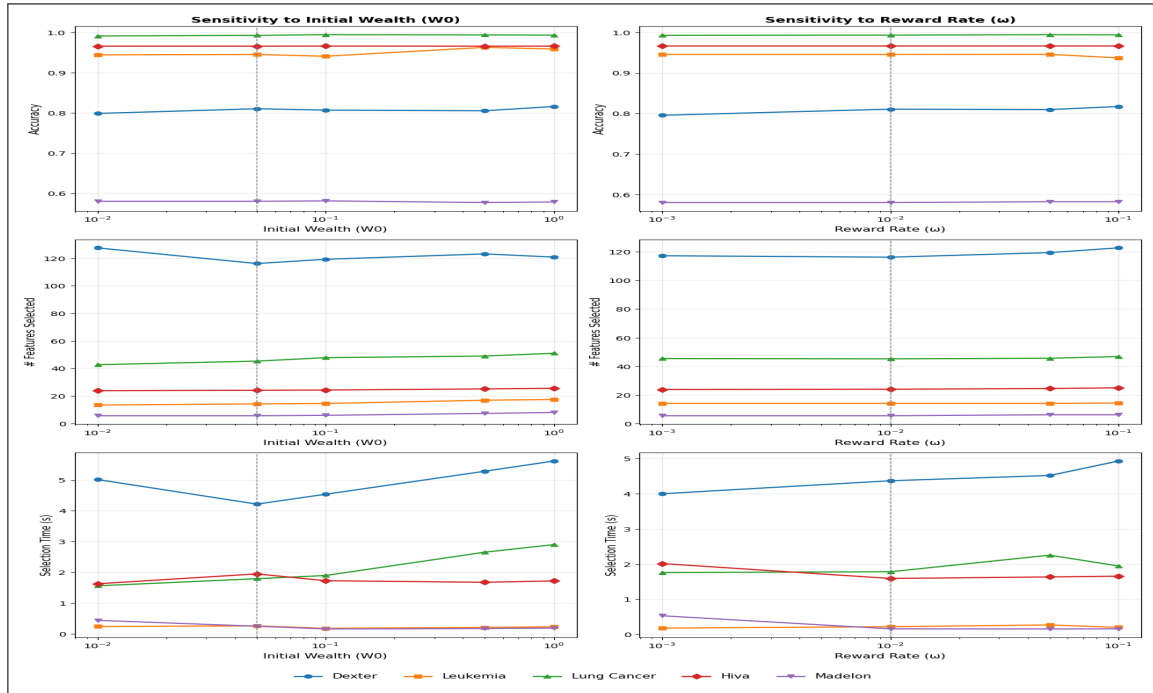


Figure 2. Sensitivity of W_0 , and ω across accuracy, feature count, and computational cost

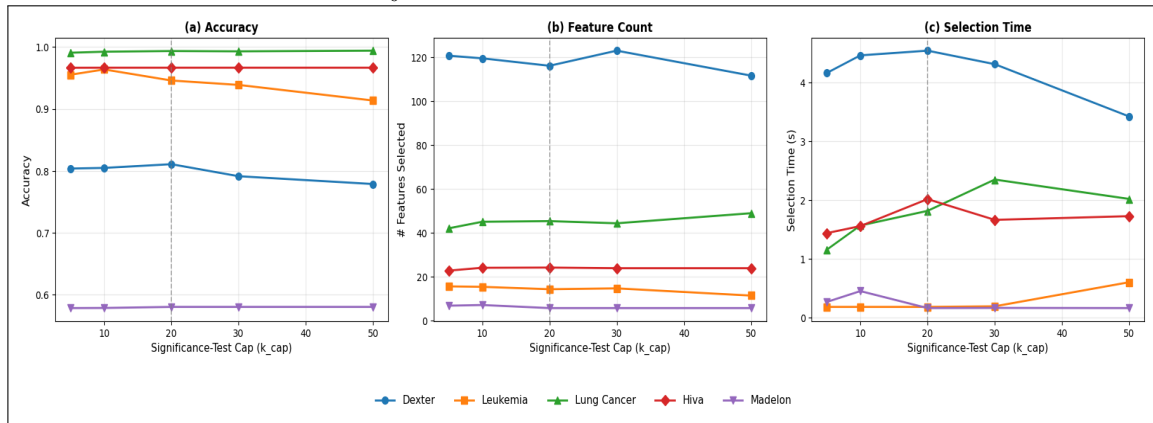


Figure 3. Sensitivity of k_{cap} across accuracy, feature count, and computational cost

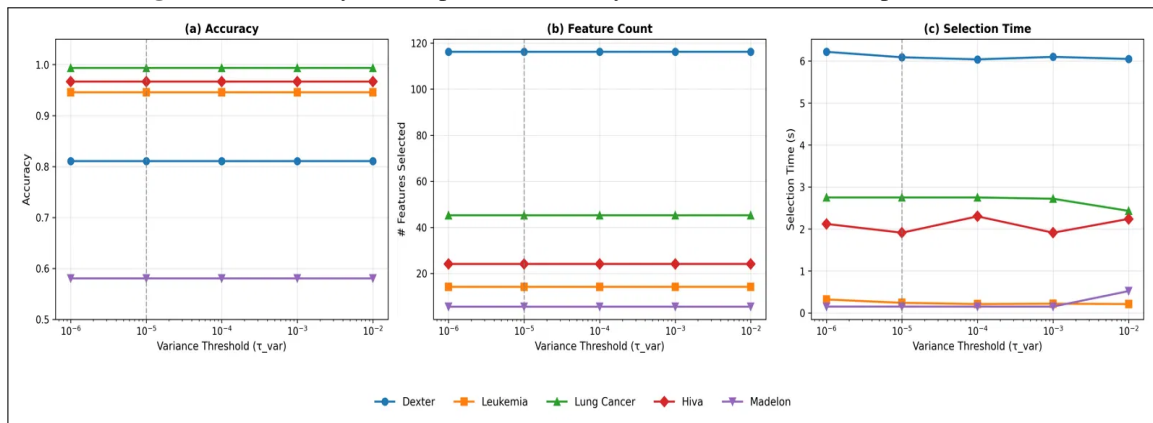


Figure 4. Sensitivity of variance threshold across accuracy, feature count, and computational cost

4.7 Individual Component Contribution

Table 12 demonstrates the leave-one-component-out ablation tested across six datasets.

It indicates that variance filtering contributed substantially on the sparse, high - dimensional datasets Dexter and Dorothea, while group size capping contributed across four datasets (Dexter, Dorothea, Leukemia, Lung Cancer) with the largest gain on high - dimensional data. The remaining components Adaptive δ , Inter-first Ordering, and Adaptive Wealth also contributed in smaller, consistent gains across the benchmark datasets.

Table 12. Leave-one-component-out ablation across six datasets (accuracy drop when component removed, %; KNN, k=5)

Component	Dexter	Leukemia	Lung Cancer	Hiva	Dorothea	Madelon
Variance Filter	16.60%	0.00%	0.00%	0.00%	1.90%	0.00%
Group-size Capping	14.50%	7.40%	1.00%	0.00%	4.90%	0.00%
Adaptive δ	-0.1%	0.80%	0.10%	0.03%	1.30%	-0.26%
Inter-first Ordering	0.60%	-0.6%	0.10%	0.02%	-0.08%	-0.13%
Adaptive Wealth	0.10%	0.40%	0.00%	0.00%	0.40%	-0.11%

4.8 Validation Across Multiple Classifiers

Table 13 shows validation across multiple classifiers for all datasets for 10 independent runs. Logistic regression achieves the highest accuracy on 5 out of 6 datasets. It also shows that AGFS selected features generalize well across classifiers.

Table 13. Accuracy comparison across different classifiers (10-Run Mean $\pm$ SD)

Dataset	KNN	SVM	Random Forest	Logistic Regression	XGBoost
Leukemia	0.95 $\pm$ 0.02	0.97 $\pm$ 0.03	0.95 $\pm$ 0.03	0.97 $\pm$ 0.02	0.91 $\pm$ 0.03
Madelon	0.58 $\pm$ 0.04	0.62 $\pm$ 0.02	0.60 $\pm$ 0.04	0.61 $\pm$ 0.00	0.59 $\pm$ 0.05
Lung Cancer	0.99 $\pm$ 0.00	0.99 $\pm$ 0.00	0.99 $\pm$ 0.00	1.00 $\pm$ 0.00	0.98 $\pm$ 0.01
Hiva	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00	0.97 $\pm$ 0.00
Dexter	0.81 $\pm$ 0.02	0.82 $\pm$ 0.03	0.83 $\pm$ 0.03	0.85 $\pm$ 0.02	0.78 $\pm$ 0.03
Dorothea	0.94 $\pm$ 0.01	0.97 $\pm$ 0.01	0.96 $\pm$ 0.00	0.97 $\pm$ 0.00	0.95 $\pm$ 0.00

4.9 Detailed Metric per Classifier

Tables 14, 15, 16, 17 and 18 report classifier performance (KNN, SVM, Random Forest, Logistic Regression, XGBoost) across six datasets (10-Run Mean $\pm$ SD). Precision and Recall are closely matched on Leukemia, Lung Cancer, and Dexter, but on Hiva, Recall falls well below Precision (0.56–0.58 vs. 0.79–0.83) alongside low ROC-AUC and MCC, reflecting severe class imbalance ($\sim$ 96.5%/3.5%). Dorothea shows a milder version of this gap. Logistic Regression generally achieves the best-Balanced Accuracy and MCC, while XGBoost tends to underperform, particularly on Madelon and Dorothea.

Table 14. Classification performance of KNN (k=5) using AGFS-selected features

Dataset	Precision	Recall	F1	ROC-AUC	MCC	Bal. Acc.
Leukemia	0.96 $\pm$ 0.02	0.93 $\pm$ 0.04	0.94 $\pm$ 0.03	0.99 $\pm$ 0.01	0.88 $\pm$ 0.05	0.93 $\pm$ 0.04
Madelon	0.58 $\pm$ 0.04	0.58 $\pm$ 0.04	0.58 $\pm$ 0.04	0.61 $\pm$ 0.06	0.16 $\pm$ 0.08	0.58 $\pm$ 0.04
Lung Cancer	1.00 $\pm$ 0.00	0.98 $\pm$ 0.01	0.99 $\pm$ 0.01	0.99 $\pm$ 0.01	0.98 $\pm$ 0.01	0.98 $\pm$ 0.01
Hiva	0.81 $\pm$ 0.03	0.57 $\pm$ 0.01	0.60 $\pm$ 0.01	0.68 $\pm$ 0.02	0.28 $\pm$ 0.02	0.57 $\pm$ 0.01

Dexter	0.83±0.03	0.81±0.02	0.81±0.02	0.88±0.03	0.64±0.05	0.81±0.02
Dorothea	0.95±0.01	0.74±0.03	0.80±0.03	0.93±0.03	0.66±0.04	0.74±0.03

Table 15. Classification performance of SVM using AGFS-selected features

Dataset	Precision	Recall	F1	ROC-AUC	MCC	Bal. Acc.
Leukemia	0.97±0.03	0.96±0.04	0.96±0.03	0.99±0.01	0.93±0.06	0.96±0.04
Madelon	0.63±0.02	0.62±0.02	0.62±0.02	0.66±0.04	0.25±0.05	0.62±0.02
Lung Cancer	0.98±0.01	0.99±0.01	0.98±0.01	1.00±0.00	0.96±0.01	0.99±0.01
Hiva	0.83±0.04	0.56±0.02	0.59±0.02	0.61±0.04	0.27±0.05	0.56±0.02
Dexter	0.84±0.03	0.82±0.03	0.82±0.03	0.92±0.02	0.66±0.05	0.82±0.03
Dorothea	0.94±0.02	0.91±0.05	0.91±0.04	0.99±0.00	0.84±0.07	0.91±0.05

Table 16. Classification performance of random forest using AGFS-selected features

Dataset	Precision	Recall	F1	ROC-AUC	MCC	Bal. Acc.
Leukemia	0.96±0.03	0.93±0.04	0.94±0.04	0.99±0.01	0.88±0.06	0.93±0.04
Madelon	0.60±0.04	0.60±0.04	0.60±0.04	0.64±0.06	0.21±0.08	0.60±0.04
Lung Cancer	1.00±0.00	0.98±0.01	0.99±0.00	1.00±0.00	0.98±0.01	0.98±0.01
Hiva	0.80±0.03	0.58±0.02	0.62±0.03	0.74±0.02	0.31±0.05	0.58±0.02
Dexter	0.84±0.03	0.83±0.03	0.83±0.03	0.91±0.03	0.67±0.06	0.83±0.03
Dorothea	0.92±0.01	0.88±0.02	0.90±0.01	0.99±0.01	0.80±0.03	0.88±0.02

Table 17. Classification performance of logistic regression using AGFS-selected features

Dataset	Precision	Recall	F1	ROC-AUC	MCC	Bal. Acc.
Leukemia	0.97±0.02	0.96±0.02	0.97±0.02	1.00±0.00	0.94±0.04	0.96±0.02
Madelon	0.61±0.00	0.61±0.00	0.61±0.01	0.64±0.01	0.23±0.01	0.61±0.00
Lung Cancer	1.00±0.00	0.99±0.01	0.99±0.01	1.00±0.00	0.98±0.01	0.99±0.01
Hiva	0.83±0.05	0.57±0.01	0.61±0.01	0.77±0.02	0.30±0.03	0.57±0.01
Dexter	0.86±0.02	0.85±0.02	0.85±0.02	0.93±0.02	0.72±0.04	0.85±0.02
Dorothea	0.96±0.01	0.91±0.02	0.93±0.01	0.99±0.00	0.86±0.02	0.91±0.02

Table 18. Classification performance of XGBoost using AGFS-selected features

Dataset	Precision	Recall	F1	ROC-AUC	MCC	Bal. Acc.
Leukemia	0.92±0.02	0.89±0.04	0.90±0.04	0.98±0.01	0.81±0.06	0.89±0.04
Madelon	0.59±0.05	0.59±0.05	0.59±0.05	0.62±0.07	0.17±0.10	0.59±0.05
Lung Cancer	0.98±0.02	0.94±0.02	0.95±0.01	1.00±0.00	0.92±0.03	0.94±0.02
Hiva	0.79±0.03	0.56±0.01	0.59±0.02	0.74±0.02	0.25±0.04	0.56±0.01
Dexter	0.80±0.03	0.78±0.03	0.78±0.04	0.87±0.04	0.58±0.07	0.78±0.03
Dorothea	0.91±0.01	0.83±0.01	0.86±0.01	0.94±0.02	0.73±0.02	0.83±0.01

4.10 Feature Selection Stability

Table 19. Feature stability using Jaccard and Kuncheva stability index

Dataset	d (Total Features)	n (Samples)	Avg. Features Selected	Mean Jaccard	Mean Kuncheva Index
Leukemia	7,129	72	14.2	0.01±0.02	0.01±0.03
Madelon	500	2,000	5.6	0.07±0.08	0.12±0.13
Lung Cancer	12,533	181	45.3	0.05±0.03	0.09±0.05
Hiva	1,617	4,229	24.1	0.20±0.06	0.32±0.09
Dexter	20,000	300	116.2	0.01±0.01	0.02±0.02
Dorothea	100,000	619	119.6	0.16±0.04	0.28±0.06

Table 19 shows pairwise Jaccard and Kuncheva stability indices across the 45 run combinations, revealing considerable variation in selected feature subsets, likely due to random feature ordering and group size. However, Table 13 shows consistent predictive performance across runs and classifiers, suggesting different subsets carry comparable predictive information. Stability is thus interpreted alongside performance and other evaluation measures, not as a standalone indicator of selection quality.

4.11 Computational Cost, Scalability and Processing Throughput Across Datasets:

Table 20 reports feature reduction ratio, selection time, peak memory, and processing throughput. Memory usage scales with overall dataset size (samples × features) rather than dimensionality alone — Hiva requires more memory than the higher-dimensional Dexter, reflecting its larger sample size (n=4,229 vs. n=300). Throughput shows the same pattern: Hiva's 847 features/s is lower than Dexter's 4,301 features/s, despite Dexter having far more features. Feature reduction remains efficient ($\geq 98.5\%$) across all datasets.

Table 20. Computational cost scalability and processing throughput across datasets

Dataset	Avg. Features selected	Reduction Ratio	Selection Time	Peak Memory	processing throughput (features/s)
Leukemia	14.2	99.80%	0.18s	11.99 MB	39,606
Madelon	5.6	98.88%	0.16s	22.98 MB	3,125
Lung Cancer	45.3	99.64%	1.87s	52.28 MB	6,702
Hiva	24.1	98.51%	1.91s	156.65 MB	847
Dexter	116.2	99.42%	4.65s	137.85 MB	4,301
Dorothea	119.6	99.88%	125.58s	1419.15 MB	796

4.12 Running Time Comparison with Existing Methods

Table 21. Running time comparison with other methods (values of group-SAOLA, OGFS, and SGLasso are taken from [10])

Dataset	Group-SAOLA	OGFS	SGLasso	AGFS (Proposed)
Madelon	3	1	18	0.16
Hiva	0.1	0.1	27	1.91
Leukemia	9	2	1752	0.18
Lung Cancer	2	1	734	1.87
Dexter	2	4	69	4.65
Dorothea	23	22	112	125.58

Table 21 compares running times across methods. AGFS was fastest on Madelon and Leukemia, and generally faster than SGLasso except on Dorothea. On Hiva, it was slower than group-SAOLA and OGFS. On Dorothea, however, AGFS was slowest (125.58s), reflecting its capping mechanism, which enables evaluation of large feature groups that other methods discard outright.

4.13 Extended Evaluation on a Real-World, Temporally-Ordered Feature Stream (IoT Cybersecurity Data)

To address the concern that randomly partitioning static datasets does not demonstrate performance under genuine online feature arrival conditions, AGFS was additionally evaluated on the real-world N-BaIoT dataset [25]. N-BaIoT contains 115 network-traffic features derived from five temporal observation windows (100 ms, 500 ms, 1.5 s, 10 s, and 1 min), with 23 features associated with each window. A near-balanced subset of 9,982 instances (4,992 benign and 4,990 attack) was obtained through stratified sampling using a fixed seed (42). Unlike the random feature-stream construction used in the benchmark experiments, the temporal groups

Table 22 shows accuracy increasing from 0.9952 to 0.9983 as temporal feature groups were incorporated sequentially, with 17 features retained — comparable to the 0.9986 accuracy under standard random ordering. Leave-one-device-out validation achieved 99.84% confirming generalization to unseen devices. However, because N-BaIoT is a pre-collected benchmark dataset, the experiment represents an application-derived temporal configuration rather than a live deployment with real-time feature generation. Thus, the results provide evidence of AGFS performance under temporally ordered feature arrival while leaving evaluation under fully live streaming conditions for future work.

Table 22. Temporally-ordered feature stream

Group	Temporal Window	Feature Indices	Count	Cumulative Retained Features	Accuracy
G1	100 ms	X[0], X[1], X[2], X[15]–X[17], X[30]–X[36], X[65]–X[67], X[80]–X[86]	23	5	0.9952
G2	500 ms	X[3]–X[5], X[18]–X[20], X[37]–X[43], X[68]–X[70], X[87]–X[93]	23	8	0.9955
G3	1.5 s	X[6]–X[8], X[21]–X[23], X[44]–X[50], X[71]–X[73], X[94]–X[100]	23	12	0.9959
G4	10 s	X[9]–X[11], X[24]–X[26], X[51]–X[57], X[74]–X[76], X[101]–X[107]	23	15	0.998
G5	1 min	X[12]–X[14], X[27]–X[29], X[58]–X[64], X[77]–X[79], X[108]–X[114]	23	17	0.9983

4.14 Robustness Analysis under Noisy, Incomplete, Imbalanced, and Dynamic Feature Streams

Table 23 summarizes the robustness of AGFS under practical feature-stream conditions. AGFS maintains stable predictive performance in the presence of noisy features and degrades gradually as missingness increases. Under class imbalance, Balanced Accuracy and MCC

provide more informative measures than raw accuracy. AGFS also remains functional across different feature-group sizes, achieving accuracies of 0.9140, 0.9310, and 0.8997 for small, large, and unequal groups, respectively. Overall, these results indicate practical robustness across several feature-stream challenges, while explicit adaptation to concept drift remains a direction for future work.

Table 23. Robustness analysis against practical challenges

Practical Challenge	Dataset	Stress Condition	Observed Result	Interpretation
Noisy features	Leukemia	Random noise: 0%, 25%, 50%, 75%, 100% of original feature count	Accuracy: 0.9343–0.9484; average noise features selected: 0 (at 0%) to 5.7 (at 100%)	Stable performance under substantial noise
Missing values	Leukemia	MCAR: 5–50% missing cells with median imputation	Accuracy: decreased from 0.9472 to 0.8348 across 5–50% missingness	Resilient to moderate missingness, with gradual degradation at higher missingness
Class imbalance	Hiva / Dorothea	Class ratios: 96.5/3.5% and 89/11%	Balanced Accuracy — Hiva: 56–58%; Dorothea: 74–91%	Demonstrates performance under severe class imbalance using an appropriate metric
Variable feature-group sizes	Leukemia	Equal-small: 10; equal-large: 50; unequal: 10–30–15–40–25...	Accuracy: 0.9140, 0.9310, and 0.8997, respectively	Performs better at equal sized group
Concept drift	—	Distributional change over time	Not evaluated	Concept drift handling remains an open direction for future work

5. Conclusion

This study proposes an adaptive group-based feature selection framework for feature streams where features arrive dynamically in randomly sized groups. The framework extends the existing Alpha-Investing framework from single-feature to group-wise streams through an adaptive wealth-update mechanism. Key contributions include a capping mechanism in relevance checking, addressing the rejection of larger groups before testing, and an Inter-First redundancy handling strategy, in which intra-group redundancy is handled implicitly. Although the framework improves average classification accuracy over existing methods on the evaluated benchmark datasets, it has some limitations. First, Pearson correlation is used for relevancy and redundancy measurement, capturing only linear dependencies. Second, the significance-test cap is fixed at $k_{\text{cap}} = 20$ through sensitivity analysis; while this performed well across tested datasets, its generalizability to substantially different datasets remains unconfirmed. Third, the method was primarily evaluated on simulated streams via random partitioning; a supplementary N-BaIoT experiment using genuine temporal feature structure showed comparable performance, although its applicability under real-time conditions remains a direction for future work. Fourth, redundancy checking operates on the full, uncapped group to avoid discarding useful features; empirical analysis confirmed this dominates computational cost at high dimensionality, exceeding 90% of runtime on the largest dataset tested. Future

work will extend AGFS toward real-time processing through nonlinear dependency measures, more efficient redundancy-checking, parallel processing for scalability, and concept drift handling.

Conflict of Interest

The authors declare that they have no conflict of interest.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Ethical Statement

This study did not involve human participants, human data, or animal subjects. All datasets used in this study are publicly available benchmark datasets obtained from established repositories.

Acknowledgement

The authors would like to thank the Department of Computer Engineering, MET Institute of Engineering, Savitribai Phule Pune University (SPPU), for their support during this research.

References

- [1] Dhillon, Paramveer, Dean Foster, and Lyle Ungar. "Feature Selection Using Multiple Streams." In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, 153-160.
- [2] AlNuaimi, Noura, Mohammad Mehedy Masud, Mohamed Adel Serhani, and Nazar Zaki. "Streaming Feature Selection Algorithms for Big Data: A Survey." Applied Computing and Informatics 18, no. 1-2 (2022): 113-135.
- [3] Peng, Hanchuan, Fuhui Long, and Chris Ding. "Feature Selection Based on Mutual Information Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy." IEEE Transactions on pattern analysis and machine intelligence 27, no. 8 (2005): 1226-1238.
- [4] Villa-Blanco, Carlos, Concha Bielza, and Pedro Larrañaga. "Feature Subset Selection for Data and Feature Streams: A Review: C. Villa-Blanco et al." Artificial Intelligence Review 56, no. Suppl 1 (2023): 1011-1062.
- [5] Zaman, Ezzatul Akmal Kamaru, Azlinah Mohamed, and Azlin Ahmad. "Feature Selection for Online Streaming High-Dimensional Data: A State-of-the-Art Review." Applied Soft Computing 127 (2022): 109355.
- [6] Perkins, Simon, and James Theiler. "Online Feature Selection Using Grafting." In Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003, 592-599.
- [7] Zhou, Jing, Dean Foster, Robert Stine, and Lyle Ungar. "Streaming Feature Selection Using Alpha-Investing." In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, 2005, 384-393.
- [8] Wu, Xindong, Kui Yu, Hao Wang, and Wei Ding. "Online Streaming Feature Selection."

- In Proceedings of the 27th international conference on machine learning (ICML-10), 2010, 1159-1166.
- [9] Wu, Xindong, Kui Yu, Wei Ding, Hao Wang, and Xingquan Zhu. "Online Feature Selection with Streaming Features." *IEEE transactions on pattern analysis and machine intelligence* 35, no. 5 (2012): 1178-1192.
 - [10] Yu, Kui, Xindong Wu, Wei Ding, and Jian Pei. "Scalable and Accurate Online Feature Selection for Big Data." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11, no. 2 (2016): 1-39.
 - [11] Zhou, Peng, Shu Zhao, Yuanting Yan, and Xindong Wu. "Online Scalable Streaming Feature Selection via Dynamic Decision." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, no. 5 (2022): 1-20.
 - [12] Almusallam, Naif, Zahir Tari, Jeffrey Chan, Adil Fahad, Abdulatif Alabdulatif, and Mohammed Al-Naeem. "Towards an Unsupervised Feature Selection Method for Effective Dynamic Features." *IEEE Access* 9 (2021): 77149-77163.
 - [13] Yan, Xuyang, Abdollah Homaifar, Mrinmoy Sarkar, Benjamin Lartey, and Kishor Datta Gupta. "An Online Unsupervised Streaming Features Selection Through Dynamic Feature Clustering." *IEEE Transactions on Artificial Intelligence* 4, no. 5 (2022): 1281-1292.
 - [14] Simon, Noah, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "A Sparse-Group Lasso." *Journal of computational and graphical statistics* 22, no. 2 (2013): 231-245.
 - [15] Li, Haiguang, Xindong Wu, Zhao Li, and Wei Ding. "Group Feature Selection with Streaming Features." In *2013 IEEE 13th International Conference on Data Mining, IEEE, 2013*, 1109-1114.
 - [16] Wang, Jing, Meng Wang, Peipei Li, Luoqi Liu, Zhongqiu Zhao, Xuegang Hu, and Xindong Wu. "Online Feature Selection with Group Structure Analysis." *IEEE Transactions on Knowledge and Data Engineering* 27, no. 11 (2015): 3029-3041.
 - [17] Yan, Huyong, Feilong Chen, Jia Chen, and Zhao Li. "OS2FSG: Online Sparse Streaming Feature Selection via Gaussian Copula." *Neurocomputing* (2026): 134273.
 - [18] Hochma, Yael, and Mark Last. "Fast Online Feature Selection in Streaming Data." *Machine Learning* 114, no. 1 (2025): 1.
 - [19] Vivek, Yelleti, Vadlamani Ravi, and P. Radha Krishna. "Online Feature Subset Selection for Mining Feature Streams in Big Data via Incremental Learning and Evolutionary Computation." *Swarm and Evolutionary Computation* 94 (2025): 101896.
 - [20] Duan, Zhangling, Tianci Li, Zhaolong Ling, Xingyu Wu, Jingye Yang, and Zhaohong Jia. "Fair streaming feature selection." *Neurocomputing* 624 (2025): 129394.
 - [21] Peduzzi, Peter, John Concato, Elizabeth Kemper, Theodore R. Holford, and Alvan R. Feinstein. "A Simulation Study of the Number of Events Per Variable in Logistic Regression Analysis." *Journal of clinical epidemiology* 49, no. 12 (1996): 1373-1379.
 - [22] Vittinghoff, Eric, and Charles E. McCulloch. "Relaxing the Rule of Ten Events Per Variable in Logistic and Cox Regression." *American journal of epidemiology* 165, no. 6 (2007): 710-718.

- [23] Guyon, Isabelle, Steve Gunn, Asa Ben-Hur, and Gideon Dror. "Result Analysis of the Nips 2003 Feature Selection Challenge." *Advances in neural information processing systems* 17 (2004).
- [24] OpenML. "Hiva, Leukemia, and Lung Cancer Datasets." OpenML. Hiva, Dataset ID 1039; Leukemia, Dataset ID 1104; Lung Cancer, Dataset ID 45093.
- [25] Meidan, Yair, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. "N-baiot—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders." *IEEE Pervasive Computing* 17, no. 3 (2018): 12-22.