Applied Computing and Informatics xxx (xxxx) xxx

Contents lists available at ScienceDirect

Applied Computing and Informatics

journal homepage: www.sciencedirect.com

# Streaming feature selection algorithms for big data: A survey

Noura AlNuaimi *, Mohammad Mehedy Masud, Mohamed Adel Serhani, Nazar Zaki

*College of Information Technology, United Arab Emirates University, Sheik Khalifa Bin Zayed Street, Al Ain, Abu Dhabi, United Arab Emirates*

ARTICLE INFO

*Article history:*
Received 24 October 2018
Revised 9 January 2019
Accepted 15 January 2019
Available online xxxx

*Keywords:*
Big data
Redundant features
Relevant features
Streaming feature grouping
Streaming feature selection

ABSTRACT

Organizations in many domains generate a considerable amount of heterogeneous data every day. Such data can be processed to enhance these organizations' decisions in real time. However, storing and processing large and varied datasets (known as big data) is challenging to do in real time. In machine learning, streaming feature selection has always been considered a superior technique for selecting the relevant subset features from highly dimensional data and thus reducing learning complexity. In the relevant literature, streaming feature selection refers to the features that arrive consecutively over time; despite a lack of exact figure on the number of features, numbers of instances are well-established. Many scholars in the field have proposed streaming-feature-selection algorithms in attempts to find the proper solution to this problem. This paper presents an exhaustive and methodological introduction of these techniques. This study provides a review of the traditional feature-selection algorithms and then scrutinizes the current algorithms that use streaming feature selection to determine their strengths and weaknesses. The survey also sheds light on the ongoing challenges in big-data research.


**Contents**



* Corresponding author.
*E-mail addresses:* noura.alnuaimi@uaeu.ac.ae (N. AlNuaimi), m.masud@uaeu.ac.ae (M.M. Masud), serhanim@uaeu.ac.ae (M.A. Serhani), nzaki@uaeu.ac.ae (N. Zaki).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.aci.2019.01.001

## 1. Introduction

Feature-selection techniques are an important part of machine learning. Feature selection is often termed as variable selection, attribute selection and variable subset selection. It is the process of reducing input features to the most informative ones for use in model construction. Feature selection should be distinguished from feature extraction. Although, both techniques are used to reduce the number of features in a dataset, feature extraction is reduction technique in dimensionality that creates new combinations of attributes, whereas feature selection includes and excludes the attributes that are present in the data without changing them.

Streaming feature selection has recently received attention with regard to real-time applications. Feature selection with streaming data, known as streaming feature selection or online streaming feature selection is a popular technique that uses selection of features that are most informative to reduce streaming data size.

In streaming feature selection, the candidate features arrive sequentially. The size of these features is unknown. Streaming feature selection has a critical role in real time applications, for which the required action must be taken immediately. In applications such as weather forecasting, transportation, stock markets, clinical research, natural disasters, call records, and vital-sign monitoring, streaming feature selection plays a key role in efficiently and effectively preparing big data for the analysis process in real time.

At present, contemporary methods in machine learning are being challenged by big data as newer and faster algorithms deal with variable volumes of data. Making decisions in real time from such continuous data could bring data monetization benefit which is a major source of revenue. The world is projected to generate over 180 zettabytes (or 180 trillion gigabytes) of data by 2025. This figure when compared with 10 zettabytes worth data created as of 2015 seems ubiquitous. The presence of large datasets is the reason for emergence of deep learning which further led to artificial intelligence. Companies such as Google, Facebook, Baidu, Amazon, IBM, Intel, and Microsoft are investing in capturing talent pool to understand big data and release open artificial intelligence hardware and software [1].

Using big data for streaming feature selection is regarded as a solution to select the most informative features that could support the development of robust and accurate machine learning models. There are several techniques in data analytics. The newer algorithms on dimensionality reduction are asymptotically better than the previous algorithms. Prior research on feature selection has targeted searching for relevant features only. John et al. [2] proposed three categories belonging to X input features and its importance in C target class: (1) strongly relevant, (2) weakly relevant, and (3) irrelevant. Yu and Liu [3] improved this categorization by proposing a definition of feature redundancy therefore creating a path for efficient elimination of redundant features.

Let $F$ be a full set of features, $F_i$ a feature and $S_i = F - \{F_i\}$. The definitions and techniques are listed as follows:

**Definition 1 (Strong relevance).** *Feature Fi is strongly relevant if and only if*

$$P(C|F_i S_i) \neq P(C|S_i). \quad (1)$$

Thus, a feature with strong relevance will always be in the final, optimal feature subset.

**Definition 2 (Weak relevance).** *Feature $F_i$ is weakly relevant if and only if*

$$P(C|F_i, S_i) = P(C|S_i), \text{ and } \exists S_i' \subset S_i, \text{ such that } P(C|F_i, S_i') \neq P(C|S_i'). \quad (2)$$

A feature with weak relevance is not always in the final, optimal feature subset, but ideally, it would be included.

**Definition 3 (Irrelevance).** *Feature $F_i$ is irrelevant if and only if*

$$\forall S_i' \subseteq S_i, P(C|F_i, S_i') = P(C|S_i'). \quad (3)$$

Irrelevant features are not necessary at all and thus should be discarded.

According to Yu and Liu [3] important and relevant features are segregated into necessary and unnecessary features. Yu and Liu's definition [3], which is based on Markov blanket is that redundant features provide no extra information than the currently selected features and irrelevant features provide no useful information in the final model.

The definition is from other authors is given below:

**Definition 4 (Markov blanket).** *Given a feature $F_i$, let $M_i \subset F(F_i \notin M_i)$, $M_i$ is said to be a Markov blanket for $F_i$ if and only if*

$$P(F - M_i - \{F_i\}, C|F_i, M_i) = P(F - M_i - \{F_i\}, C|M_i). \quad (4)$$

**Definition 5 (Redundant feature).** *Let G be the current set of features. A feature is redundant and hence needs to be removed from G if and only if there is a weak relevance and has a Markov blanket $M_i$ within G.*

Fig. 1 shows the relationship between redundancy and importance of a feature. The figure shows segregation of entire feature sets into four disjointed subsets comprising of a) irrelevant feature (I) b) redundant features (II) and less relevant features c) less relevant but non-redundant features (III) and d) features that are strongly relevant (IV). It also depicts an optimal subset having features of both (III) and (IV). It is necessary to mention that parts (II) and (III) are disjointed but multiple partitions of these parts can form due to Markov-blanket filtering.

The purpose of this paper is to survey the existing approaches to streaming feature selection algorithms and to review the definitions related to streaming feature selection. The study begins from Section 2 presenting the difference between streaming feature and traditional feature selection. Section 3 illustrates details of feature relevance and feature redundancy. Section 4 reports the challenges of using streaming feature selection in the analysis of big data. Section 5 presents a discussion and comparison of the current approaches to streaming feature selection. Finally, Section 6 provides the c feature selection.

## 2. Feature selection taxonomy for classification

In systems based on machine learning, streaming feature selection sometimes referred to as Online Streaming Feature Selection (OSFS) or online feature selection is a method used to choose a group of important features (e.g. variable X or multiple predictors) from streaming data to construct a theoretical model. Streaming feature selection allows for the most informative features to be selected by eliminating redundant and irrelevant features. In comparison with older feature selection methods, online feature selection leads to (a) models that are easier for researchers and users to interpret (b) lesser training time, avoiding issues and challenges related to dimensionality and (c) greater generalization through reduced over-fitting [4]. Fig. 2 illustrates the feature selection classification of data from two perspectives: static feature selection and streaming feature selection. In static data, all features and instances of data are assumed to be captured well in advance,

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

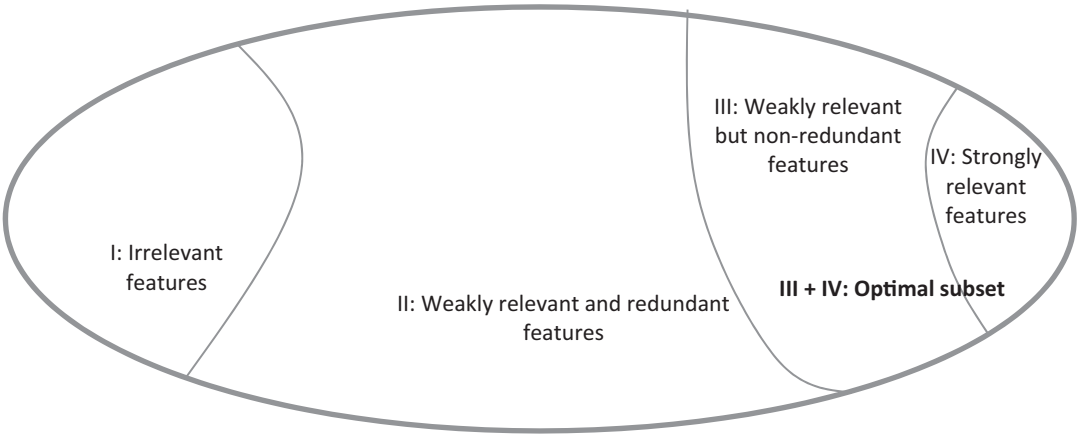


**Fig. 1.** Feature relevance and redundancy relationships.

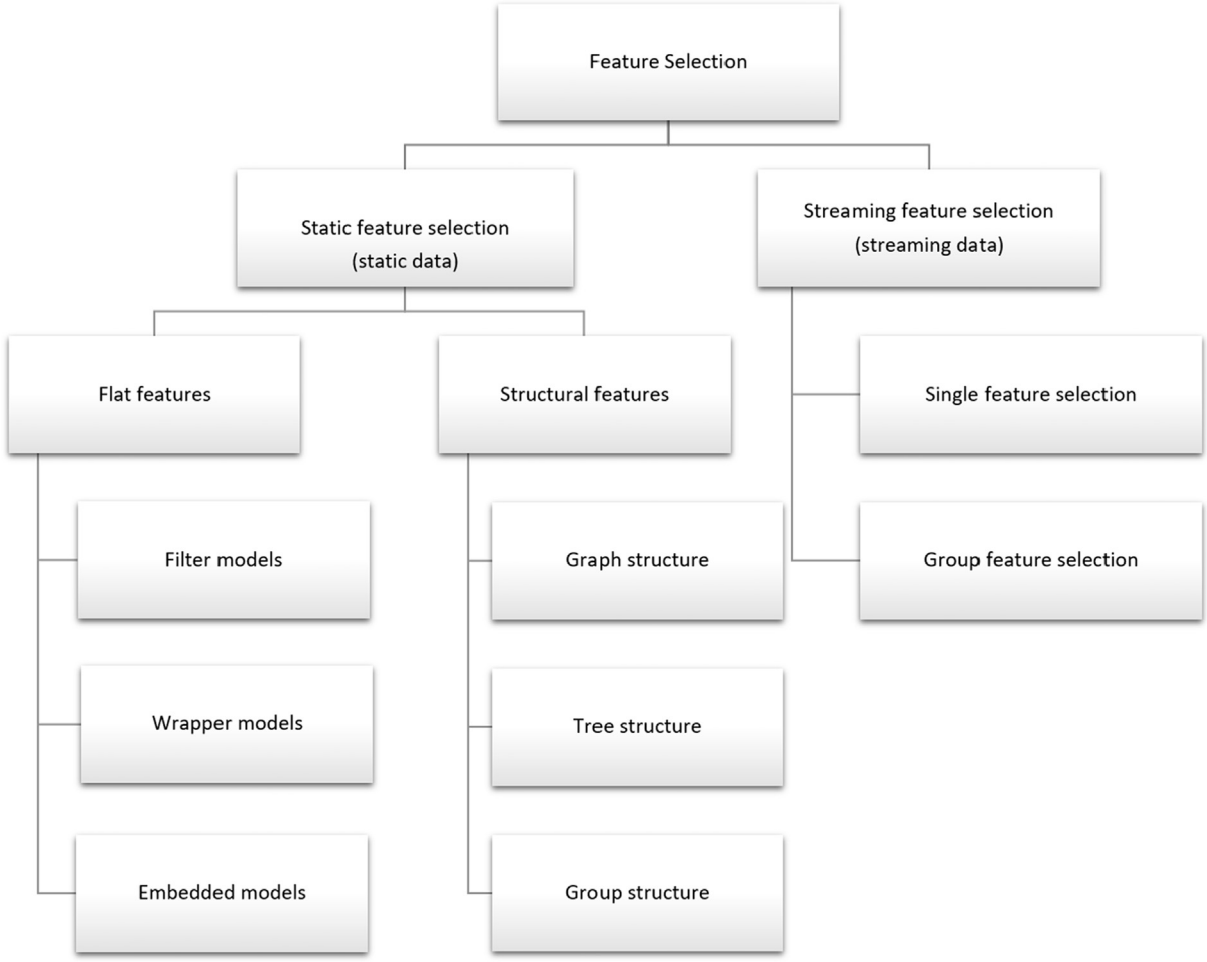


**Fig. 2.** Feature-selection classification taxonomy.

whereas streaming data has unknown numbers of data instances, features or both.

### 2.1. Static feature selection

From the features perspective, static features can be categorized as flat features or structured features. Flat features are independent. However, structures features are usually in the form of the graph structure, tree structure or group structure. A conventional approach to feature-selection is aimed at working with flat features which can be regarded as independent. Algorithms in the flat-features category are subcategorized into three main groups: filters, wrappers, and embedded models.

#### 2.1.1. Flat features

*2.1.1.1. Filter methods.* Feature selection is not related to machine-learning algorithms. Instead, they focus on application of statistical measures for assigning scores for each feature. This is followed by

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

score based feature ranking that may be selected or removed from the datasets. The methods are sometimes univariate and could consider the features independently or with regard to the dependent variable, as shown in Fig. 3. Famous algorithms from this category include the Fisher score [5,6], information theory based methods [7–9], and ReliefF and its variants [10,11].

The Fisher score, also known as the scoring algorithm [5,6] is a form of Newton's method used in statistics to numerically solve maximum likelihood equations. It is named after Ronald Fisher.

Information theory based methods which is represented as a family consisting of feature selecting algorithms are primarily methods that have its antecedents in information theory as shown in Table 1. In probability and information theory, the amount of information that two random variables share is a measure of their mutual dependence.

ReliefF and its variant feature-selection algorithms are used in the binary classification that Kira and Rendell proposed in 1992 [20], features having high quality should give matching values to instances belonging to the same class and non-matching values in case instances belong to different classes. The strength of this method is that it does not depend on the heuristics and uses low-order polynomial time to execute. There is a significant factor of noise tolerance and it is tough to feature interactions along with the applicability of binary or continuous data. Conversely, ReliefF will not discriminate among the existing redundant features and it is easy to fool the algorithm by using less number of instances [10,11]. According to Kononenko, the reliability of the probability approximation of the ReliefF algorithm can be improved through some updates and made more resilient to incomplete data. Therefore, concluding it as a problem classified under multi-class problem [21].

In recent works, scholars have proposed feature grouping to pinpoint groups with correlated features. This is an innovative method as it reduces the multi-dimensionality of large datasets. We highlight some of these efforts below.

Among one of the strategies that uses feature grouping for increasing the efficiency of feature search is called predominant group based variable neighborhood search (PGVNS) García-Torres et al. [22]. PGVNS uses approximate Markov blanket and a predominant feature. García-Torres et al. [22] also introduced the concept of predominant groups and argued in favor of a heuristic strategy called GreedyPGG that group input space. While conducting the experiment they used synthetic and real datasets obtained from the microarray and text-mining domains. The results were compared with fast correlation based filter (FCBF) [3], fast clustering-based feature-selection algorithm (FAST) [23], and CVNS [22] which are the three popular algorithms on feature selection.

Gangurde [24] and Gangurde and Metre [25] have argued in favor of a clustering concept that gives feature selection to handle the issue of dimensionality reduction in big data. A minimum spanning tree is used to create a cluster formation therefore reducing the computational complexity of feature selection. However, the study primarily deals with the reduction of irrelevant features and graph clustering.

Lei Yu and Huan Liu [3] proposed a hybrid FCBF to find the most appropriate optimal discriminative feature subset by trying to remove redundancy in features. Song et al. [23] has proposed FAST for multidimensional data. The algorithm is a little different because it operates in two stages. The first stage divides the features into clusters using graph theory and the second stage selects the most informative features that are closely related to the target class in each cluster to create a subset of final features.

*2.1.1.2. Wrapper methods.* They use a subset of features to train models. Based on a previously generated model, features are added or removed from the selected subset. The problem is thus essen-


**Fig. 3.** The process for a filter method.

**Table 1**
Categories of information theory based methods.

| Refs. | Information method | Description |
|---|---|---|
| [12] | Mutual information maximization (or information gain) | Mutual information maximization (also known as information gain) feature importance level by its correlation with a class label. The assumption of this method is that in the event of a feature having strong correlations with a class label, it can be used to accomplish good classification performance. |
| [13] | Mutual information feature selection (MIFS) | MIFS was introduced to resolve the limitation of mutual information maximization. It can take into consideration feature relevance and feature redundancy at the same time during feature selection phase. |
| [8] | Minimum redundancy maximum relevance (mRMR) | To reduce the effect of feature redundancy, mRMR is used to select features that have a high correlation with the class (output) and low correlations among themselves. |
| [14] | Conditional infomax feature extraction | Conditional infomax feature extraction was introduced to resolve the gaps in both MIFS and mRMR, which both consider feature relevance and feature redundancy at the same time.This method assumes that given the class labels if feature redundancy is stronger than intra-feature redundancy then there is a negative effect on feature selection. |
| [15] | Joint mutual information | Since MIFS and mRMR are useful in lowering feature redundancy during the process of feature selection, this alternative method known as joint mutual information was recommended to increase the sharing of complementary information between a new unselected feature and the selected feature when the class labels are given. |
| [16] | Conditional mutual information maximization (CMIM) | In CMIM, features are iteratively selected to enhance the sharing of mutual information with class labels when the selected features are given. In other words, CMIM does not select the feature that is most similar to the previously selected ones, even though the predictive power of that feature for the class labels would be strong. |
| [17] | Informative fragments | The intuition behind informative fragments is that adding a new feature should maximize the value of conditional information that the new feature and the existing features share rather than the information that the features and the class share. |
| [18] | Interaction capping | Interaction capping is similar to CMIM, but instead of restricting the formula, interaction capping is non-negative. |
| [19] | Double input symmetrical relevance | Another type of information theory based method known as double input symmetrical relevance takes advantage of normalization approaches to normalize mutually exclusive information. |
| [9] | Fast correlation based filtering (Yu and Liu, 2003) | This filtering method takes advantage of feature-feature and feature-class correlations at the same time, using feature selection methods that cannot be turned into a unified conditional likelihood maximization framework easily. |

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

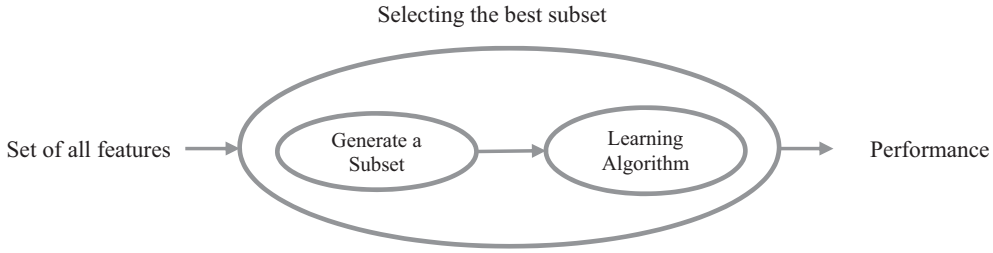


**Fig. 4.** The process for a wrapper method.

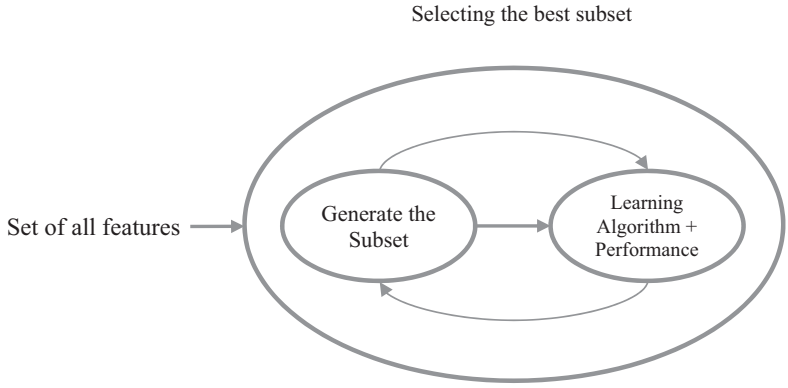


**Fig. 5.** The process for an embedded method.

tially reduced to a search problem as shown in Fig. 4. The only limitation is that the method is computationally expensive. Some examples available in feature selection are forward feature selection, backward feature elimination, and recursive feature elimination. The recursive feature elimination algorithm, is an example from this category [26].

Recursive feature elimination [27] selects features by selecting smaller sets recursively according to the features. The first step is to train an estimator from an initial set of features. This is to develop a deep learning on the importance of each feature. This process is conducted recursively and pruned till the desired number of features is achieved.

*2.1.1.3. Embedded methods.* These methods benefit from qualities of filter and wrapper methods combined. They are implemented using algorithms with inbuilt feature selection methods. They are based on learning about which feature contributes the most to the accuracy of the model as it is being created as shown in Fig. 5. Embedded methods have three types: pruning methods, models with inbuilt mechanisms for feature selection and regularization models.

Pruning methods begins by using all the available features to train a model. This step is followed by an attempt to eliminate the features by setting the value as 0 of corresponding coefficients without reducing the performance. These methods use models such as recursive feature elimination with a support vector machine (SVM) [27] which is a supervised machine learning algorithm that can be used for both classification and regression challenges.

Models with inbuilt mechanisms for feature selection include ID3 [27] and C4.5 [27]. The ID3 [27] iterative dichotomizer was the first of three decision tree implementations that Ross Quinlan developed. ID3 builds a decision tree for the given data in a top down fashion starting from a set of objects. C4.5 [27] is an improved version of Quinlan's earlier ID3 algorithm and is used to generate a classification decision tree from a set of training data (in the same way as in ID3) using the concept of information entropy.

Regularization models rely mostly on objective functions to reduce fitting errors to the lowest. IT also aims to force the coefficients to be small and potentially reaching to 0 in the meantime. Due to the good performance of regularization models, researchers have made more efforts in this area. Famous algorithms from this category include lasso [28,29] and elastic net [30].

Lasso [28,29] is method of regression analysis performing both the tasks of selecting a variable and regularizing. This improves the prediction accuracy and interpretability of the statistical model. Robert Tibshirani [28] introduced this method, which is based on Leo Breiman's nonnegative garrote.

Elastic net regularization [30] is an improved version of lasso [28,29]. It improves the performance of regression analysis models of Lasso by penalizing for additional regression in case there are more predictors than the sample size. This leads to improvements in prediction accuracy by allowing the methods to select only the strongly correlated variables.

*2.1.2. Structured features*

This section provides a review of feature selection algorithms for structured features. These features are treated like groups that have some regulatory relationships. These structural features include graph, group and tree structures [31].

*2.1.2.1. Graph structure.* A graph is a set of objects in which some pairs of objects are connected by links. Let $\mathcal{G} = (N, E)$ be a given graph where $N = (1, 2, \cdots .., m)$ is a set of nodes and a set of edges $E$. Node $i$ is equivalent to the $i$th feature and $A \in \mathbb{R}^{mxm}$ is used to donate the adjacent matrix of $\mathcal{G}$. Thus, the nodes are representative of the features and the edges represent the relationships between those features [31]. A real application of this category is natural language processing. An instance of this is WordNet. It could indicate the words that are synonyms or antonyms. There is evidence in biological studies that genes work in groups based on their bio-

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

logical functions. Some regulatory relationships have been found among those genes. Three typical algorithms are Laplacian lasso [32], graph-guided fused lasso (GFLasso) [33] and GOSCAR [34].

In a Laplacian lasso [32] features show graph structures. When two features are connected by an edge, chances are that they will be selected together. Therefore, they will show matching feature coefficients. This can be achieved via a graph lasso by adding a graph regularization to the feature graphs on the basis of the lasso method.

Graph-guided fused lasso (GFLasso) [33] is also a lasso variant. It was created to solve the limitations found in the original technique. GFLasso considers positive and negative feature correlations combined explicitly. The limiting factor for GFLasso is the use of pairwise sample correlations for measuring feature dependencies. It is a choice that leads to an added estimation bias. In a small sample size, GFLasso restricts the correct estimation of feature dependencies.

GOSCAR [34] was created to resolve the problems encountered in GFLasso [33] by forcing pairwise feature coefficients to be equal if they were connected over the feature graph.

*2.1.2.2. Group structure.* Group structure is about extracting highly informative subgraphs from a set of graphs. However, some criterion of filtering must be applied. Frequency of sub-graph is a commonly used method. An application of this category in real world can be found in speed and signal processing. Here, groups can represent the various frequency bands. Two typical algorithms are group lasso [35] and sparse group lasso [36].

Group lasso [35] provides for a combined selection of covariates as a single unit. In this case, it proves quite beneficial. One of the applications of this technique is in performing group selections or selecting group subsets. If a group is chosen, it means that all the contained features are selected as well.

Sparse group lasso [36] has the added ability to choose groups and features in the selected groups in parallel.

*2.1.2.3. Tree structure.* In a tree structure, the features are used to simulate a hierarchical tree with a root value and subtrees (children of parent nodes). It is represented as a set of linked nodes. A real application of this category is in image processing. In image processing, a tree structure is used to represent the pixels from an image with a face in it. The parent node holds the information of series of child nodes of the image describing spatial locality. Genes and proteins in biological studies can form a certain tree structure according to hierarchy.

The typical algorithm in this structure is a guided tree group lasso [37]. It was proposed for handling feature selection represented in the form of an index tree. In a tree-guided group lasso, the structure of the features can be shown as a tree and the leaf nodes are the features. The internal nodes represents the group of features in a way that each internal node is taken as a root of a subtree and all the features that are grouped are the leaf nodes. Every internal node is assigned a weight and height of that subtree which indicates the tightness of features of that subtree.

## 2.2. Streaming feature selection

A preliminary distinction is needed between streaming data and streaming features. For streaming data, the total number of features is fixed. Also, candidate instances in streaming data are generated dynamically if the size of the instances is unknown. On the other hand, streaming features are the opposite case since the number of instances is fixed. However, the candidate features are generated dynamically if the size of the features is unknown. Streaming feature selection has practical significance in many applications. For example, users of the famous microblogging website Twitter produce more than 250 million tweets per day, including many new words and abbreviations (i.e., features). In the case of tweets, performing feature selection is not recommended due to longer wait time until all the features are generated. Therefore, the use of streaming feature selection is preferred. Fig. 6 presents a basic framework for this method.

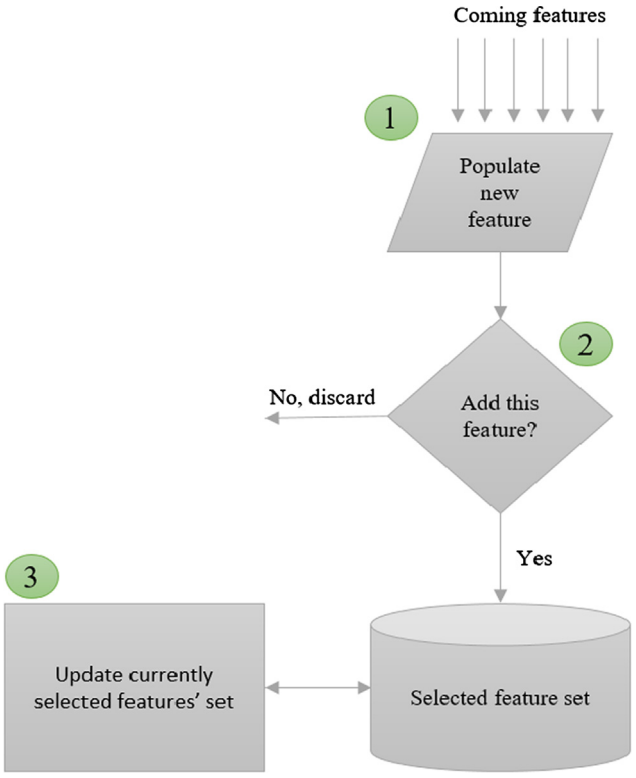


**Fig. 6.** General framework for streaming feature selection.

Step 1: Populate a new feature from the feature stream.
Step 2: Determine whether adding the new feature to the selected feature set is needed.
Step 3: Update the exiting feature set.
Step 4: Repeat Steps 1 through 3.

The algorithm could have diverse implementations for Steps 2 and 3. In some studies [38,39,40,41], Step 3 is considered an optional step in which only some of the streaming feature selection algorithm from Step 2 is implemented.

The benefit of this framework selection is in its ability to find optimal subset. This framework avoids implicitly handling feature redundancy and efficiently eliminates features that are not required by explicitly managing redundancy found in the features [3].

### 2.2.1. Single feature selection

IBM [42] defined "big-data analytics" as the use of techniques that can handle datasets from large and diverse backgrounds and multiple types. It does not matter whether it is structured and unstructured or streaming and varies according to sizes. Performing feature selection to lower data dimensionality is a desired phase in big-data analytics. This phase comes before prediction.

Grafting [38] was considered as the first attempt towards streaming feature selection. It was proposed in 2003 by Perkins and Theiler. Grafting is a popular framework for streaming feature selection and regarded as a general technique for application in a

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

variety of parameterized models using a weight vector $\boldsymbol{w}$ that is subject to $\ell 1$ regularization. The variables in the proposed algorithms are considered one at a time. The weights are re-optimized according to the available set of variables. The tasks in Perkins and Theiler's study were to select the feature subset and return the corresponding model for every unit time step. According to Perkins and Theiler, there were uncertainties in the performance of feature selection methods in this situation. They provided an alternative method known as grafting which was stage-wise technique for gradient descent.

In 2006, Zhou Jing et al. [40] proposed alpha investing, another of the earliest representative online feature selection approaches (along with grafting [38]). Alpha investing or $\alpha$ investing used p values rather than information theory. In the case of a p-value linked with t-statistic, it is the probability that coefficients of observed sizes can be estimated through chance, even in the event of the true coefficient being zero.

The aim behind alpha investing was to control the threshold during feature selection. This was made possible by selecting new features in the model. Alpha was "invested" thereby increasing the wealth and threshold and allowing for a slight increase in inclusion of incorrect features in future. In every instance when a feature is tested and determined to be insignificant, wealth is "spent" which reduces the threshold [31]. In the case of alpha investing method, it sequentially acknowledges newer features for feeding into a predictive model and modeling the set of candidate features in the form of a dynamically generated stream. One of the benefits of using alpha investing is its ability to handle feature sets of unknown sizes even up to infinity. The use of linear and logistic regression to dynamically adjust the reduction threshold for errors is favored such that the predictive model needs to evaluate a new feature for inclusion for each instance.

In another study Xindong Wu et al. [39] uses information theory to find answer to streaming feature selection by utilizing Markov blanket concept. In earlier studies, Xindong Wu et al. developed a framework that used feature relevance and a new algorithm called as OSFS along with its novel adaptation called as Fast-OSFS. According to the published definitions in the study, the features could be classified into one of these four categories: irrelevant features, redundant features, weakly relevant but non-redundant features and strongly relevant features. Thus, OSFS finds its application in online selection for features that are non-redundant and strongly relevant using two step method. The first step is analysis of its online relevance and second is online redundancy analysis. Furthermore, Xindong Wu et al. described the working of a Fast-OSFS algorithm that improves the efficiency of OSFS. The concept behind Fast-OSFS is the breakup of online redundancy analysis into two steps a) inner-redundancy analysis and b) outer-redundancy analysis. Additionally, the same authors published an updated study [43] in which they introduced an efficient Fast-OSFS algorithm that improved the performance of streaming feature selection. The algorithm proposed in this study was evaluated on a large scale using multidimensional datasets.

Kui Yu et al. [44] proposed another approach known as scalable and accurate online approach (SAOLA) for handling multidimensional datasets feature selection sequentially. SAOLA is based on a theoretical analysis and derived it from a low bound of correlations between features for pairwise comparisons. It was followed by a set of pairwise online comparisons for maintaining the parsimonious online model over longer durations.

Eskandari and Javidi [41] proposed a new algorithm called OS-NRRSAR-SA algorithm to resolve OSFS from the rough sets (RS) perspective. This algorithm adopts classical concept of RS based feature significance to reduce non-relevant features. Eskandari and Javidi claimed that the primary advantage of the algorithm was that it did not need prior domain knowledge concerning the feature space making it a viable alternative for true OSFS scenarios.

Wang et al. [45] proposed the dimension incremental algorithm for reduction computation (DIA-RED). This algorithm maintained the RS-based entropy value of the currently selected subsets and updated that value whenever new conditional features were added. While DIA-RED is capable of handling streaming scenarios despite having limited or no knowledge of the feature space, it can manage with the information contained in the lower approximation of a set and avoid using information contained in the boundary region. Therefore, real-value datasets cannot benefit from this algorithm. Also, DIA-RED algorithm does not possess an effective mechanism that eliminates redundant attributes which leads to the generation of large subsets during feature streaming. This is a prime reason for ineffective partitioning and at the time of calculating RS approximations. Therefore, the algorithm falls short of its expectations in handling most real-world datasets.

Gangurde [24] and Gangurde and Metre [25] proposed a novel clustering concept to manage big data dimensionality reduction problem. A minimum spanning tree was used to reduce the complexity in calculating feature selection and obtain a formatting of clusters. However, this concept's work scope is limited to dimensional reduction.

Javidi and Eskandari [46] have proposed a method that employs significance analysis concept in the theory of rough sets for controlling unknown feature space in SFS problems. The primary motivation for their consideration was that RS-based mining of data hardly used any domain knowledge besides the datasets that were provided. The algorithm was evaluated using several multidimensional datasets for its compactness, running time and classification accuracy.

Tommasel and Godoy [47] presented an online feature selection method for multidimensional data that is dependent on the combination of social and contextual information. The goal of their work was classifying short texts that are generated simultaneously in social networks.

#### 2.2.2. Group feature selection

Xindong Wu et al. [48] proposes group feature selection with streaming feature (GFSSF) at both levels – individual and group as a feature stream instead of predefined feature set. Xindong Wu et al. also illustrated the GFSSF algorithm, which is segregated into two distinct levels of selection. The first one at the feature level and second at the group level is based on the tenets of information theory. Features from the same group are processed in the case of feature level selection. Redundancy analysis is used for selecting the best feature subset from the features that have arrived so far. In contrast, a set of feature groups were reviewed to cover the uncertainty to a large extent in the class labels at a minimum cost during the group level selection phase. Later on, this method finds a subset of features that seem relevant and are sparse in both individual and group feature levels. In the work done till date, single features are being targeted primarily and group features are left unaddressed. Information theory is being used only for recognizing irrelevant features.

In 2015, Kui Yu et al. [49] extended SAOLA, their previous method [44] to handle a type of online streaming group feature selection and called this group-SAOLA. The new group-SAOLA algorithm could maintain an online set of feature groups that is sparse at the group feature level as well as individual feature levels at the same time. For the group level, Kui Yu et al. claimed that the group-SAOLA algorithm, while online could generate a set of feature groups that is sparse both between groups and within each group. This would maximize the methods predictive performance in classification.

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

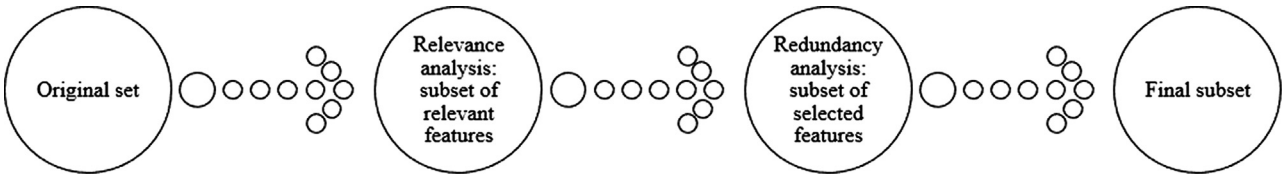


**Fig. 7.** Relevancy and redundancy evaluation.

Jing Wang et al. [50,51] tried to handle both single and group streaming feature selection by introducing an online group feature selection (OGFS) algorithm for image classification and face verification. Jing Wang et al. divided online group feature selection into online intragroup selection and intergroup selection. They designed two criteria for intragroup selection based on spectral analysis and introduced the lasso algorithm to reduce the redundancy in intergroup selections.

## 3. Attribute evaluation relevancy and feature redundancy

The objective of streaming feature selection is to choose (while online) the subset of features from a multidimensional data which leads to an increase in accuracy and robustness. This can be achieved by removing the features that are irrelevant and redundant.

In streaming feature selection, the optimal, final feature subset should be relevant to the class and should not be redundant with any other existing features to increase robustness. Thus, we can determine two feature testing stages that would be used in selecting the final and most optimal subset. Thus, we can use relevance analysis which can determine the subset of relevant features while removing the ones that are irrelevant. Similarly, we can use redundancy analysis to remove redundant features and leave a final subset as depicted in Fig. 7.

### 3.1. Relevance analysis

In relevance analysis, a single feature's relevance to the selected class is evaluated. The criterion for relevance decides how effectively a variable can distinguish between a class or a feature and a class [52].

$$Relevance\ Test\ (X,Y) = how\ useful\ X\ is\ for\ predicting\ Y \tag{5}$$

In feature relevance, a feature is evaluated individually and discarded if it fails to reach a chosen cutoff point. Table 2 is a comparison of some existing algorithms that are used to evaluate a feature's relevance to a class as part of a classification problem.

Chi-squared [53] is used to calculate the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.

Gain ratio (GR) [53] is used to evaluate the worth of an attribute by measuring the gain ratio with respect to the class. The gain ratio is given by

$$GR = \frac{H(class) - H(class|attribute)}{H(attribute)} \tag{6}$$

where H is the entropy.

Information gain (IG) [53] is used to evaluate an attribute's worth by measuring the information gain with respect to the class. The information gain is given by

$$IG = H(class) - H(class|attribute). \tag{7}$$

ReliefF [53] is used to evaluate an attribute's worth by sampling an instance several times and taking the value of the given attribute for the nearest instance of the same class and of a different class. The formula for ReliefF is

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

where

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

Significance [53] is used to evaluate an attribute's worth by computing its probabilistic significance as a two-way function (both attribute-class and class-attribute associations).

Symmetrical uncertainty (SU) [53] is used to evaluate an attribute's worth by measuring its symmetrical uncertainty with respect to a class; it is given by

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

### 3.2. Redundancy analysis

Redundancy analysis is used to evaluate features' similarity. In other words, it is used to answer the question: How much can adding a new feature improve the accuracy of a machine-learning model?

Yu and Liu (2004) [12] defined a feature as predominant (both relevant and non-redundant) if it does not have an approximate Markov blanket in the current set. For two relevant features, $F_i and F_j (i \neq j)$, $F_j$ forms an approximate Markov blanket for $F_i$ if

$$SU_{j,c} \geq SU_{i,c} and SU_{i,j} \geq SU_{i,c}, \tag{11}$$

where $SU_{j,c}$ is a correlation between any feature and class and $SU_{i,j}$ is a correlation between any pair of features, $F_i$ if and $F_j (i \neq j)$.

Correlation-based feature selection (CFS) [54,53] is a popular technique for ranking the relevance of features by measuring the correlations between features and classes and between features and other features.

Given $k$ features and $C$ classes, CFS defines the relevance of the feature subset using Pearson's correlation equation:

$$Merit_s = \frac{k r_{kc}}{\sqrt{k + (k-1) r_{kk}}}, \tag{12}$$

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. Normally, CFS adds or deletes one feature at a time using forward or backward selection. However, this research used sequential forward floating search (SFFS) as the search direction.

Sequential forward floating search (SFFS) [53,55] is a classic heuristic searching method. It is a variation of bidirectional search and sequential forward search and is thus part of the dominant direction of forward search. SFFS removes features (backward elimination) after adding features (forward selection).

The numbers of forward and backward steps are not fixed and can be controlled dynamically depending on the criterion of the selected subset. This eliminates the need for parameter setting.

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

**Table 2**
Properties of the experiments on streaming feature selection.

| Algorithm | Properties |
|---|---|
| Grafting[38] | • *Single or group feature selection:* single.<br>• *Compared with which algorithms*: none.<br>• *Datasets:* Two synthetic datasets (A and B) and Pima Indian Diabetes dataset (Blake & Merz, 1998) [69].<br>• *Classifiers:* Combination of the speed of filters and the accuracy of the wrapper.<br>• *Environment:* Not mentioned. |
| Alpha investing [40] | • *Single or group feature selection:* single.<br>• *Compared with which algorithms*: none. The appraisal was limited to the accuracy of the whole dataset.<br>• *Datasets:* Seven datasets from the UCI [57] repository: cleve, internet, ionosphere, spam, spect, wdbc, and wpbc. Three datasets on gene expression: aml, ha, and hung.<br>• *Classifiers:* C4.5, fivefold cross-validation.<br>• *Environment:* Not mentioned. |
| OSFS and Fast-OSFS [39] | • *Single or group feature selection:* single.<br>• *Compared with which algorithms*: Grafting and alpha investing [71].<br>• *Datasets:* Ten public challenge datasets: lymphoma, ovarian-cancer, breast-cancer, hiva, nova, manelon, arcene, dexter, dorohthea and sido0.<br>• *Classifiers:* k-nn, decision tree (J48) and random forest (Spider 2010).<br>• *Environment:* Windows XP, a 2.6 GHz CPU, and 2 GB memory. |
| SAOLA [44] | • *Single or group feature selection:* single.<br>• *Compared with which algorithms*: Fast-OSFS [43], alpha investing [71], OFS [72], FCBF [3], as well as two state-of-the-art algorithms, SPSF-LAR [73] and GDM [74].<br>• *Datasets:* Ten high-dimensional datasets: two public microarray datasets (lung cancer and leukemia), two text-categorization datasets (ohsumed and apcj etiology), two biomedical datasets (hiva and breast cancer), three NIPS 2003 (dexter, madelon, and dorothea) and the thrombin dataset, which was chosen from KDD Cup 2001. Four extremely high-dimensional datasets from the Libsvm dataset website: news20, url1, webspam, and kdd2010.<br>• *Classifiers:* KNN and J48, which are provided in the Spider Toolbox2 [75].<br>• *Environment:* Intel i7-2600 with a 3.4 GHz CPU and 24 GB of memory. |
| OS-NRRSAR-SA [41] | • *Single or group feature selection:* single.<br>• *Compared with which algorithms*: Grafting, information investing [71], fast-OSFS, and DIA-RED.<br>• *Datasets:* Fourteen high-dimensional datasets: The dorothea, arcene, dexter, and madelon datasets from the NIPS 2003 Feature-Selection Challenge. The nova, sylva, and hiva datasets from the WCCI 2006 Performance Prediction Challenges. The sido0 and cina0 datasets from the WCCI 2008 Causation and Prediction Challenges. The arrhythmia and multiple features datasets from the UCI Machine Learning Repository. Three synthetic datasets: tm1, tm2, and tm3.<br>• *Classifiers:* J48, JRip, Naive Bayes, and kernel SVM with the RBF kernel function.<br>• *Environment:* Dell workstation with Windows 7, 2 GB of memory, and a 2.4 GHz CPU. |
| DIA-RED [45] | • *Single or group feature selection:* single.<br>• *Compared with which algorithms*: None.<br>• *Datasets:* Six datasets from the UCI [57] Machine-Learning Repository: Backup-large, Dermatology, Splice, Kr-vs-kp, Mushroom, and Ticdata2000.<br>• *Classifiers:* information entropy used to measure the uncertainty of a dataset: complementary entropy [76], combination entropy [77], and Shannon's entropy [78].<br>• *Environment:* Windows 7, an Intel Core i7-2600 CPU (2.66 GHz), and 4 GB of memory. |
| GFSSF [48] | • *Single or group feature selection:* singleand Group selection.<br>• *Compared with which algorithms*: Five standard feature-selection algorithms: MIFS [13], joint mutual information [79], mRMR [8], ReliefF [20], and lasso [28]. Four streaming-feature-selection algorithms: grafting [38], $\alpha$ investing [40], OSFS [39], and Fast-OSFS [39]. One group-feature-selection algorithm: group lasso [35].<br>• *Datasets:* Five UCI [57] benchmark datasets: WDBC, WPBC, IONOSPHERE, SPECTF, and ARRHYTHMIA. Five challenge datasets with relatively high feature dimensions) downloaded from http://mldata.org/repository): DLBCL (7,130 features; 77 instances), LUNG (7,130 features; 96 instances), CNS (7,130 features; 96 instances), ARCENE (10,000 features; 100 instances), and OVARIAN (15,155 features; 253 instances). Five UCI [57] datasets with generated group structures: HILL-VALLEY (400 features; 606 instances), NORTHIX (800 features; 115 instances), MADELON (2,000 features; 4,400 instances), ISOLET (2,468 features; 7,797 instances), and MULTI-FEATURES (2,567 features; 2,000 instances).<br>• *Classifiers:* NaiveBayes [80], k-NN [81], C4.5 [82], and Randomforest [83].<br>• *Environment:* Windows 7, a 3.33 GHz dual-core CPU, and 4 GB of memory. |
| group-SAOLA [49] | • *Single or group feature selection:* group<br>• *Compared with which algorithms*: Three state-of-the-art online-feature-selection methods:<br>• Fast-OSFS [43], alpha investing [40], and OFS [43]. Three batch methods: one well-established algorithm (FCBF) [3], and two state-of-the-art algorithms (SPSF-LAR [73] and GDM [74]).<br>• *Datasets:* Ten high-dimensional datasets: madelon, hiva, leukemia, lung-cancer, ohsumed, breast-cancer, dexter, apcj-etiology, dorothea, and thrombin. Four extremely high-dimensional datasets: news20, url1, webspam, and kdd2010.<br>• *Classifiers:* KNN and J48, which are provided in the Spider Toolbox [75], and SVM.<br>• *Environment:* Intel i7-2600, a 3.4 GHz CPU, and 24 GB of memory. |
| OGFS [50 51] | • *Single or group feature selection:* singleand group.<br>• *Compared with which algorithms*: Grafting, alpha investing, and OSFS.<br>• *Datasets:* Eight datasets from UCI: Wdbc, Ionosphere, Spectf, Spambase, Colon, Prostate, Leukemia and Lungcancer. Three datasets from the real world: Soccer, Flower-17, and 15 Scenes.<br>• *Classifiers:* appraisal was based on number of the selected features.<br>• *Environment:* Windows XP, a 2.5 GHz CPU, and 2 GB of memory. |

## 4. Challenges of using streaming feature selection big data analytics

As mentioned earlier, big data has created challenges that are yet to be addressed by traditional machine learning practices. This has led to the adoption of methodologies capable of handling increasingly large data volumes. To overcome this challenge, improving streaming feature selection is necessary to introduce better and more efficient approaches for handling extremely high dimensionality of big data. In this section, we highlight some of

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

these challenges which could be considered hot topics in streaming feature selection.

### 4.1. The extremely high dimensionality of big data

In big data, feature selection is generally considered a strong technique for selecting a subset of relevant features and reducing the dimensionality of extremely high dimensional data. The streaming of big data is more challenging as the number of unknown features is high. Sometimes, it is reaching levels that render existing feature-selection methods obsolete.

Today, in the age of big data, social media is considered the main source of streaming data. Big data is extremely large and growing as a fast pace. In short, big data can be so large and complex that none of the traditional data management tools can store or process it efficiently. Feature selection is generally considered a strong technique for preferring a subset of relevant features and lowering the multidimensionality of data. However, in the case of streaming big data, streaming feature selection is more challenging because of the large number of unknown features.

Big data can be characterized by the 5 V's [56]:

(1) Volume – The quantity of generated and stored data determines its value and potential insight that can be drawn from it and also if it can be considered as big data.
(2) Variety – The type and nature of the data helps analysts to effectively use the resulting insight.
(3) Velocity – Means the rate at which data is created and processed to fulfill the needs of growth and development challenges.
(4) Variability – Inconsistency can hamper processes that is meant to manage a dataset.
(5) Veracity – Captured data can vary greatly in terms of quality thus affecting the accuracy of the analysis.

The UC Irvine Machine Learning Repository (UCI) is a collection of databases, domain theories, and data generators used by researchers devoted to machine learning using empirical analysis of machine learning algorithms [57]. This repository started around 1987 when the maximum dimensionality of data in a multivariate dataset was 8,124 instances and 22 features. A thyroid disease dataset in the same time had 7,200 instances and 21 features. However, the number of instances and features had increased to millions by the end of 2017. In the case of the causal-discovery KASANDR dataset which has 17,764,280 instances and 2,158,859 features.

In this scenario, the methods that experience the greatest challenges are feature selection and streaming feature selection. For example, Zhai et al. [58] needed more than 24 h of computational effort, using state-of-the-art feature selectors (SVM/recursive feature elimination and mRMR) to analyze data for a psoriasis single-nucleotide polymorphism dataset composed of only half a million features. Moreover, many modern feature selection methods are based on algorithm designs for computing pairwise correlations. In the case of a million features, the machine must be capable of handling a trillion correlations effectively which poses a significant issue for machine learning researchers [59].

### 4.2. Scalability

Scalability is defined as "the impact of an increase in the size of the training set on the computational performance of an algorithm in terms of accuracy, training time and allocated memory" [59]. Today, with the exposure of big data, those who use traditional methods are struggling to cope with the extreme high-dimensionality of big data as they attempt to extract satisfactory results in a reasonable time.

The extremely multidimensional big data is unable to load in the memory in a single data scan. Therefore, it is challenging to get a score of feature relevance without considering sufficient density surrounding every sample.

Considering the available approaches for large-scale selection of features there are two prominent phases. The first phase measures the relevance of individual features and then ranks them according to their relevance values. The values that show the highest relevance only are used for input in the second phase. However, this approach presents the limitations that it may remove the features that are lowly ranked or even consider its interactions with other features [60].

### 4.3. Stability

The stability of feature selection is defined [61] as the sensitivity that the selection process has to data perturbation in the training set. Stability quantifies how a training set affects feature selection. The feature selection algorithm for classification is measured using classification accuracy. Thus, the stability of any algorithm is a critical factor when developing feature selection.

Alelyaniet al. [62] has presented and argued for some characteristics of data that may play a vital role in stabilizing the algorithm. They are dimensionality (m), size of sample (n) and data distribution across folds. Therefore, the stability issue tends to be dependent on data.

A measure of stability requires a similarity measure for feature preferences. Researchers have proposed various stability measures to evaluate robustness [63,64,59]. These measures can be placed in three categories:

Category 1: A weight or score is assigned to each feature, indicating its importance.
For a vector of features$f = (f_1, f_2, \cdots, f_m)$, this category produces a feature set as follows:

$$weighting - scoring : w = (w_1, w_2, \cdots, w_m), w \in W \subseteq R^m.$$

Category 2: This is a simplification of the first category; ranks are assigned to features instead of weights.
For a vector of features $f = (f_1, f_2, \cdots, f_m)$, this category produces a feature set as: follows

$$ranking : r = (r_1, r_2, \cdots, r_m), 1 \leq r_i \leq m.$$

Category 3: These measures consist of sets of selected features for which no weighting or ranking is considered.
For a vector of features $f = (f_1, f_2, \cdots, f_m)$, this category produces a feature set as follows:
$subsetoffeatures : s = (s_1, s_2, \cdots, s_m), s_i \in \{0, 1\}$, w *i*th 0 indicating absence of a feature and 1 for presence.

For streaming feature selection, the challenge lies with the unknown features. Selecting the most informative features from among the current features challenges the stability of any proposed algorithm. As a result, updating the selected subset also challenges the robustness of the algorithm.

### 4.4. Sustainability

The volume of data increases by 90% of the data in the world which has been created in the last two years. Data is generated from different resources like mobile phones, sensors, and social

media in continuous manner. This data is expected to grow in the near future dramatically. The data revolution would pose a challenge for resources sustainability. Sustainability means the ability to optimize resource usage. Thus, finding a new way to reduce the extremely high dimensionality of big data would result in big savings in the analytic process. It is clear from previous examples that feature selection would be considered as the first option to reduce the dimensionality of any data. This would allow picking informative features only rather than considering all of them. Consequently, the streaming feature selection would efficiently resolve the sustainability issue of streaming big data. Recently [65–68] highlight has been the greening issue of big data in the big data analytics. The process of big data analytics are accompanied with lot of computing workloads, which is time consuming at the same time energy and resource inefficient.

## 5. Discussion and comparison

This section discusses streaming feature selection algorithms and examples that we demonstrated in Section 2. It also compares these algorithms based on the big-data challenges that were discussed in Section 4. Table 2 is a comparison of the reviewed streaming feature selection algorithms. Note that these algorithms use either single feature selection, group feature selection or both. Table 2 presents a comparison of the algorithms based on the feature selection type, how they compare to other online feature selection methods, datasets and classifiers that were used to report the classification accuracy and the environment of the experiment.

As mentioned earlier, grafting [38] and alpha investing [40] are two of the earliest methods for online feature selection. Grafting algorithm is based on a stage wise gradient descent approach for streaming feature selection. However grafting has some limitations. It can obtain a global optimum with respect to features included in the model, it is not optimal as some features are dropped during online selection. Besides, the gradient retesting over all the selected features greatly increases the total time cost. Thus, tuning a good value for the important regularization parameter $\lambda$ requires the information of the global feature space. Similarly, Alpha-investing does not reevaluate the selected features, it hence performs efficiently, but it is probably performing ineffectively in the subsequent feature selection for never evaluating the redundancy of selected features [50]. These limitations for high-dimensional data were recognized at the time they were created. For example, the Pima Indian Diabetes dataset (Blake & Merz, 1998) [69] found that grafting has 768 instances and eight attributes. Likewise, alpha investing used a spam dataset [70], which had 4,601 instances and 57 attributes. Jing Wang et al. in their OGFS experiments [50,51], used the method of grafting for performing feature selection using the gradient descent technique which can be quite effective in pixel classification.

However, this method still requires a global feature space for defining key parameters during the selection of features. Therefore, it presents limitations in cases where feature stream is infinite or has an unknown size. Also, alpha investing calculates each new feature using a p-value that is from a regression model. In case where the p-value of a new feature goes to a certain limit or threshold (known as $\alpha$), the algorithm selects the feature. Therefore, alpha investing never discards a feature once it has been selected.

Currently, researchers focusing on OSFS, Fast-OSFS [39], SAOLA [44] and group-SAOLA [49] are taking the lead in this area. Following their work history, these researchers started with the OSFS [39], Fast-OSFS [39], and SAOLA [44] to handle single feature selection. After that, they introduced group-SAOLA [49] to handle both single and group feature selection. In OSFS [39] features are selected according to the relevance they have online and whether they are redundant or not. Based on the relevance it holds to the class label, input features are labeled as strongly relevant, weakly relevant or non-relevant. Online relevance analysis provides for the features that are relevant. Markov blankets are used to remove redundant features. In the case of OSFS, every time a method includes a new feature, it is necessary to reanalyze the redundancy of all selected features. To improve the performance of conducting redundancy analysis, a fast-version of OSFS is proposed known as Fast-OSFS [39]. The Fast-OSFS experiments uses eight UCI [57] benchmark databases. Researchers compared Fast-OSFS's performance with those of grafting and alpha investing [71] algorithms using the k-nearest neighbor (or k-nn), decision tree, and random forest datasets. SAOLA managed to handle a multidimensional dataset which allowed it to overcome the two challenges of big data – scalability and extreme multidimensionality.

Another attempt to resolve the problem of streaming feature selection is OS-NRRSAR-SA [41]. This method uses RS-based data mining to control unknown feature space without needing any domain knowledge. During experiments, Eskandari and Javidi compared the algorithm's performance with those of four modern algorithms (grafting, information investing [71], fast-OSFS, and DIA-RED) using 14 benchmark datasets. For these experiments, the computer had 24 GB of memory which gave this algorithm a performance benefit relative to other algorithms.

DIA-RED [45], another single feature selection algorithm was proposed to resolve the issue of streaming feature selection. In the experiments on this method, the researchers used only six datasets from UCI's [57] repository of machine learning: Backup-large, Dermatology, Splice, Kr-vs-kp, Mushroom, and Ticdata2000. However, the researchers didn't compare their method to other state-of-art streaming-feature-selection algorithms. They only

**Table 3**
Comparison of related works.

| Related work | Method | | | |
|---|---|---|---|---|
| | Feature selection | Streaming feature selection | Feature grouping | Streaming feature grouping and selection |
| Grafting [38] | √ | √ | | |
| Alpha investing [40] | √ | √ | | |
| PGVNS [22] | √ | | √ | |
| FCBF [3] | √ | | √ | |
| OSFS and Fast-OSFS [39] | √ | √ | | |
| SAOLA [44] | √ | √ | | |
| OS-NRRSAR-SA [41] | √ | √ | | |
| DIA-RED [45] | √ | | | |
| Gangurde [24] and Gangurde and Metre [25] | √ | | √ | |
| group-SAOLA [49] | √ | √ | | |
| OGFS [50,51] | √ | √ | | |

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

measured the uncertainty of the tested datasets compared to the traditional feature selection approaches.

On the other hand, GFSSF [48], group-SAOLA [49] and OGFS [50,51] were designed to handle group feature selection. The GFSSF algorithm has the edge over both group-SAOLA [49] and OGFS [50,51] according to a comparison with lasso [35] which is a group feature selection algorithm. However, in terms of big data, group-SAOLA used fewer resources such as memory. Using more resources would enhance this methods chance of prevailing in the big-data scalability challenge. Table 3 contains a comparison of some of the reviewed streaming feature selection algorithms. This comparison is based on the approach used to reduce the redundancy of the received features.

## 6. Conclusion and future work

Streaming feature selection plays an important role in the pre-processing stage of big-data mining. It also has relevance in machine-learning applications as it can reduce the extreme high-dimensionality of big data. In machine learning, streaming feature selection is generally considered a strong technique for selecting a subset of relevant features. This is because it can reduce the dimensionality in an online fashion. Therefore, streaming feature selection is considered an attractive research topic in big data.

This survey paper is intended to provide a comprehensive overview of recently developed streaming feature selection algorithms to promote research in this area. First, we introduced the background of traditional feature selection and streaming feature selection. This was followed by describing the difference between both. It was followed by an illustration of feature relevance and redundancy. Then, we highlighted some challenges of streaming feature selection in the context of big data. We also surveyed the current efforts in streaming feature selection by discussing and comparing them with the general framework.

The algorithms reviewed in this survey provides the necessary learning to suggest future research directions and to resolve the present challenges in the use of streaming feature selection for big data. The existing approaches for streaming feature selection involves testing new features one by one to select the optimal subset. This procedure does not work with the extreme high-dimensionality of big data for which more innovative approaches are needed.

Another research direction is in the stability of streaming feature selection. Big data brings challenges related to an unknown or even infinite number of features. In this context, selecting the most informative features will change the stability of any proposed algorithm.

The scalability challenge is another future direction for research into online feature-selection algorithms. Even with the significant power of existing computers, big data cannot be loaded in memory in a single data scan. It is a big challenge to obtain a relevance score for features without having sufficient density around each sample.

## References

[1] Gil Press, 6 Predictions For The $203 Billion Big Data Analytics Market, Forbes, 20-Jan-2017. [Online]. Available: https://www.forbes.com/sites/gilpress/2017/01/20/6-predictions-for-the-203-billion-big-data-analytics-market/#6b26d76c2083.
[2] G.H. John, R. Kohavi, K. Pfleger, and others, Irrelevant features and the subset selection problem, in: Machine Learning: Proceedings of the Eleventh International Conference, 1994, pp. 121–129.
[3] Yu. Lei, Huan Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (Oct. 2004) 1205–1224.
[4] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer, 2013.
[5] N.T. Longford, A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects, Biometrika 74 (4) (1987) 817–827.
[6] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, 2012.
[7] Daphne Koller, Mehran Sahami, Toward Optimal Feature Selection, 1996.
[8] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[9] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003, pp. 856–863.
[10] K. Kira, L.A. Rendell, A practical approach to feature selection, in: Proceedings of the ninth international workshop on Machine learning, 1992, pp. 249–256.
[11] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (1–2) (2003) 23–69.
[12] D.D. Lewis, Feature selection and feature extraction for text categorization, in: Proceedings of the workshop on Speech and Natural Language, 1992, pp. 212–217.
[13] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Netw. 5 (4) (1994) 537–550.
[14] D. Lin, X. Tang, Conditional infomax learning: an integrated framework for feature extraction and fusion, in: Comput Vision–ECCV 2006, 2006, pp. 68–82.
[15] H.H. Yang, J. Moody, Data visualization and feature selection: new algorithms for nongaussian data, in: Advances in Neural Information Processing Systems, 2000, pp. 687–693.
[16] F. Fleuret, Fast binary feature selection with conditional mutual information, J. Mach. Learn. Res. vol. 5 (2004) 1531–1555.
[17] M. Vidal-Naquet, S. Ullman, Object Recognition with Informative Features and Linear Classification, in: ICCV, 2003, p. 281.
[18] A. Jakulin, Machine Learning Based on Attribute Interactions, Univerza v Ljubljani, 2005.
[19] P.E. Meyer, G. Bontempi, On the use of variable complementarity for feature selection in cancer classification, in: Workshops on Applications of Evolutionary Computation, 2006, pp. 91–102.
[20] K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, in: Aaai, 1992, pp. 129–134.
[21] I. Kononenko, E. Šimec, M. Robnik-Šikonja, Overcoming the myopia of inductive learning algorithms with RELIEFF, Appl. Intell. 7 (1) (1997) 39–55.
[22] Miguel García-Torres, Francisco Gómez-Vela, Belén Melián-Batista, J. Marcos Moreno-Vega, High-dimensional feature selection via feature grouping: a variable neighborhood search approach, Inf. Sci. 326 (2016) 102–118.
[23] Qinbao Song, Jingjie Ni, Guangtao Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, IEEE Trans. Knowl. Data Eng. 25 (1) (2013) 1–14.
[24] Harshali D. Gangurde, Feature Selection using clustering approach for Big Data, Int. J. Comput. Appl. (2014).
[25] Harshali D. Gangurde, K.V. Metre, Mining of high dimensional data using feature selection, Int. J. Comput. Sci. Mob. Comput. 4 (6) (2015) 901–906.
[26] Ron Kohavi, George H. John, Wrappers for feature subset selection, Artif. Intell. 1 (1997) 273–324.
[27] Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining, Practical Machine Learning Tools and Techniques, third ed., Elsevier, Amsterdam, 2011.
[28] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Methodol. (1996) 267–288.
[29] M. Yamada, W. Jitkrittum, L. Sigal, E. Xing, M. Sugiyama, High-dimensional feature selection by feature-wise non-linear lasso. arXiv preprint, ArXiv Prepr. ArXiv12020515, 2012.
[30] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B Stat. Methodol. 67 (2) (2005) 301–320.
[31] C.C. Aggarwal, Data Classification: Algorithms and Applications, CRC Press, 2014.
[32] J. Ye, J. Liu, Sparse methods for biomedical data, ACM Sigkdd Explor. Newsl. 14 (1) (2012) 4–15.
[33] S. Kim, E.P. Xing, Statistical estimation of correlated genome associations to a quantitative trait network, PLoS Genet. 5 (8) (2009) e1000587.
[34] S. Yang, L. Yuan, Y.-C. Lai, X. Shen, P. Wonka, J. Ye, Feature grouping and selection over an undirected graph, in: Graph Embedding for Pattern Analysis, Springer, 2013, pp. 27–43.
[35] M. Yuan, Y. Lin, Model, selection and estimation in regression with grouped variables, J. R. Stat. Soc. Ser. B Stat. Methodol. 68 (1) (2006) 49–67.
[36] J. Friedman, T. Hastie, R. Tibshirani, A note on the group lasso and a sparse group lasso, ArXiv Prepr. ArXiv10010736, 2010.
[37] J. Liu, J. Ye, Moreau-Yosida regularization for grouped tree structure learning, in: Advances in Neural Information Processing Systems, 2010, pp. 1459–1467.
[38] Simon Perkins, James Theiler, Online Feature Selection using Grafting, ICML, 2003.
[39] Xindong Wu, Kui Yu, Hao Wang, and Wei Ding, Online streaming feature selection, in: 27th International Conference on Machine Learning (ICML-10), 2010, pp. 1159–1166.
[40] J. Zhou, D. Foster, R. Stine, L. Ungar, Streaming feature selection using alpha-investing, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 2005, pp. 384–393.
[41] S. Eskandari, M. Javidi, Online streaming feature selection using rough sets, Int. J. Approx. Reason. 69 (2016) 35–57.
[42] "What is Big Data Analytics?" [Online]. Available: http://www-01.ibm.com/software/data/infosphere/hadoop/what-is-big-data-analytics.html.
[43] X. Wu, K. Yu, W. Ding, H. Wang, X. Zhu, Online feature selection with streaming features, IEEE Trans. Pattern Anal. Mach. Intell. 35 (5) (2013) 1178–1192.

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001

[44] Kui Yu, Xindong Wu, Wei Ding, Jian Pei, Towards scalable and accurate online feature selection for big data, in: IEEE International Conference on Data Mining (ICDM), 2014, pp. 660–669.
[45] F. Wang, J. Liang, Y. Qian, Attribute reduction: a dimension incremental strategy, Knowl.-Based Syst. 39 (2013) 95–108.
[46] M.M. Javidi, S. Eskandari, Streamwise feature selection: a rough set method, Int. J. Mach. Learn. Cybern. 9 (4) (2018) 667–676.
[47] A. Tommasel, D. Godoy, A Social-aware online short-text feature selection technique for social media, Inf. Fusion 40 (2018) 1–17.
[48] Haiguang Li, Xindong Wu, Zhao Li, Wei Ding, Group feature selection with streaming features, in: Presented at the 13th International Conference on Data Mining, 2013, pp. 1109–1114.
[49] K. Yu, X. Wu, W. Ding, J. Pei, Scalable and accurate online feature selection for big data, ACM Trans. Knowl. Discov. Data TKDD 11 (2) (2016) 16.
[50] J. Wang et al., Online feature selection with group structure analysis, IEEE Trans. Knowl. Data Eng. 27 (11) (2015) 3029–3041.
[51] J. Wang, Z.-Q. Zhao, X. Hu, Y.-M. Cheung, M. Wang, X. Wu, Online group feature selection, in: IJCAI, 2013, pp. 1757–1763.
[52] B. Auffarth, M. López, J. Cerquides, Comparison of redundancy and relevance measures for feature selection in tissue classification of CT images, in: ICDM, 2010, pp. 248–262.
[53] R. Duangsoithong, T. Windeatt, Relevance and redundancy analysis for ensemble classifiers, Mach. Learn. Data Min. Pattern Recognit. (2009) 206–220.
[54] M.A. Hall, Correlation-based Feature Selection of Discrete and Numeric Class Machine Learning, 2000.
[55] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, Pattern Recognit. Lett. 15 (11) (1994) 1119–1125.
[56] M. Hilbert, Big data for development: a review of promises and challenges, Dev. Policy Rev. 34 (1) (2016) 135–174.
[57] UCI Machine Learning Repository: Data Sets. [Online]. Available: https://archive.ics.uci.edu/ml/datasets.html.
[58] Y. Zhai, Y.-S. Ong, I.W. Tsang, The Emerging "Big Dimensionality", IEEE Comput. Intell. Mag. 9 (3) (2014) 14–26.
[59] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Recent advances and emerging challenges of feature selection in the context of big data, Knowl.-Based Syst. 86 (2015) 33–45.
[60] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626.
[61] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: a review, in: Data Classif. Algorithms Appl., 2014, p. 37.
[62] S. Alelyani, H. Liu, L. Wang, The effect of the characteristics of the dataset on the selection stability, in: Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on, 2011, pp. 970–977.
[63] D. Dernoncourt, B. Hanczar, J.-D. Zucker, Analysis of feature selection stability on high dimension and small sample data, Comput. Stat. Data Anal. 71 (2014) 681–693.
[64] A.-C. Haury, P. Gestraud, J.-P. Vert, The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures, PloS One 6 (12) (2011) e28210.
[65] J. Wu, S. Guo, J. Li, D. Zeng, Big data meet green challenges: greening big data, IEEE Syst. J. 10 (3) (2016) 873–887.
[66] J. Wu, S. Guo, H. Huang, W. Liu, Y. Xiang, Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives, IEEE Commun. Surv. Tutor. (2018).
[67] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, Y. Yang, Big data meet cyber-physical systems: a panoramic survey, IEEE Access 6 (2018) 73603–73636.
[68] J. Wu, S. Guo, J. Li, D. Zeng, Big data meet green challenges: big data toward green applications, IEEE Syst. J. 10 (3) (2016) 888–900.
[69] Pima Indians Diabetes Data Set. [Online]. Available: https://archive.ics.uci.edu/ml/support/pima+indians+diabetes.
[70] Spambase Data Set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase.
[71] J. Zhou, D.P. Foster, R.A. Stine, L.H. Ungar, Streamwise feature selection, J. Mach. Learn. Res. 7 (2006) 1861–1885.
[72] J. Wang, P. Zhao, S.C. Hoi, R. Jin, Online feature selection and its applications, IEEE Trans. Knowl. Data Eng. (2013) 1–14.
[73] Z. Zhao, L. Wang, H. Liu, J. Ye, On similarity preserving feature selection, IEEE Trans. Knowl. Data Eng. 25 (3) (2013) 619–632.
[74] Y. Zhai, M. Tan, I. Tsang, Y.S. Ong, Discovering support and affiliated features from very high dimensions, ArXiv Prepr. ArXiv12066477, 2012.
[75] "The Spider." [Online]. Available: http://people.kyb.tuebingen.mpg.de/spider/.
[76] J. Liang, Z. Shi, "The information entropy, rough entropy and knowledge granulation in rough set theory, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 12 (01) (2004) 37–46.
[77] Y. Qian, J. Liang, Combination entropy and combination granulation in rough set theory, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 16 (02) (2008) 179–193.
[78] C.E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mob. Comput. Commun. Rev. 5 (1) (2001) 3–55.
[79] H. Yang, J. Moody, Feature selection based on joint mutual information, in: Proceedings of International ICSC Symposium on Advances in Intelligent Data Analysis, 1999, pp. 22–25.
[80] G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995, pp. 338–345.
[81] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66.
[82] J.R. Quinlan et al., Bagging, Boosting, and C4. 5, in: AAAI/IAAI, 1996, pp. 725–730.
[83] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

Please cite this article as: N. AlNuaimi, M. M. Masud, M. A. Serhani et al., Streaming feature selection algorithms for big data: A survey, Applied Computing and Informatics, https://doi.org/10.1016/j.aci.2019.01.001