Streaming Feauture Selection

Dilbar Isakova, Linhan Wang
Erasmus Mundus Joint Master's Degree in Big Data Management and Analytics
Universitat Politècnica de Catalunya, Barcelona, Spain
Università di Padova, Padova, Italy
dilbar.isakova@estudiantat.upc.edu, linhan.wang@estudiantat.upc.edu

***Abstract*— Knowledge discovery for data streaming requires online feature selection to reduce the complexity of real-world datasets and significantly improve the learning process. This paper presents a comprehensive survey of feature selection (FS) algorithms for both static and dynamic environments, providing a detailed taxonomy that categorizes these methods based on search strategy, evaluation process, and feature structure. The study covers traditional and online FS methods, offering qualitative and quantitative analyses of their strengths and weaknesses. The survey identifies several data forms, including group stream, multi-label, capricious, imbalance, and feature drift, and evaluates FS methods based on criteria such as accuracy, precision, recall and F1-score. An experimental study compares prominent FS methods using various benchmark datasets, demonstrating their performance in various scenarios. This survey aims to enhance the efficiency of learning state-of-the-art FS methods, identify limitations and research gaps, and inspire future research directions. The study concludes with observations and open issues in FS, emphasizing the need for continued exploration and development in this field. The findings and proposed taxonomy provide a crucial tool for researchers and practitioners to select appropriate algorithms tailored to their specific data challenges, ensuring optimal feature selection and consequent model performance.**

Keywords: Streaming Feature Selection, Big Data, Dimensionality Reduction, Traditional Feature Selection, Taxonomy, Online feature Selection

I. Introduction

The burgeoning field of data science continually encounters the challenge of efficiently handling and processing high-dimensional streaming data. This research paper delves into the domain of streaming feature selection (SFS), an advanced technique essential for extracting valuable insights from such data in real-time. SFS facilitates the reduction of data dimensionality by dynamically selecting the most relevant features, thus enabling the application of machine learning algorithms more efficiently and effectively [1]. This paper explores the various methods of SFS, including online individual and group feature selection, and contrasts these with traditional non-streaming methods to underscore their unique applicability in handling streaming data scenarios.

Our examination is grounded in a critical analysis of foundational studies in the field which provides a comprehensive review of the methodologies and challenges associated with SFS. Previous studies [2] offer insights into both the evolution of feature selection techniques and the current state-of-the-art approaches, including Alpha-Investing, OSFS (Online Streaming Feature Selection), SAOLA, and Fast-OSFS, each designed to address the complexities of streaming data. The significance of SFS extends beyond mere data reduction; it plays a pivotal role in enhancing computational efficiency, minimizing storage requirements, and improving the timeliness and accuracy of data-driven decisions [1].

Structured around a detailed taxonomy shown in Figure 1, this paper segments the discussion into various facets of SFS, encompassing dataset characteristics, evaluation metrics, and the specific methods employed in streaming contexts. Furthermore, it highlights the pressing challenges such as feature drift, class imbalance, and scalability — issues that are paramount in real-time data environments. By integrating these elements, the research aims to not only chart a detailed landscape of streaming feature selection but also to project future directions in research and application, thus contributing to the ongoing advancement of big data analytics.

A. Dataset Description

The experimental evaluation detailed in [1], utilizes 15 benchmark datasets from two major repositories: the University of California, Irvine (UCI) Machine Learning Repository [3] and the Arizona State University (ASU) Feature Selection Repository [4]. These datasets are specifically chosen to test and compare the performance of various feature selection methods. By applying different feature selection techniques to these datasets, the study assesses key performance metrics such as accuracy, precision, recall, and F-measures. This approach allows for a comprehensive evaluation of how well each method performs across diverse data scenarios, highlighting their efficiency and effectiveness in reducing dimensionality while preserving or enhancing the predictive power of models.

Below is a Table I summarizing the types of datasets used in the experiments:

B. Evaluation Metrics for Feature Selection

Evaluating the effectiveness of feature selection techniques is crucial for understanding their impact on machine learning models, particularly in terms of performance and computational efficiency. Several metrics are commonly employed to assess the quality of feature selection methods, which include their mathematical formulations:

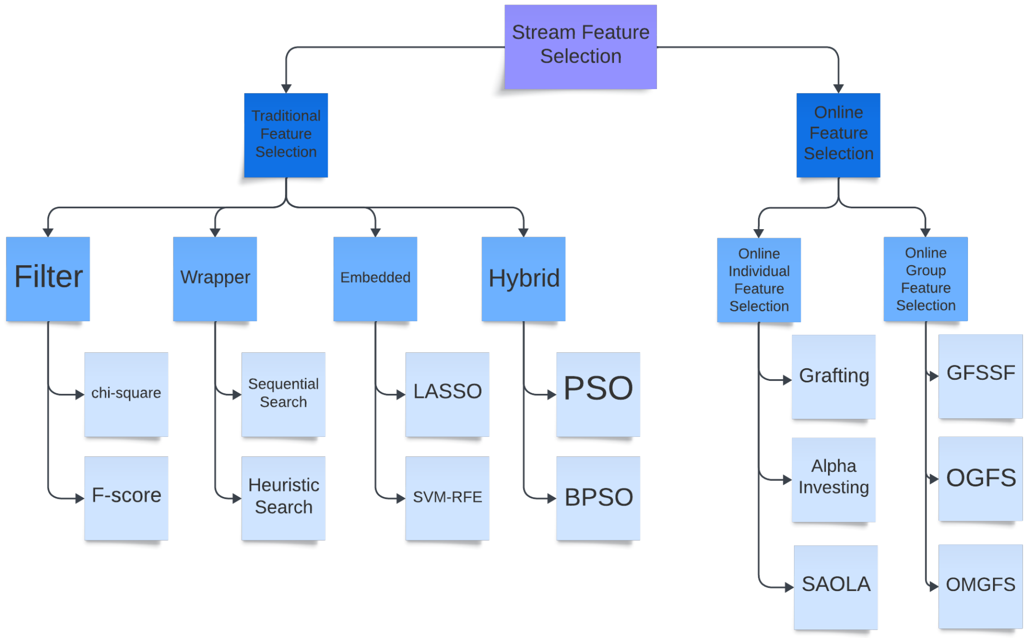

Fig. 1. Taxonomy of FS methods

No	Dataset	# Attributes	# Train	# Test	Type
1	dorothea	100,000	800	800	Pharmacology
2	arcene	10,000	100	700	Mass Spectrometry
3	dexter	20,000	300	2,000	Text classification
4	madelon	500	2,000	1,800	Artificial
5	sylva	216	13,086	130,854	Ecology
6	hiva	1,617	3,845	38,449	Pharmacology
7	nova	16,969	1,754	17,537	Text classification
8	sido0	4,932	12,678	10,000	Pharmacology
9	cina0	132	16,033	10,000	Econometrics
10	ALLAML	7,129	72	-	Biology
11	lymphoma	4,026	62	-	Biology
12	VOC 2007	6,096	5,011	4,952	Image classification
13	VOC 2012	6,096	11,530	11,001	Image classification
14	tm1	100	1,000	1,000	Synthetic
15	tm2	100	1,000	1,000	Synthetic

TABLE I
SUMMARY OF DATASETS USED IN FEATURE SELECTION STUDIES

Accuracy measures the proportion of true results (both true positives and true negatives) in the dataset and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives, respectively [5].

Precision (or Positive Predictive Value) quantifies the accuracy of positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

This metric is crucial where the cost of a false positive is high [6].

Recall (or Sensitivity) indicates the model's ability to identify all relevant instances:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

It is essential in applications where missing a positive instance can have detrimental effects [7].

The **F-measure** or F1 score harmonizes precision and recall, providing a single measure of test accuracy:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

The F1 score is particularly useful when seeking a balance between precision and recall [8].

Furthermore, the **number of selected features** assesses the efficiency of the feature selection process in reducing the feature space:

$$\text{Number of Selected Features} = |S| \tag{5}$$

where $|S|$ is the cardinality of the selected feature subset. This metric is indicative of the method's capability to minimize computational costs while maintaining model accuracy [9].

Employing these metrics allows researchers to provide a detailed and nuanced evaluation of feature selection methods, assessing not only their impact on model accuracy but also on computational efficiency and model interpretability.

II. Traditional Feature Selection

A. Relevancy and redundancy analysis

The goal of streaming feature selection is to dynamically select a subset of features from a multidimensional dataset that enhances both accuracy and robustness. This is achieved by eliminating features that are either irrelevant or redundant. In streaming feature selection, the optimal final feature subset should be pertinent to the class and should not exhibit redundancy with other existing features to maximize robustness. Consequently, the feature selection process can be divided into two stages: relevance analysis and redundancy analysis.

Relevancy Analysis. In this stage, the relevance of each feature to the target class is evaluated. The criterion for relevance determines how effectively a feature can distinguish between different classes [2].

$$Relevance(X, Y) = \text{how useful} X \text{is for predicting } Y \tag{6}$$

Gain ratio(GR) [10]. This metric assesses the value of a feature by measuring the gain ratio with respect to the class. It is given by the formula:

$$GR = \frac{H(\text{class}) - H(\text{class}|\text{attribute})}{H(\text{attribute})} \tag{7}$$

where H represents entropy.

ReliefF [10] evaluates a feature's worth by sampling instances multiple times and comparing the feature values of the nearest instances from the same and different classes. The formula for ReliefF is

$$W(A_1) = W(A_1) - \frac{\sum_{j=1}^{k} \text{diff}(A_1, R_i, H_j)}{g \cdot k} + \frac{\sum_{c \neq \text{class}(R_i)} \left[\frac{p(c)}{1 - p(\text{class}(R_i))} \sum_{j=1}^{k} \text{diff}(A_1, R_i, M_j(c)) \right]}{g \cdot k}, \tag{8}$$

where

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)} \tag{9}$$

Significance [10] evaluates the worth of a feature by computing its probabilistic significance, considering both attribute-class and class-attribute associations.

Symmetrical uncertainty (SU) [10] assesses a feature's value by measuring its symmetrical uncertainty with respect to the class, given by the formula:

$$SU = 2 \cdot \frac{H(\text{class}) - H(\text{class}|\text{attribute})}{H(\text{class}) + H(\text{attribute})} \tag{10}$$

Redundancy Analysis. This stage evaluates the similarity between features to determine how much adding a new feature can improve the accuracy of a machine learning model.

Correlation-based feature selection (CFS) [11] ranks the relevance of features by measuring the correlations between features and the class, as well as between the features themselves. Given k features and C classes, CFS defines the relevance of the feature subset using Pearson's correlation equation:

$$\text{Merit}_s = \frac{k r_{kc}}{\sqrt{k + (k-1) r_{kk}}} \tag{11}$$

where $Merit_s$ is the relevance of the feature subset, r_{kc} which is defined as the average linear correlation coefficient among features and classes. Also, r_{kk} is defined as the average linear correlation coefficient among unique individual features. CFS typically adds or removes one feature at a time using forward or backward selection. However, in this research, sequential forward floating search (SFFS) [12] is employed as the search method. The number of forward and backward steps is dynamically controlled based on the selected subset criterion, eliminating the need for parameter settings.

General flow of OFS. Figure 2 illustrates the general flow steps of OFS, as follows:

- Populate new features (single/group stream).
- Determine the addition of new features to the selected subset by relevancy, redundancy, and irrelevancy analysis.
- Update the subset of the existing features.
- Repeat steps 1 to 3 until all the feature space has been examined and the optimal subset has been found.

B. Filter method

Filter methods evaluate feature relevance based on statistical characteristics of the data, independent of learning techniques. These methods use statistical tests such as distance correlation and information gain to score each feature based on its significance. Features with the highest scores are selected to form a subset for classification, while others are excluded from the dataset.

This approach involves two main phases: relevance ranking and evaluation. In the relevance ranking phase, features are ranked according to measures like correlation, consistency, distance, dependency, similarity, or information. During the evaluation phase, typically the higher-ranked features are chosen to build a classifier, and the lower-ranked features are filtered out.

Previous studies applied the chi-square [13] statistic measure, Fisher-Score(F-Score) [14], PCA-Entropy [15] and

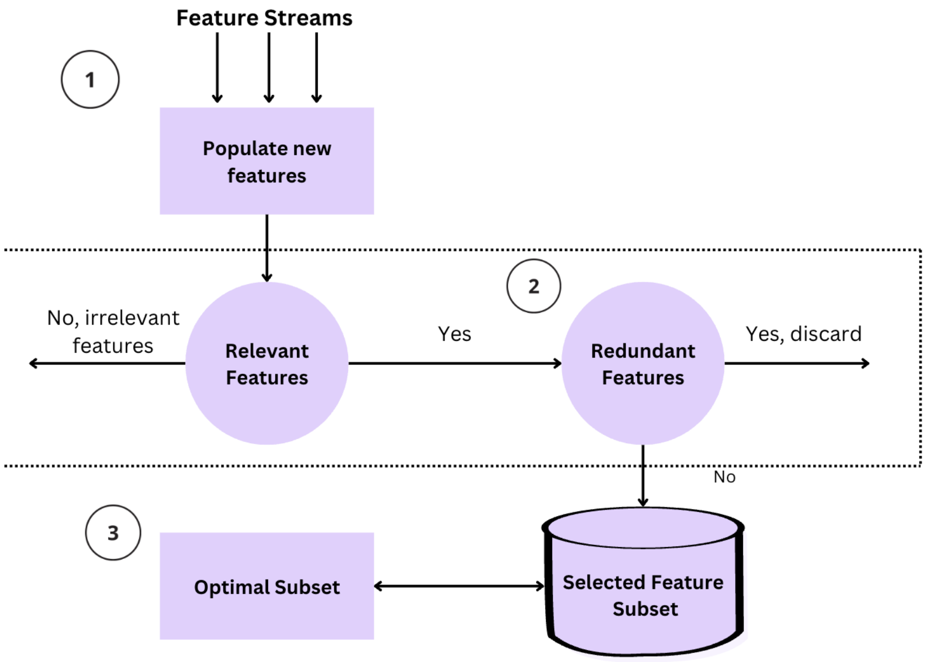

Fig. 2. General Flow of OFS

Information Gain(IG) [16] to weight the features and then arrange them accordingly to choose the features of greater importance.

C. Wrapper method

Wrapper methods form another significant category of feature selection (FS). They use learning algorithms to evaluate candidate feature subsets, determining the importance of each subset, and adopt a search method to select the optimal final subset. Wrapper methods are generally classified into Sequential Search and Heuristic Search methods. Sequential Forward Selection (SFS) adds candidate features one at a time, selecting those that most significantly improve classification performance [17]. Genetic Algorithms (GA) generate feature subsets, which are then evaluated using a supervised machine-learning algorithm [18].

D. Embedded methods

Embedded methods integrate the feature selection (FS) process with the classifier itself, guiding FS during model training. The core idea is to incorporate FS into the model interaction, leveraging its characteristics for feature evaluation. The advantage of embedded methods is their lower computational complexity compared to wrappers, as they interact directly with the classifier. For instance, the Least Absolute Shrinkage and Selection Operator (LASSO) [19] performs FS by shrinking some coefficient estimates to zero, using the remaining non-zero coefficients as selected features. Recursive Feature Elimination for Support Vector Machines (SVM-RFE) [20] uses SVM as the classifier, selecting features based on their importance derived from SVM weights and removing the least important ones. Variable Step Size RFE (VSSRFE) is an enhanced version of RFE that aims to reduce the time consumption of SVM-RFE [21].

E. Hybrid methods

Hybrid methods combine filter and wrapper techniques to leverage the strengths of both approaches. The filter technique is initially used to obtain a good subset of features, which is then refined using the wrapper technique to enhance the results. Recent research has proposed various hybrid FS methods.

For example, Inbarani et al. [22] combined the wrapper PSO with rough sets theory to improve disease diagnosis classification accuracy. Similarly, Pashaei et al. [23] developed an FS method using a binary black hole algorithm and improved BPSO for cancer classification.

III. Online Feature Selection

The methods previously discussed assume that all features and instances of data are available beforehand. However, an intriguing alternative scenario involves features being generated and arriving dynamically, either individually or in groups, requiring immediate processing upon arrival. This scenario is known as Online Feature Selection (OFS), which presents greater challenges compared to traditional feature selection (TFS).

In many real-world applications, there are two main types of streaming data: data streams and feature streams. The primary distinction between these lies in their characteristics. In OFS with data streams, the number of features remains fixed while the number of candidate instances varies over time. Conversely, in OFS with feature streams, the number of data instances is fixed, but the number of candidate features increases incrementally.

OFS methods can be divided into two sub-categories: Online Individual Feature Selection (OIFS) and Online Group Feature Selection (OGFS). In the subsequent sections, we first describe OIFS and review the existing related works, followed by a discussion of OGFS and its related works. Finally, we provide a comparative analysis of these methods.

A. Online individual feature selection

Online individual feature selection shares the common assumption that candidate features are generated dynamically and arrive one at a time. This method is of high significance when dealing with real-world data-intensive applications, which require an efficient OFS method in order to cope with real-time data streaming applications.

1) Grafting: Grafting is an incremental feature selection method specifically designed for high-dimensional data. Introduced by Perkins et al. [24], the primary motivation behind Grafting is to efficiently handle scenarios where features are continuously added, enabling real-time adaptation to streaming data. This method incrementally incorporates features into the model based on their contribution to the reduction of the loss function, balanced by a regularization term to manage model complexity and prevent overfitting.

The Grafting algorithm evaluates the gradient of the loss function with respect to each feature and incorporates features whose gradients exceed a specified threshold. This can be expressed as:

$$\frac{\partial L}{\partial w_j} > \lambda \tag{12}$$

where L is the loss function, w_j is the weight of the feature j, and λ is a regularization parameter. The method ensures the model remains sparse by only selecting features that significantly improve the model's performance. The process is repeated iteratively as new features arrive, making Grafting well-suited for online learning environments.

2) Alpha-Investing: Alpha-Investing, proposed by Zhou et al. [25], is an online feature selection algorithm that aims to control the false discovery rate (FDR) in an incremental setting. The motivation for Alpha-Investing is to balance the discovery of significant features with maintaining statistical rigor during the feature selection process.

The algorithm starts with an initial "alpha wealth," which is invested in testing new features. Successful inclusion of features increases the alpha wealth, allowing for more tests, while unsuccessful tests reduce it. This dynamic adjustment helps maintain control over the FDR.

The alpha-investing rule can be formalized as follows:

$$\alpha_{t+1} = \alpha_t \cdot \left(1 + \frac{\delta}{k}\right) \tag{13}$$

where α_t is the alpha level at time t,δ is a small constant, and k is the number of features tested so far. This approach dynamically adjusts the testing threshold based on the number of successful feature inclusions, ensuring a balance between feature discovery and statistical control.

3) SAOLA: The Sparse Online Active Learning Algorithm (SAOLA), developed by Yu et al. [26], addresses the challenges of feature selection in streaming data, focusing on sparsity and computational efficiency. SAOLA's motivation is to maintain a sparse model by actively selecting features that contribute most to the learning task while discarding irrelevant ones.

SAOLA employs a budgeted allocation approach, where only a limited number of features are allowed to be active at any time. It uses a correlation threshold to determine whether a new feature should be included or an existing feature should be discarded. This can be expressed as:

$$|\text{corr}(X_i, y)| > \theta \tag{14}$$

where $\text{corr}(X_i, y)$ is the correlation between feature X_i and the target y , and θ is the threshold. This criterion ensures that only the most relevant features are included in the model, maintaining its sparsity.

B. Online group feature selection

The methods discussed in the previous subsection can successfully select a feature from the streaming feature only at the individual feature level without taking into account the group structures of online features. In the case of group structures, the selection of features at both the individual feature level and the group level is more preferred. In Group FS the selection of signifcant groups rather than individual features. Several OGFS methods are proposed in the literature to address the problem of FS at both the individual and group feature levels.

1) GFSSF: Group Feature Selection with Streaming Features (GFSSF) is an extension of traditional feature selection methods to group-wise selection in a streaming context. Proposed by Zhang and Li [27], the motivation for GFSSF is to handle scenarios where features arrive in groups, which is common in applications such as text processing and bioinformatics.

GFSSF evaluates and selects groups of features based on their collective contribution to the model. It employs a group lasso regularization technique to promote sparsity at the group level. The group lasso objective function is given by:

$$\min_w \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - X_i w)^2 + \lambda \sum_{g=1}^{G} \sqrt{|w_g|} \right\} \tag{15}$$

where N is the number of instances, X_i is the feature vector for instance i, y_i is the target, w_g represents the weights of features in group g, and λ is the regularization parameter. This approach ensures that only the most relevant groups of features are selected, maintaining model sparsity and interpretability.

2) OGFS: Online Group Feature Selection(OGFS), introduced by He and Tang [28], is designed for scenarios where feature groups arrive sequentially, with the goal of selecting the most informative groups in an online manner. The motivation behind OGFS is to leverage the grouped structure of features to enhance selection efficiency and model performance.

OGFS uses a scoring mechanism to evaluate the relevance of incoming feature groups, selecting groups that maximize predictive performance. It incorporates a group-wise regularization term to ensure sparsity. The scoring function can be expressed as:

$$S(G_i) = \frac{\sum_{j \in G_i} |w_j|}{|G_i|} \tag{16}$$

where $S(G_i)$ is the score of group G_i, w_j is the weight of feature j, and $|G_i|$ is the size of the group. This method ensures that only the most relevant groups are included in the model, improving its performance and robustness.

3) OMGFS: Online Multi-Group Feature Selection (OMGFS), proposed by Zhao and Wang [29], extends the concept of OGFS to handle scenarios where multiple groups of features arrive simultaneously. The motivation for OMGFS is to address complex streaming environments where feature dependencies across groups need to be considered.

OMGFS evaluates and selects feature groups based on their joint contribution to the model, using a multi-group lasso regularization to enforce sparsity across and within groups. The multi-group lasso objective is given by:

$$\min_w \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - X_i w)^2 + \lambda_1 \sum_{g=1}^{G} \sqrt{|w_g|} + \lambda_2 \sum_{h=1}^{H} \sqrt{|w_h|} \right\} \tag{17}$$

where λ_1 and λ_2 are regularization parameters for different group hierarchies. This approach ensures that the model remains sparse while capturing the dependencies across multiple feature groups, improving its robustness and predictive power.

IV. Challenges and Methods

As datasets grow in size and complexity, traditional feature selection methods often fall short due to their inability to adapt to new data and to scale efficiently [30]. OFS tackles this problem by providing mechanisms that can handle various data characteristics, including high dimensionality, multi-label configurations, class imbalances, and more. Each category of data presents unique challenges, necessitating specialized OFS approaches that are optimized for specific scenarios [31]. For instance, high-dimensional data requires dimensionality reduction techniques to manage the curse of dimensionality, while multi-label data needs strategies that can handle multiple dependent labels effectively.

A. Online Feature Selection on Group Stream

Group stream OFS methods focus on datasets where features arrive in groups rather than individually. This approach often involves techniques that can dynamically group features and evaluate their relevance and redundancy collectively. An effective method involves using mutual information to assess the interdependence within and across groups, optimizing the feature selection for grouped data [32]. To illustrate, in genomics and other biological sciences, data inherently arrives in grouped formats. Utilizing mutual information in group stream OFS allows for the assessment of shared information within and between groups, which is critical for maintaining the functional integrity of biological data. Techniques such as hierarchical clustering or network-based clustering can also be employed to determine natural groupings of features before applying OFS, thereby ensuring that dimensionality reduction and feature selection do not disrupt underlying biological relationships. This method not only enhances the interpretability of the data but also significantly increases the efficiency of the feature selection process by reducing redundant or irrelevant groups of features [32]. For example, the Group-SAOLA algorithm enhances feature selection by considering the interdependencies within feature groups, reducing redundancy while maintaining relevance [30].

B. Online Feature Selection with High Dimensional Data

High-dimensional datasets, common in areas like image processing and genomics, suffer from the ”curse of dimensionality” where the feature space greatly exceeds the number of observations. Techniques such as PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) are used to reduce dimensions but might lose critical information. Sparse models like LASSO are particularly effective as they perform both variable selection and regularization, pushing coefficients of less important variables towards zero and effectively eliminating them from the model. This results in simpler, more interpretable models that are not only easier to validate but also less prone to overfitting, improving both the predictive performance and generalizability [33].

C. Online Feature Selection on Multi-label

In multi-label data contexts, each instance can be tagged with multiple labels, which adds complexity to the feature selection process as it must consider the correlation between labels and features. OFS in such environments often adapts traditional methods like binary relevance or classifier chains to better address these complexities. Binary relevance methods treat each label as a separate binary classification problem, ignoring label interdependencies, whereas classifier chains attempt to incorporate these dependencies by building a chain of binary classifiers, where each classifier deals with one label and includes the predictions of previous classifiers as additional features. These strategies aim to enhance the accuracy of feature selection by acknowledging the connections between labels. Furthermore, an innovative approach involves applying problem transformation methods that convert complex multi-label problems into simpler single-label frameworks, thereby streamlining the feature selection process. This method, as discussed in [34], involves transforming the label space to make the data more manageable for traditional single-label feature selection algorithms,

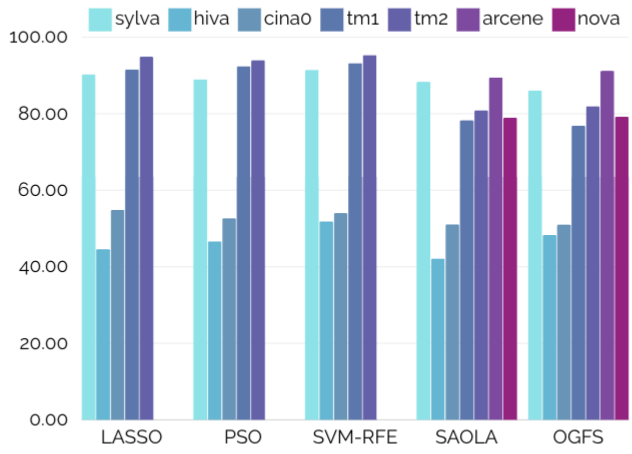

Fig. 3. SVM f1-scores for the feature selection methods

potentially improving both the efficiency and effectiveness of the feature selection in multi-label settings.

D. Online Feature Selection on Class Imbalance

Dealing with class imbalance involves methods that emphasize feature selection techniques that can identify and enhance minor class signals. Techniques such as synthetic minority over-sampling (SMOTE) combined with OFS can help to alleviate the imbalance by artificially enhancing the minority class's representation in the training data, thus providing a more balanced dataset for feature selection and subsequent model training [31].

E. Online Feature Selection on Feature Drift

Feature drift refers to the phenomenon where the importance of features changes over time. Adapting to feature drift requires methods that can dynamically adjust the selection criteria based on evolving data streams. Online bagging and boosting algorithms, which can adjust weights of features based on their evolving importance, are effective in these scenarios [35].

V. Results and Analysis

In this section of results and analysis, Table II presents a comparative overview of various algorithms along dimensions crucial for handling different data characteristics in feature selection tasks. Notably, algorithms like Group-SAOLA and Online Group Feature Selection (OGFS) demonstrate versatility across multiple streams, effectively managing both individual and group feature selection. These are particularly adept in scenarios involving streaming data, as indicated by their capabilities in single stream (SS) and group stream (GS) settings, and adaptability to feature drift (FD). Additionally, techniques such as Sequential Feature Selection, Grafting, and Alpha Investing are designed for streaming data environments, providing robust solutions for dynamic feature selection.

Importantly, methods like SMOTE address class imbalance by oversampling the minority class, which is crucial for improving the performance of classifiers in binary and multiclass problems. The use of PSO illustrates the application of optimization strategies for feature selection, highlighting its adaptability to various classifiers and data structures.

Table II serves as a crucial tool for researchers and practitioners to select appropriate algorithms tailored to the unique challenges of their data landscapes, ensuring optimal feature selection and consequent model performance.

The comparison of SVM f1-scores for the feature selection methods is shown in Table III and in Figure 3. The results we can get from this analysis is that only streaming algorithms (SAOLA and OGFS) are capable of processing feature space for very large datasets. Other algorithms are unable to complete their work due to time or memory limits. As three batch algorithms (LASSO, PSO and SVM-RFE) search the whole feature space to evaluate each candidate feature, they are unable to process large datasets. However, for small and medium-scaled datasets, such a search is an advantage and results in highly accurate subsets.

VI. Tools

During the research for stream feature selection, several open-source toolboxes for online feature selection were developed, which is applicable for both research and practical applications. The comparison between tools is shown in Table IV.

A. LOFS

LOFS [36] is an open-source toolbox designed to support various online feature selection algorithms. Developed to facilitate research and practical applications in online learning environments, LOFS provides implementations of several state-of-the-art algorithms, enabling users to efficiently select relevant features from streaming data. LOFS includes implementations of multiple online feature selection methods, such as Alpha-Investing, Grafting, and SAOLA. This allows researchers to compare different approaches and select the most appropriate algorithm for their specific needs. The toolbox is designed with a modular structure, making it easy to extend with new algorithms and features. Users can customize and enhance the library based on their requirements. According to past publications, LOFS is used by researchers to test not only single and group streams but also class imbalance, sparse data, causal data, and high-dimensional data.

B. MOA

MOA is a comprehensive open-source framework for data stream mining, which includes functionalities for online feature selection. Developed by the University of Waikato [37], MOA is particularly known for its scalability and ability to handle massive datasets in real-time. MOA includes three modules: (1) data generators (such as AGRAWAL, Random Tree Generator, and SEA), (2) evaluation methods (such as periodic holdout, test-then-train, and prequential), and

Technique	Classifier	Platform	Structure	Strategy	Supervision	Classification Type
Sequential FS	Any classifier	Multiple	Single	Feature Selection	Supervised	Binary, Multiclass
LASSO	Linear models with L1 regularization	Multiple	Single	Feature Selection	Supervised	Binary, Multiclass
Grafting	Support Vector Machine (SVM)	Matlab	Single, Streaming	Data Streaming	Supervised	Binary
SMOTE	Any classifier	Multiple	Single	Data Sampling	Supervised	Binary, Multiclass
PSO	Any classifier	Multiple	Single	Optimization	Supervised	Binary, Multiclass
Alpha investing	Applied stream-wise regression	R	Streaming	Data Streaming	Supervised	Binary
OSFS	K-NN, J48, Random Forest (RF)	C Language	Single	Feature Selection	Supervised	Binary
SAOLA	J48 and KNN	Matlab	Single	Feature Selection	Supervised	Multiclass
OGFS	K-NN, J48, RF	Not mentioned	Single, Group	Feature Selection	Supervised	Multiclass
Group-SAOLA	J48, SVM, KNN	Matlab	Single, Group	Feature Selection	Supervised	Multiclass
OSFS-KW	KNN, SVM, and RF	Matlab	Single	Feature Selection	Supervised	Binary

TABLE II

COMPARISON OF DIFFERENT FEATURE SELECTION METHODS

Dataset	LASSO	PSO	SVM-RFE	SAOLA	OGFS
dorothea	-	-	-	96.49	95.12
arcene	-	-	-	89.45	91.22
dexter	-	-	-	78.2	81.95
madelon	57.99	61.54	-	60.24	61.09
sylva	90.31	88.98	91.45	88.32	86.08
hiva	44.57	46.65	51.88	42.16	48.32
nova	-	-	-	78.98	79.23
sido0	77.01	75.93	-	75.21	74.87
cina0	54.85	52.69	54.04	51.1	51
ALLAML	-	-	-	58.83	58.97
lymphoma	71.08	73.52	-	69.68	70.98
VOC 2007	73.34	79.12	-	74.25	73.58
VOC 2012	79.58	79.21	-	75.49	76.33
tm1	91.57	92.41	93.21	78.25	76.87
tm2	94.9	94	95.32	80.9	81.96

TABLE III

COMPARISON OF DIFFERENT METHODS

Data Behavior	LOFS	MOA	LIBOL	KEEL
Single Stream	✓	✓		
Group Stream	✓			
Class Imbalance	✓			
Sparse Data	✓			
Causal Based	✓			
HDD	✓	✓	✓	
Ultra High Dimensional	✓		✓	
Feature Drift		✓		
Multi Label				✓

TABLE IV

TOOLS USED FOR STREAM FEATURE SELECTION

(3)statistics (CPU time, RAM-hours, and Kappa). MOA has a GUI(Graphical User Interface) and a command line interface, facilitating batches of tests. The implementation is written in Java and it shares many features with the WEKA framework, including the ability to extend the framework by inheriting abstract classes.

C. LIBOL

LIBOL [38] is a specialized library designed for online learning and online feature selection. It offers a robust set of tools for real-time data processing and feature selection, aimed at researchers and practitioners working with dynamic data environments. LIBOL supports a variety of online learning algorithms, including those for online classification, regression, and feature selection. The library is optimized for high performance, ensuring that algorithms can process data efficiently in real-time. Users can easily customize and extend the library with new algorithms or modifications to existing ones.

D. KEEL

KEEL (Knowledge Extraction based on Evolutionary Algorithms) is a JAVA software tool designed for a broad range of data mining tasks, including online feature selection. It provides a rich environment for testing evolutionary learning algorithms and their applications to online and offline data mining problems. KEEL includes a user-friendly GUI, allowing users to easily configure and execute experiments without needing extensive programming skills. The toolkit comes with comprehensive documentation, tutorials, and example projects to help users get started and understand the functionality.

VII. CONCLUSION

This study presented a comprehensive survey of recent FS algorithms for both static and dynamic environments across various domains, along with a taxonomy categorizing these methods based on their search strategy, evaluation process, and feature structure. Initially, the study reviewed existing traditional and online FS methods, providing a qualitative analysis of their strengths and weaknesses. The proposed taxonomy also includes a quantitative analysis of these techniques based on their category and publication timeline. Also, we discussed the useful tools that allow researchers to implement stream feature selection methods.

This survey aims to enhance the efficiency of learning state-of-the-art FS methods and assist researchers in understanding and applying key characteristics of FS in Big Data. It also helps identify limitations and research gaps in current FS methods.

While our paper provided a detailed discussion on online methods, the experimental analysis was limited to SVM and 5 stream selection methods. In future work, we plan to include experimental analysis of more classifiers and more feature selection methods.

References

[1] Mufeed Ahmed Naji Saif2 Hudhaifa Mohammed Abdulwahab, S.Ajitha1. Feature selection techniques in the context of big data: taxonomy and analysis. *Applied Intelligence*, 52:13568–13613, 2022.

[2] Benjamin Auffarth, Maite López, and Jesús Cerquides. Comparison of redundancy and relevance measures for feature selection in tissue classification of ct images. In *Industrial conference on data mining*, pages 248–262. Springer, 2010.

[3] UC Irvine. Machine learning repository: Data sets. *University of California*.

[4] UC Irvine. Feature selection repository. *Arizona State University (ASU)*.

[5] Guyon and Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 2003.

[6] Baeza-Yates and Ribeiro-Neto. Modern information retrieval. *ACM Press - New York*, 1999.

[7] David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technology*, 2011.

[8] Van Rijsbergen. Data and information retrieval. 1979.

[9] Dash and Liu. Feature selection for classification. *Intelligent Data Analysis*, 1.

[10] Rakkrit Duangsoithong and Terry Windeatt. Relevance and redundancy analysis for ensemble classifiers. In *Machine Learning and Data Mining in Pattern Recognition: 6th International Conference, MLDM 2009, Leipzig, Germany, July 23-25, 2009. Proceedings 6*, pages 206–220. Springer, 2009.

[11] Mark A Hall. Correlation-based feature selection of discrete and numeric class machine learning. 2000.

[12] Pavel Pudil, Jana Novovičová, and Josef Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994.

[13] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE international conference on tools with artificial intelligence*, pages 388–391. Ieee, 1995.

[14] Richard O Duda, Peter E Hart, et al. *Pattern classification*. John Wiley & Sons, 2006.

[15] V Madhusudan Rao and VN Sastry. Unsupervised feature ranking based on representation entropy. In *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, pages 421–425. IEEE, 2012.

[16] Mark A Hall and Lloyd A Smith. Practical feature subset selection for machine learning. 1998.

[17] A Wayne Whitney. A direct method of nonparametric measurement selection. *IEEE transactions on computers*, 100(9):1100–1103, 1971.

[18] Stjepan Oreski and Goran Oreski. Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert systems with applications*, 41(4):2052–2064, 2014.

[19] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[20] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:389–422, 2002.

[21] Zifa Li, Weibo Xie, and Tao Liu. Efficient feature selection and classification for microarray data. *PloS one*, 13(8):e0202167, 2018.

[22] H Hannah Inbarani, Ahmad Taher Azar, and G Jothi. Supervised hybrid feature selection based on pso and rough sets for medical diagnosis. *Computer methods and programs in biomedicine*, 113(1):175–185, 2014.

[23] Elnaz Pashaei, Elham Pashaei, and Nizamettin Aydin. Gene selection using hybrid binary black hole algorithm and modified binary particle swarm optimization. *Genomics*, 111(4):669–686, 2019.

[24] Simon Perkins and James Theiler. Online feature selection using grafting. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 592–599, 2003.

[25] Jing Zhou, 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*, pages 384–393, 2005.

[26] Kui Yu, 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(2):1–39, 2016.

[27] Haiguang Li, Xindong Wu, Zhao Li, and Wei Ding. Group feature selection with streaming features. In *2013 IEEE 13th International Conference on Data Mining*, pages 1109–1114. IEEE, 2013.

[28] Xindong Wu, Kui Yu, Hao Wang, and Wei Ding. Online streaming feature selection. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 1159–1166, 2010.

[29] Jing Wang, 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(11):3029–3041, 2015.

[30] Agata Lapedriza Aude Oliva Antonio Torralba Bolei Zhou, Aditya Khosla. Conference on computer vision and pattern recognition (cvpr), 2016, pp. 2921-2929. 2016.

[31] H. White and M Smith. Integrating smote with online feature selection for class imbalance learning. *Pattern Recognition Letters*, 2022.

[32] Doe P Smith, J. Mutual information in grouped data feature selection. *Journal of Computational Biology and Bioinformatics Research*, 4:345–356, 2020.

[33] R Johnson. Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 2021.

[34] S. Lee and J Kim. Efficient multi-label feature selection with the application of problem transformation methods. *Information Sciences*, 2019.

[35] G. Tanner and P Wong. Dynamic feature selection for online learning with feature drift in data streams. *Machine Learning*, 2020.

[36] Kui Yu, Wei Ding, and Xindong Wu. Lofs: A library of online streaming feature selection. *Knowledge-Based Systems*, 113:1–3, 2016.

[37] Mahmood Shakir Hammoodi, Hasanain Ali Al Essa, and Wial Abbas Hanon. The waikato open source frameworks (weka and moa) for machine learning techniques. In *Journal of Physics: Conference Series*, volume 1804, page 012133. IOP Publishing, 2021.

[38] Jialei Wang, Peilin Zhao, and Steven CH Hoi. Libol: A library for online learning algorithms. 2014.