Information Sciences 480 (2019) 365–380

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Ensemble feature selection using election methods and ranker clustering

Peter Drotár[a,*], Matej Gazda[b], Liberios Vokorokos[a]

[a] Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Kosice, Letna 9, Kosice 040 01, Slovakia
[b] Department of Mathematics and Theoretical Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Kosice, Letna 9, Kosice 040 01, Slovakia

ARTICLE INFO

Article history:
Received 14 November 2017
Revised 19 December 2018
Accepted 20 December 2018
Available online 24 December 2018

Keywords:
Feature selection
Ensemble
Voting
Mean shift clustering
Borda count

ABSTRACT

Feature selection (FS) has become a significant part of the data processing pipeline. Recently, ensemble FS has emerged as a new methodology that promises to improve FS robustness and performance. In this paper, we propose several ensemble FS methods built on voting aggregation schemes such as plurality vote, single transferable vote, Borda count, and novel weighted Borda count. Additionally, we present the new concept of clustering FS methods prior to building ensembles using a mean-shift clustering algorithm. The proposed methods are examined using three accuracy measures: the ability to correctly identify relevant features, FS stability, and influence on classification. The ensembles and clustered ensembles based on a weighted Borda count show very balanced performance, achieving quality results in all investigated measures and outperforming the other methods examined.

1. Introduction

We are currently experiencing the availability of huge amount of data; billions of devices—from wearable sensors to space telescopes—generate heterogeneous data. Recent estimates expect 4 – 43 Exabytes of required annual storage needs in 2025 [42] when considering only four major generators of Big Data: genomics, YouTube, Twitter, and astronomy. If we consider other major players in Big Data, such as social networks or other bioinformatics fields, the amount of data that will be generated and must be stored is even higher. Not only is the volume of data increasing, but so is the dimensionality of individual datasets; there are now more high- and ultra-high-dimensional datasets across domains. While dimensionality is rapidly growing, achieving tens or hundreds of thousands of features, sample size has not seen the same rate of increase. We frequently encounter high-dimensional datasets consisting of only hundreds (or even fewer) samples. These conditions make it difficult to create effective mathematical models for data analysis.

Processing high-dimensional data leads to several issues that are known as the *curse of dimensionality*. This phenomenon was initially observed in 1968 by Bellman [8], and later analyzed by Hughes [27], but related situation are still being explored and are a principal research topic in data mining [6]. Frequent consequences of the curse of dimensionality are inconsistent data mining algorithms, overfitting, and prolonged computational times.

* Corresponding author.
E-mail address: peter.drotar@tuke.sk (P. Drotár).

https://doi.org/10.1016/j.ins.2018.12.033
0020-0255/© 2018 Published by Elsevier Inc.

To effectively employ data mining methods, feature selection (FS) and dimensionality reduction methods are often applied as a preprocessing step to high-dimensional datasets [11]. The goal of both approaches is to minimize the dimensionality of the data to the number of features (latent variables) necessary to describe the data, i.e., to their *intrinsic* dimension. Dimensionality reduction is the transformation of high-dimensional data into a meaningful representation in a distinct feature space of reduced dimensionality. The apparent disadvantage of this approach is that the distinct feature space has no physical meaning for interpretation. On the contrary, FS techniques reduce the original feature space without transformation, so that the original features are preserved and cogent interpretation is possible. Other benefits of FS include the removal of irrelevant or redundant features that may produce accidental correlations in data mining algorithms and degrade the model performance; the production of models built on fewer features that are simpler, and easier to interpret and visualize; and the need for less data storage space [1]. Additionally, a reduction of feature space is associated with a reduction in the search space that must be explored by a mining algorithm, which saves computational resources and speeds up data mining procedures.

Feature selection is currently used in many different areas. Aspects specific to particular application domain needs to be considered in FS design. Probably the most representative applications of FS are bioinformatics, multimedia and social networks [33]. FS in bioinformatics has to face very challenging scenarios of high dimensionality small sample size datasets. The usual task is to identify biomarkers for cancer diagnosis or improve the disease prediction accuracy. Especially for biomarker discovery, is the stability of feature selection very important topic. In area of multimedia retrieval, the obtained features are frequently over-complete to describe certain semantics. The task of FS is to select the limited number of discriminative features for certain semantics to provide better interpretability of multimedia [46]. The social media, are very specific for FS, in the sense that they bring new problem of FS for linked social media data. The FS for social media needs to solve two fundamental problems: what are distinctive relations that can be extracted and how to represent these relations and integrate them in FS [43].

FS methods are commonly divided into filter, wrapper, and embedded methods. The main difference between these approaches is in the way they interact with a classifier. In filter FS methods, the FS search is completely isolated from the construction of the classification model. The feature relevance score is calculated according some criterion and the features that achieve the lowest scores are discarded from further processing. An extensive survey of filter FS for biomarker discovery is available in [31]. The advantages of filter FS are that it is scalable, not computationally demanding, and relatively fast compared to other FS approaches. Filter FS can be further divided into univariate and multivariate techniques. Whereas univariate methods evaluate each feature individually, multivariate techniques also consider feature dependencies. Although univariate techniques are quite simple, their performance is competitive with multivariate techniques, and even more complex wrapper methods [21]. Wrapper FS methods utilize the classification model as part of the FS process. Selected feature subsets are evaluated by training and testing classifiers. The final subset is the one that achieves the highest evaluation score. Usually, the search through all possible combinations of features is computationally unfeasible, so a different search approach must be employed; most frequently, deterministic approaches such as sequential forward selection or sequential backward elimination are used. However, some recently proposed wrapper FS techniques take advantage of evolutionary computation and utilize methods such as particle swarm optimization [45] and genetic programming [2]. A very recent comprehensive review of evolutionary computation approaches to FS is provided in [47]. Drawbacks of the wrapper approach are the risk of overfitting and its high computational requirements. Additionally, they are classifier specific. An effective approach to lowering computational requirements is to combine a simple filter feature ranking scheme with a wrapper method into two-step FS [24]. In the first step, the most salient features are selected by the filter method, significantly reducing the feature space. Then, in the second step, the wrapper FS searches the reduced feature space for the optimal feature subset. Embedded methods provide an alternative with better computational complexity than wrapper methods. Unlike the wrapper approach, it avoids repetitive execution of the classifier and the evaluation of numerous feature subsets. The FS process is part of the learning algorithm and uses its properties to evaluate feature significance. However, similarly as the wrapper methods the embedded methods are classifier specific. Besides these three well-known FS approaches, a new group of methods has recently emerged that is built on top of existing FS methods: ensemble FS [10,39]. Ensemble FS constructs groups of feature subsets and then combine these subsets to generate aggregated results. The aim of ensemble FS is to provide more robust and stable FS performance when dealing with high-dimensional data. The main drawback of the ensemble approach is that since the final output is built as an ensemble of multiple base selectors it is difficult to understand and interpret.

Many of the previously proposed FS methods were not designed to work with high-dimensional data and as such are not sufficient nowadays. Driven by motivation to propose efficient FS methods, new papers on topic are appearing regularly. The most recent contributions, advances and shortcomings in the area of FS are studied in several review papers [4,14,32,33]. The most of the recent review studies suggest the ensemble FS as the very promising approach to deal with high dimensional data.

In this paper, we present an approach to build FS ensembles by aggregating FS through voting techniques. We employ voting techniques such as Borda count, plurality vote, single transferable vote (STV), and several modifications of Borda count. Additionally, we propose a novel clustering approach for grouping similar FS outputs prior to aggregation. Experimental results on several artificial and real-world datasets demonstrate the validity of this approach and show improvements in stability and predictive performance over conventional FS methods.

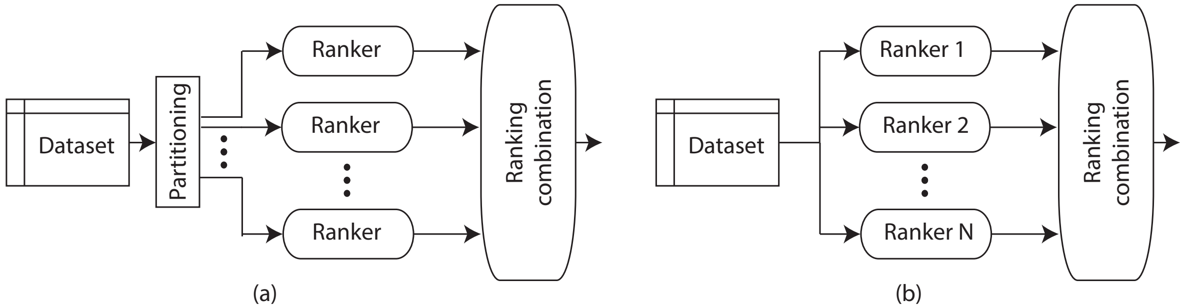

Fig. 1. Block diagram of (a)homogeneous and (b)heterogeneous ensemble FS.

The remainder of this paper is organized as follows. In the next section, we outline the principle and types of ensemble FS. Next, we describe the proposed ensemble FS approach, by describing the FS methods, aggregation techniques, and mean-shift-based clustering algorithm used. In the fourth section, we present our data and experimental results. We evaluate the FS sensitivity, FS stability, and classification accuracy of the proposed methods.

2. Ensemble feature selection

Ensembles of classifiers have been used successfully to solve many classification problems [20]. Thanks to their inherent robustness, they can tackle problems that are problematic for single classifiers; naturally, a similar ensemble approach has therefore propagated to FS. Ensemble FS exploits several selectors to rank features and then combines their results.

Ensemble FS can be carried out in several ways. One possibility is to exploit the diversity of the data (homogeneous ensembles); the other is through so-called functional diversity (heterogeneous ensembles) [10,39]. Homogeneous ensembles are constructed using the same FS method that is being applied to multiple subsamples of the dataset. Depending on the resampling strategy, we distinguish between vertical and horizontal partitioning [34]. In vertical partitioning, each subsample contains data from all observations in the dataset, but features are distributed to the subsamples based on a particular subsampling strategy. In horizontal partitioning, the dataset is divided into subsamples that each have the same features as the original dataset but contain only a subset of the observations. After data resampling and the application of FS, the results are aggregated as illustrated in Fig. 1(a). The majority of work on ensemble FS uses the homogeneous approach. Heterogeneous FS ensembles is comparatively unexplored; only a few studies on functional diversity are available. Contrary to the previous approach, heterogeneous ensembles use the same dataset throughout the FS process. To create diversity, multiple FS techniques are applied to rank the features for each technique. Similarly to homogeneous ensembles, the multiple ranked lists are aggregated to one final ordered list, as illustrated in Fig. 1(b).

3. Proposed approach

Ensembles can be thought of as multiple voters expressing their preferences for candidates; the winner or group of winners is selected based on these votes. This concept motivates our approach to using voting methods to create FS ensembles. Here, FS methods represent voters, and features can be thought of as candidates.

We employ heterogeneous ensembles with different FS methods as rankers. The ranked features are aggregated according to one of the selected strategies. We consider simple ensembles, such as *min, max, mean*, and *median* ensembles as well as ensembles based on voting strategies: plurality vote, single transferable vote, Borda voting, and weighted Borda voting.

FS techniques with similar underlying concepts tend to produce similar output [19,21]. If multiple FS techniques are combined and several are similar, they will dominate in aggregation and the resulting output will be strongly biased toward their choice. This outcome can be avoided by careful selection of the FS methods for an ensemble or robust voting/aggregation approach. However, which FS methods have similar backgrounds may not be apparent, and it is difficult to design a robust voting approach that works in every scenario. Therefore, in addition to a conventional heterogeneous ensemble, we also propose a clustered ensemble. After being ranked with the base FS methods, outputs are clustered and only then are they aggregated, as shown in Fig. 2. Clustering recognizes similar FS outputs (or outputs that are more similar to each other than to others) and groups them. Since similar FS outputs are clustered together, they have less chance to over-vote the other methods, which enhances diversity.

3.1. Feature selection techniques

Among the broad suite of FS algorithms available in the literature, we selected eight methods, all of which are based on different metrics. The utilized methods belong to the group of filter FS methods, since these have the advantage of being less computationally complex than wrapper methods, while still providing competitive results.

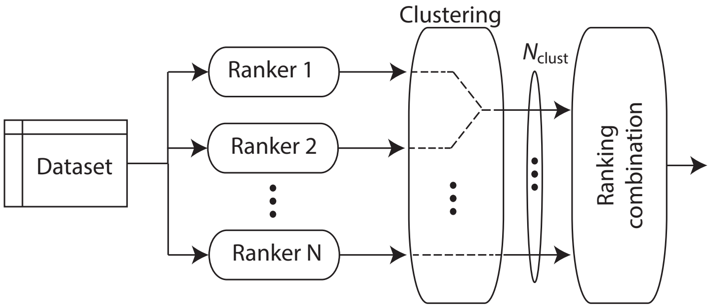

Fig. 2. Block diagram of clustered heterogeneous ensemble FS.

Statistically based FS methods are used very frequently because of their wide range of implementations and relative simplicity. They are mostly based on statistical hypothesis tests. Particularly, we use *t*-test FS, ANOVA-based FS, and FS based on the Pearson correlation coefficient and Gini index. FS based on a maximal information coefficient is used to represent information-theoretic FS [22]. We also use robust feature selection (RFS) via joint *l*2, 1-norms minimalization, which has its roots in sparsity regularization [35]. Finally, we employ two additional popular methods: RELIEF [38] and FS based on the Fisher criterion [22].

3.2. Election methods for combining ranked lists

Methods designed to combine several ranked lists into a single final decision are, in general, known as ensemble or rank aggregation techniques. In this study, we employ several basic ensembles—*mean, median, min*, and *max*—and also propose ensembles based on election methods—*plurality vote* (E-plu), *single transferable vote* (E-STV), *Borda vote* (E-Borda), and the novel *weighted Borda vote*. The choice of ensemble method clearly has a strong effect on the resulting output. Some approaches are more biased towards the majority vote, whereas others may enhance the results of divergent rankers. There is probably no preferred strategy. Enhancing the result of divergent voters can be beneficial if the divergent rankers discovered a pattern that the other rankers missed. However, if a divergent ranker is misled, enhancing this vote can degrade performance. We compare several voting schemes in the heterogeneous ensemble setup and analyze how they affect the final ranker decision.

Simple ensembles. determine the output by finding the feature position across all rankings and computing the *mean, median, minimum*, or *maximum* value of the feature rank. We denote the ensemble FS methods built on the simple ensemble approach as E-mean, E-median, E-min, and E-max, respectively.

Plurality vote. (sometimes called relative majority) is a frequently employed voting scheme. Each ranker selects its preferred winner. The candidate that obtains the most preference votes is selected for the resulting output and is removed from the list of candidates. The procedure is repeated until required number of candidate features is selected. Note that a candidate does not need to have the majority of votes.

Single transferable vote. (STV) theoretically ensures proportional representation of voter preference. This diversity means that even features preferred by a minority of voters are selected in the final subset. Whereas most FS methods (voters) are based on the same underlying principle, yielding very similar output, STV allows minority feature selectors to elect features into the final subset in proportion to the minority's size. Under STV, each voter provides an ordered list of candidates from most preferred to least preferred. If any of the candidates achieves a number of votes higher than or equal to a *quota*, this candidate is declared the winner. The *quota* is the lowest number of votes required for a candidate to win. Here, we apply the Droop quota, defined as $Q = \lceil |V|/(N_W + 1) \rceil + 1$, where $|V|$ is the number of voters (i.e., FS methods) and N_W represents the number of attributes to be selected by the voting algorithm [7]. Votes assigned to the winning attribute that exceed the quota are reassigned to other candidate attributes. If there is no winning feature, the feature with the fewest votes is eliminated and the votes are reassigned. This iteration continues until N_W attributes are found.

Borda count. is probably the most popular voting approach, in which each attribute receives $b_i = \sum_{|V|} N_f - p_v$ points. p_v is the position of the *i*th attribute in an ordered list produced by the *v*th ranker and $i = 1, \ldots, N_f$, where N_f represents the total number of features. The features with the highest b_i are selected as an output set. In the following, we present a weighted modification of the Borda count. The choice of a linear score system is arbitrary, and not suited to FS problem; it is usually not correct to assume that ordinal ranking should translate into a linear score preference. Therefore, in the next section, we propose modifications of the Borda count to adapt it to FS problems.

The weighted Borda count. adds weights to the linear preference used in the conventional Borda scheme. The selection of an appropriate weighting function can significantly influence the behavior of the Borda count. Let us denote the score assigned to the ith feature by the vth ranker as s_v^i. Then,

$$b_i = \sum_{|V|} N_f - p_v = \sum_{|V|} s_v^i \tag{1}$$

and

$$\boldsymbol{b} = \sum_{|V|} \boldsymbol{s}_v, \tag{2}$$

where $\boldsymbol{b} = \{b_1, \ldots, b_{N_f}\}$ is a vector of feature scores.

We propose several weighting schemes to enhance the performance of the conventional Borda count. The scoring in the weighted Borda count can be expressed as follows. First, let us define bijective function/operation $\psi(\cdot)$ as the operation performing descending ordering of the vector elements. For instance, for vector $\boldsymbol{a} = (3, 1, 4, 2)$, one can write $\boldsymbol{a}' = \psi(\boldsymbol{a}) = (4, 3, 2, 1)$. The inverse operation $\psi^{-1}(\cdot)$ returns elements to their original order.

To obtain weighted scores, the sorted Borda score $\boldsymbol{s}'_v = \psi(\boldsymbol{s}_v)$ is multiplied by the weighting function

$$\boldsymbol{s}'_v \odot \boldsymbol{w}. \tag{3}$$

Then, the weighted score of each feature is determined by $\boldsymbol{b} = \sum_{|V|} \psi^{-1}(\boldsymbol{s}' \odot \boldsymbol{w})$.

We employ two weighting functions: *step weighting* and *power weighting.*

The *staircase (step) weighting* function is built by summing multiple step functions. Let us define discrete step function $u[n]$ as

$$u[n] = \begin{cases} 1, & n > 0, \\ 0, & n \leq 0, \end{cases}$$

where $n \in \mathcal{N}$. Two parameters must be set: the number of elements to be weighted M and the number of steps K. Then, the length of one step is $L = M/K$ and the weighting function is defined as

$$\boldsymbol{w[n]} = \sum_{k=1}^{K} u[k \cdot L - n]. \tag{4}$$

The staircase weighting function is utilized to build ensemble E-W$_{\text{stair}}$B. We also consider the special case of E-W$_{\text{stair}}$B obtained by setting $K = 1$, resulting in the *unit step weighting function* $\boldsymbol{w[n]} = u[M - n]$. Let us denote this approach as E-W$_{\text{step}}$B.

The *power function* is defined as

$$\boldsymbol{w[n]} = (M - k)^p u[k], \tag{5}$$

for $k = 0, \ldots, M$. Increasing the value of p leads to the steeper descent of assigned weights for decreasing rank. In this study, $p = 3$.

3.3. Clustering for ensemble FS using a mean-shift algorithm

In addition to conventional ensemble FS, we also propose clustered ensembles. The preferences of similar rankers are first clustered and then used to build ensembles. We use mean-shift clustering, since it does not require a priori specification of the number of clusters and provides a good trade-off between performance and computational complexity [18].

The mean shift is a nonparametric iterative algorithm for locating the modes of a density function. Given n data points $\boldsymbol{x}_i$, $i = 1, \ldots, n$ in d-dimensional space $\mathcal{R}^d$, the multivariate kernel density estimator, with kernel $K(x)$ and window radius h, is given by

$$f(\boldsymbol{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right). \tag{6}$$

We focus only on symmetric kernels satisfying $K(\boldsymbol{x}) = c_{k,d} k(||\boldsymbol{x}^2||)$ with strictly positive normalization constant $c_{k,\,d} > 0$, ensuring that $K(\boldsymbol{x})$ integrates to one. The modes are located among the zeros of gradient function $\nabla f(x) = 0$. Assuming $g(x) = -k'(x)$, the gradient of the density estimator is

$$\begin{aligned} \nabla f(x) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{x}) g\left(\left|\left| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right|\right|^2 \right) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left|\left| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right|\right| \right) \right] \left[\frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\left|\left| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right|\right|^2 \right)}{\sum_{i=1}^{n} g\left(\left|\left| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right|\right|^2 \right)} - \boldsymbol{x} \right]. \end{aligned} \tag{7}$$

The first term in (7) is proportional to the density estimate at $\boldsymbol{x}$ computed with kernel $G(\boldsymbol{x}) = c_{k,d}g(||\boldsymbol{x}||^2)$. The second term,

$$\boldsymbol{m}_{h,G(\boldsymbol{x})} = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\left|\left|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right|\right|^2\right)}{\sum_{i=1}^{n} g\left(\left|\left|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right|\right|^2\right)} - \boldsymbol{x}, \tag{8}$$

is called the mean shift [18]. The mean shift always points toward the direction of the maximum increase in density. The mean shift is obtained by successive computation of Eq. (8) and translation of the kernel $G((\boldsymbol{x}))$ by $\boldsymbol{m}_{h,G(\boldsymbol{x})}$, and convergence is guaranteed for a uniform kernel.

The clustering is embedded after ranker blocks, as indicated in Fig. 2, and reduces the number of ranked outputs that are input to the voting block. An issue to consider is how many of the highest-ranking features should be fed to the clustering algorithm. Using too many features from each ranker makes cluster construction difficult; therefore, the number of features for clustering N_{clu} should be close to the number of relevant features.

The input to clustering algorithm are data samples of dimensionality N_{clu} representing output of different rankers. Mean-shift identifies clusters and labels data samples belonging to the same cluster. The centroid of each cluster is determined and later processed by voting stage. The centroid is calculated by finding mean rank of every feature in particular cluster. To determine centroid, full ranked list is used, not only N_{clu} highest ranking feature. In this paper, the clustered ensemble FS methods are denoted by a “C” at the beginning of their abbreviation (e.g., CE-Borda, CE-mean, CE-STV, etc.).

4. Empirical study

In this section, we present our experimental results, measured by the sensitivity of FS, which captures the ability of FS to identify relevant features; FS stability, measured by relative weighted consistency; and prediction performance, measured by accuracy and $F1$ score. The sensitivity of FS was evaluated on five artificial datasets, whereas stability and prediction performance were used to examine ten high-dimensional real-world datasets and one artificial high-dimensional dataset. We compared the proposed ensemble approaches with simple ensembles and conventional FS methods.

4.1. Sensitivity of feature selection on artificial data

We used five different artificial datasets in this study. The advantage of using artificial datasets is that we have knowledge of the relevant features. We know exactly how many relevant features are in the dataset and exactly which ones they are. In this paper, under the expression relevant features we understand both strongly relevant and weakly relevant features as defined in [29]. We used the Madelon, LED, XOR, CorrAL-100, and high-dimensional Madelon datasets.

The XOR dataset. contains 50 observation and 100 features. Of these, two features are relevant and the rest are randomly generated. The two relevant features are correlated to the class label with the XOR operation: $classlabel = f_1 \oplus f_2$.

The LED dataset. represents a seven-segment display with the segments being the seven relevant features. The remaining features are randomly generated binary features. Like the XOR dataset, LED contains 50 observations and 100 features. LED is a relatively simple classification task.

The CorrAL-100 dataset. was formed by generating all possible combinations of five binary features: f_1, f_2, f_3, f_4, and f_5. Features f_1, f_2, f_3, and f_4 are correlated to the class label by logical operations $(f_1 \wedge f_2) \vee (f_3 \wedge f_4)$. Feature f_5 is correlated to the class label by 75% and is redundant if the four relevant features are selected. By definition is f_5 weakly relevant, but since it is also redundant, we follow approach adopted in [9] and consider it as irrelevant. Another 94 irrelevant binary features are added to a create dataset of 100 features.

The Madelon dataset. is a binary classification problem that was part of the NIPS 2003 challenge. The relevant features are situated on the vertices of a five-dimensional hypercube. The irrelevant features are drawn randomly from a Gaussian distribution. We have used a slightly modified Madelon dataset that contains no redundant or repeated features.

MadelonHD. (MHD) is generated in a comparable way to the *Madelon* dataset, but to imitate real high-dimensional datasets, its dimensionality was increased to 15000 features, 15 of which are relevant. *MadelonHD* was created to simulate the real high-dimensional, low-sample datasets that are frequently encountered is many domains.

A summary of the above datasets is given in Table 1.

To evaluate FS quality, we calculated the sensitivity of FS, defined as

$$\mathrm{Sen} = \frac{R_s}{R_t} \times 100. \tag{9}$$

R_s is the number of relevant features selected and R_t is the total number of relevant features.

Table 1
Characteristics of artificial datasets used in this study.

Name	No. samples	No. all features	No. relevant features
XOR	50	100	2
LED	50	100	7
CorrAL-100	128	100	4
Madelon	100	500	5
MadelonHD	150	15000	15

Table 2
Characteristics of real-world datasets used in this study.

Dataset	Source	No. samples	No. features	No. Class 0	No. Class 1
ALO	Alon [3]	62	2000	40	22
BOR	Borovecki [12]	31	22,283	17	14
BUR	Burczynski [13]	127	22,283	85	42
CHO	Chowdary [17]	104	22,283	62	42
CHIN	Chin [16]	118	22,215	43	75
GOL	Golub [25]	72	7129	47	25
GOR	Gordon [26]	181	12,533	94	87
POM	Pomeroy [37]	60	7128	39	21
SIN	Singh [40]	102	12,600	52	50
TIA	Tian [44]	173	12,625	36	137

The number of features N_{sel} that should be returned by ranker is determined according the strategy from [9], and is considered for computing the sensitivity. For small datasets whose total number of features N_f is less than 10, $N_{sel} = N_f \times 0.75$. For $10 < N_f < 75$, $N_{sel} = N_f \times 0.4$. For the XOR, LED, and CorrAL-100 datasets, which fall into the interval $75 < N_f \leq 100$, the number of selected features $N_{sel} = N_f \times 0.1$. For high-dimensional datasets, we chose $N_{sel} = 50$.

4.2. Stability and prediction performance on high-dimensional datasets

The high-dimensional small sample size datasets represent challenging scenario in many FS applications. We focus on high-dimensional datasets and analyze stability and influence of FS on prediction performance.

4.2.1. Data

Ten real-world datasets were used to evaluate stability. An overview of the datasets, with corresponding references, is provided in Table 2. All datasets are high-dimensional, with numbers of features ranging from 2000 to 22,000. We focused on these datasets because FS is more challenging and beneficial in high-dimensional settings. All datasets are binary datasets or datasets that were converted to binary datasets. Although this narrows down the experiments coverage, note that most multiclass classification problems can be transformed into multiple binary classification tasks through one-against-one or one-against-all approaches.

4.2.2. The stability of feature selection

Important aspect of FS is its *stability*, which was defined by Kalousis [28] as “the robustness of the feature preferences it produces to differences in training sets drawn from the same generating distribution.” Stability captures how variations in data affect feature preferences.

Stability is of paramount importance in bioinformatics, where FS is not only used as a preprocessing step, but also applied for biomarker discovery. Here, for researchers to have confidence in their findings, it is crucial to obtain the same biomarkers on different data. If the biomarkers selected for the newer or partially modified data are not the same as the previously discovered biomarkers, then the conclusions are not valid. There are several reasons for low stability. A main root of this problem is high dimensionality combined with small sample sizes. Another source of instability is the FS algorithms themselves. Most current FS methods were designed without considering stability. Consider a wrapper algorithm selecting the minimal subset of features with the highest classification accuracy. There can be more features relevant for a target variable, but the algorithm selects only the part of them that achieves the highest accuracy. In the following selection method, different set of features yields highest accuracy, so these are selected. As a result, not all significant features are included in all selections and stability is lower. Redundant or correlated features also contributes to selection instability [5].

We employed a perturbation strategy to measure stability. Dataset perturbation is the process of randomly selecting a portion of the observations from a dataset to create a reduced dataset. In this study, we used 80% of the original data for reduced datasets. FS is applied to each of these reduced datasets. This process is repeated 100 times, selecting the 50 highest-ranking features. The ranked lists are then compared and evaluated using a concrete stability measure.

Several stability measures were proposed to capture variations in ranked lists produced by the FS methods. Kalousis et al. [28], in their pioneering work on FS stability, used Pearson’s correlation coefficient to measure stability through feature weighting, Spearman’s rank correlation coefficient to measure the stability of multiple rankings, and Tanimoto distance to measure the stability of multiple subsets. Dunne [23] addressed issues of stability independently of Kalousis, utilizing Hamming distance. Other measures have been developed, such as the stability index [30] and the Pseudo-Hamming index [41]. Herein, we applied the relative weighted consistency CW_r, which succeeds in identifying randomness in FS and provides a reliable stability comparison [41].

CW_r is defined as follows. Assume that $F = \{f_1, \ldots, f_{N_f}\}$ is the set of all features of cardinality N_f and $\mathcal{S} = \{S_1, \ldots, S_J\}$ is a system of J feature subsets, obtained by applying a particular FS algorithm J times on different samplings of a dataset. Then, the weighted consistency index is

$$CW(\mathcal{S}) = \sum_{f \in F} \frac{\Psi_f}{N_O} \cdot \frac{\Psi_f - 1}{J - 1}, \tag{10}$$

where $N_O = \sum_{i=1}^{J} S_i$ is the number of occurrences of any feature in $\mathcal{S}$ and Ψ_f is the number of occurrences of feature $f \in F$ in $\mathcal{S}$ [41]. The relative weighted consistency index CW_{rel} is obtained by adjusting CW on its minimal CW_{min} and maximal CW_{max} possible values as

$$CW_{\text{rel}}(\mathcal{S}) = \frac{CW(\mathcal{S}) - CW_{\text{min}}(N_O, J, F)}{CW_{\text{max}}(N_O, J) - CW_{\text{min}}(N_O, J, F)}. \tag{11}$$

4.2.3. Influence of FS on classification performance

We conducted a series of experiments to compare the influence of FS algorithms on prediction performance. The datasets used in these experiments were those employed in the stability experiments, i.e., the 10 datasets listed in Table 2 and the synthetic MadelonHD dataset.

To provide the objective estimate of classifier performance we employ stratified five-fold cross-validation. The feature subset was selected in cross-validation, using only the training data at each cross-validation iteration. The whole process is repeated five times over each dataset and the performance over five repetitions were averaged over each dataset. Prior to classification, the features were normalized on per feature basis to have a mean of zero and unit variance.

The classifiers used in these experiments were AdaBoost (Ada) with a decision tree as a base estimator and naive Bayes (NB). We used the Python scikit-learn module implementation of these classifiers [36].

The naive Bayes classifier is based on the Bayesian theorem and assumes that features in a dataset are independent [15]. The classifier combines a naive Bayes probability model with a decision rule. The most common rule is called *maximum a posteriori* and selects the most probable hypothesis. Gaussian naive Bayes assumes that continuous values are distributed according to a Gaussian distribution.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right), \tag{12}$$

where parameters σ_y and μ_y are estimated using the maximum likelihood method.

AdaBoost was the first practical boosting algorithm. Boosting is an iterative procedure that combines the results of many weak classifiers $(C_1(x), C_2(x), \ldots C_M(x))$ on modified data to produce a powerful classifier. A weak classifier is a classifier whose error rate is slightly lower than a random pick. Initially, AdaBoost builds a first weak classifier $C_1(x)$, which is in most cases a decision stump. If at least one misclassification is produced by one of the classifiers, the weight of that observation is increased. Subsequently, the second classifier is build using the new weights. The predictions from all classifiers are then combined by a weighted majority voting technique, thus producing final prediction $C(x) = \text{sign}(\sum_{i=1}^{M} a_m C_m(x))$, where $a_1, a_2, \ldots, a_M$ are numbers returned by the boosting algorithm to increase the influence of the better classifiers in the sequence [48]. Each boosting step applies weights to each of the training data points.

4.3. Results and discussion

Ensemble FS based on a weighted Borda count requires setting several parameters, which we chose experimentally as follows. Power-weighted Borda (E-W$_{\text{power}}$B) and E-W$_{\text{step}}$B required parameter $M = R_t$. For E-W$_{\text{stair}}$B, we used the rule presented in the previous section and set $M = N_f \times 0.1$ for databases of size $75 < N_f \leq 100$, and $M = 50$ for bigger databases. $L = 5$ was used for all datasets. Additionally, for clustering ensembles, we needed to specify how many features should be evaluated by the clustering algorithm. This value was set to $N_{\text{clu}} = N_f \times 0.1$ for databases of size $75 < N_f \leq 100$, and $N_{\text{clu}} = 50$ for bigger databases.

4.3.1. FS sensitivity: experimental results

The experimental results obtained for the five artificial datasets are provided in Table 3. The first, the CorrAl-100 dataset, tests the ability of FS methods to deal with redundancy and correlation. Four features are relevant and one is only partially correlated. In this case, most methods achieved a sensitivity of 100%, and assigned the relevant features the highest ranks. The only methods that did not correctly identify all these features were RFS, which picked only one relevant feature for selected subset of size N_{sel}, and (C)E-max, which selected three out of the four correct features. As will be shown later, the E-max ensemble is the worst performing ensemble method by this measure.

Table 3
Sensitivity for different FS methods on artificial datasets.

	Corrall	XOR	LED	Madelon	MadelonHD	avg
*t*test	**100**	10	28.6	40	26.7	41
RELIEF	98.5	**66.4**	28.5	77.9	54.7	65.2
RFS	25	11.8	**100**	60	60	51.4
Pearson	**100**	10	57.1	40	26.7	46.8
MIC	**100**	10.2	**100**	40	20	54
Fisher	**100**	10	**100**	40	26.7	55.3
Gini	**100**	9.9	10.3	40	26.7	37.4
ANOVA	**100**	10	**100**	40	26.7	55.3
E-min	**100**	46.1	84.7	**85.3**	**77.1**	78.6
CE-min	**100**	46.2	82.6	**85.3**	**77.1**	78.2
E-max	75.3	18	21.3	59.8	33.3	41.5
CE-max	75	18	32.3	59.8	33.3	43.7
E-median	**100**	10	83.4	40	33.3	53.4
CE-median	**100**	14.1	71.3	68.9	34.4	57.7
E-mean	**100**	13.6	60.2	60	37.3	54.2
CE-mean	**100**	17.2	53.8	72.3	37.3	56.1
E-STV	**100**	27.4	96.4	39	27.1	58
CE-STV	95.7	33.4	78.2	39.3	20.4	53.4
E-plu	**100**	10	**100**	40	26.7	55.3
CE-plu	**100**	10.4	81.5	40	26.7	51.7
E-Borda	**100**	36.8	**100**	77	65.1	75.8
CE-Borda	96.2	39.4	84.8	69	40.3	65.9
E-W_{power}B	**100**	38.4	**100**	81	74.2	78.7
CE-W_{power}B	**100**	43.7	90.1	84.1	77	**79**
E-W_{step}B	**100**	10.6	**100**	40	26.7	55.5
CE-W_{step}B	**100**	18.8	90.2	66	68.7	68.7
E-W_{stair}B	**100**	14.8	**100**	63.4	55.6	66.8
CE-W_{stair}B	**100**	34.4	92.9	82.1	76.	77.2

The LED problem is a rather simple classification task, for which the dataset contains seven significant features. Here, almost all ensemble methods based on the Borda vote assigned the highest rank to relevant features. The E-plu ensemble, RFS, ANOVA, and MIC FS methods also achieved an sensitivity of 100%. GINI FS, which performed very well on the CorrAL dataset, performed poorly in this example. This phenomenon is frequently encountered in FS; methods that perform well on one dataset may fail to identify relevant features in another dataset, although ensemble methods are more robust and provide more balanced results than single FS.

The XOR dataset represents a nonlinear problem and contains only two relevant features. Unlike the LED dataset, on which RELIEF performed poorly, here, the RELIEF algorithm produced the highest sensitivity. This is the only dataset on which a conventional FS method outperformed the ensemble methods. RELIEF performed well on nonlinear datasets, significantly better than other FS algorithms. This is also true for the Madelon dataset, on which RELIEF achieves the highest sensitivity of the conventional FS methods. However, in that case, multiple ensemble methods outscored RELIEF. Probably the most interesting results are the sensitivity scores for the MadelonHD dataset. This dataset was used to create conditions that are frequently encountered in many FS application domains. The number of features is significantly higher than the number of samples, and only a few features are relevant. Such data are typical for genomic datasets. The best performers on MadelonHD were E-min and CE-min, followed by CE-W_{power}B and E-W_{stair}B, indicating that the use of clustered ensemble schemes can be advantageous for high-dimensional datasets.

Considering the average performance on all five datasets, the nine best performers are the ensemble methods. Basically, all the clustered ensembles based on conventional and weighted Borda (CE-W_{power}B, CE-W_{stair}B, CE-W_{step}B, and CE-Borda) are between the top performers together with E-W_{power}B, E-Borda and CE-W_{stair}B. From simple ensembles only (C)E-min provided the desired performance. The best conventional FS method was RELIEF; all other methods achieved less than a 60% sensitivity.

4.3.2. Stability: experimental results

As a next step, we compared the stability of the proposed ensemble FS and conventional FS methods. Stability is measured using CW_r by applying FS to 10 real-world datasets and one artificial dataset as described in the previous sections. Results are depicted in Table 4. As expected, stability varies across datasets, since the complexity of the data varies and the data are derived from different sources. We were interested in methods that achieved balanced results on all datasets while maintaining stability.

The parameters for the weighted Borda ensembles were chosen according to the rule explained in previous sections; $M = 50,\ L = 5$ (for (C)E-W_{stair}B), and $N_{clu} = 50$ (for clustered ensembles) were used for the high-dimensional datasets

We determined the mean rank, standardized mean and median of stability for each method to compare the stability. Moreover, WTL (win/tie/loss) represents how many times particular method perform better/same/worse than all other meth-

Table 4
Stability of FS methods on 11 high-dimensional datasets.

FS method	ALO	BOR	BUR	CHIN	CHO	GOL	GOR	MHD	POM	SIN	TIA
*t*test	0.62	0.52	0.51	0.71	0.64	0.71	0.8	0.34	0.33	0.73	0.43
RELIEF	0.63	0.61	0.59	0.64	0.57	0.68	**0.81**	0.17	0.41	0.66	0.42
RFS	0.49	0.6	0.61	0.31	0.47	0.55	0.54	0.3	0.45	0.44	0.37
Pearson	0.62	0.52	0.51	0.71	0.57	0.71	0.8	0.34	0.33	0.73	0.43
MIC	0.44	0.42	0.44	0.66	0.75	0.68	0.8	0.16	0.16	0.69	0.18
Fisher	0.62	0.52	0.51	0.71	0.57	0.71	0.8	0.34	0.33	0.73	0.43
Gini	0.57	0.45	0.46	0.7	**0.79**	0.68	0.77	0.29	0.3	0.72	0.35
ANOVA	0.62	0.52	0.51	0.71	0.57	0.71	0.8	0.34	0.33	0.73	0.43
E-min	0.58	0.41	0.42	0.55	0.57	0.67	0.76	0.3	0.27	0.58	0.3
CE-min	0.58	0.41	0.42	0.55	0.57	0.67	0.76	0.3	0.27	0.59	0.31
E-max	0.52	0.62	0.62	0.41	0.57	0.64	0.75	0.18	0.31	0.62	0.31
CE-max	0.52	0.62	0.62	0.41	0.57	0.64	0.75	0.18	0.31	0.61	0.31
E-median	0.63	0.54	0.54	0.72	0.64	0.71	0.78	0.32	0.34	**0.75**	0.4
CE-median	0.62	0.54	0.55	0.71	0.68	0.71	0.78	0.25	0.34	0.73	0.36
E-mean	0.65	**0.66**	0.66	0.48	0.67	0.73	0.81	0.24	0.37	0.73	0.38
CE-mean	0.62	**0.66**	**0.67**	0.47	0.65	0.71	0.79	0.22	0.36	0.72	0.37
E-STV	0.48	0.05	0.05	0.03	0.05	0.1	0.06	0.11	0.15	0.06	0.1
CE-STV	0.47	0.04	0.04	0.03	0.04	0.1	0.06	0.11	0.14	0.07	0.06
E-plu	0.62	0.52	0.51	0.71	0.57	0.71	0.8	0.34	0.33	0.73	0.43
CE-plu	0.62	0.52	0.51	0.7	0.57	0.71	0.8	0.34	0.33	0.72	0.43
E-Borda	0.69	0.39	0.42	0.7	0.57	0.76	0.69	0.61	0.41	0.59	0.4
CE-Borda	**0.74**	0.63	0.64	**0.83**	0.76	**0.85**	0.77	**0.8**	**0.58**	0.68	**0.56**
E-W_{power}B	0.61	0.31	0.32	0.48	0.4	0.71	0.76	0.34	0.3	0.63	0.34
CE-W_{power}B	0.61	0.31	0.31	0.43	0.4	0.69	0.76	0.31	0.29	0.62	0.32
E-W_{step}B	0.62	0.52	0.51	0.71	0.58	0.71	0.8	0.34	0.33	0.73	0.43
CE-W_{step}B	0.62	0.45	0.45	0.69	0.7	0.7	0.78	0.32	0.31	0.7	0.33
E-W_{stair}B	0.62	0.48	0.49	0.69	0.61	0.72	0.77	0.35	0.33	0.74	0.41
CE-W_{stair}B	0.62	0.44	0.45	0.63	0.66	0.7	0.77	0.32	0.31	0.67	0.33

ods. These results are depicted in Table 5. The four most stable methods were the ensemble methods. Comparing mean rank and standardized mean six out of ten most stable methods were ensembles, whereas the most stable conventional FS method is *t*-test FS ranking (in fifth place). When we considered the median CW_r over all datasets, eight of the ten most stable techniques were ensembles. The most stable were CE-Borda, E-mean, E-median, *t*-test FS and E-W_{step}B.

The FS methods that are of practical interest are those that not only provide stable results, but are also able to accurately identify truly significant features. Since there is no relevant measure that would cover both these aspects, we evaluated the stability results in conjunction with the sensitivity results from previous sections. The best performers in terms of the sensitivity were CE-W_{power}B, E-W_{power}B, E-min, CE-min, CE-W_{stair}B, E-Borda, CE-W_{step}B, E-W_{stair}B, and CE-Borda, all of which outperformed the best conventional method, RELIEF.

Now, as we can see from Table 5, both ensembles based on min and the power-weighted Borda scheme were less stable than RELIEF. On the other hand, CE-Borda clearly outperformed all other methods in stability while yielding also satisfactory results in terms of Sen. Besides CE-Borda, also E-Borda, CE-W_{step}B, CE-W_{stair}B, E-W_{stair}B provided stability comparable to RELIEF, while outperforming RELIEF in sensitivity.

Focusing only on conventional FS, the most stable method was *t*-test-based FS, followed by RELIEF. Interestingly, the *t*-test performed very poorly in attempts to correctly identify true features in synthetic datasets; even though its selections were rather stable, they contained few true features, which poses serious doubts about the subsets of features generated by these methods. By contrast, RELIEF was competitive in terms of stability with ensemble FS methods.

4.3.3. Prediction performance: experimental results

The classification accuracies of AdaBoost and NB utilizing different FS methods are depicted in Tables 6 and 7, respectively. To provide also objective measure for imbalanced datasets, the prediction performance measured by *F*1 score is presented in Table 8 for AdaBoost classifier and Table 9 for naive Bayes classifier. The results yielded by both metrics showed similar trend.

Summary of the prediction performance results are provided in Tables 10 and 11 in terms of *mean rank, standardized mean, median* and WTL statistics. Ensemble methods that achieved the best performance in terms of Sen. ((C)E-Borda, (C)E-W_{power}B, (C)E-W_{stair}B, (C)E-min, CE-W_{step}B) produced the best results also in prediction performance on high-dimensional datasets. Two conventional FS methods performed below average: RFS and RELIEF. For RELIEF, this is unexpected since previous experiments indicated that it could correctly identify significant features. The conventional FS methods that scored highly on prediction performance using NB classifier were *t*-test FS, Fisher FS, Pearson FS, and ANOVA FS. However, although they were very satisfactory in *F*1 score, they generated only average FS sensitivity and stability results. Their sensitivity performance was below-average on the high-dimensional MadelonHD dataset, for which they correctly identified only a fifth of the significant features, making their classification accuracy relatively surprising, as all the datasets used in this

Table 5
Statistics of stability results of FS methods. Comparison of all evaluated methods.

FS method	mean rank	stand. mean	med	WTL
*t*test	9.23	0.41	0.62	187/39/71
RELIEF	11.18	0.38	0.61	183/4/110
RFS	18.18	−0.26	0.47	108/0/189
Pearson	9.68	0.37	0.57	180/43/74
MIC	19.18	−0.48	0.44	97/0/200
Fisher	9.68	0.37	0.57	180/43/74
Gini	15.91	0.16	0.57	133/0/164
ANOVA	9.68	0.37	0.57	180/43/74
E-min	21.68	−0.19	0.55	65/9/223
CE-min	21.5	−0.19	0.55	67/9/221
E-max	19.36	−0.13	0.57	91/8/198
CE-max	19.27	−0.14	0.57	92/8/197
E-median	8	0.42	0.63	220/0/77
CE-median	9.82	0.35	0.62	200/0/97
E-mean	7.91	0.49	0.66	221/0/76
CE-mean	10.73	0.38	0.65	190/0/107
E-STV	27	−2.69	0.06	11/0/286
CE-STV	27.82	−2.76	0.06	2/0/295
E-plu	9.45	0.37	0.57	184/40/73
CE-plu	10.91	0.37	0.57	188/0/109
E-Borda	13.82	0.48	0.59	156/0/141
CE-Borda	**4.36**	**1.55**	**0.74**	260/0/37
E-W_{power}B	19.45	−0.29	0.4	94/0/203
CE-W_{power}B	21.36	−0.37	0.4	73/0/224
E-W_{step}B	8.55	0.38	0.58	214/0/83
CE-W_{step}B	14.64	0.2	0.62	147/0/150
E-W_{stair}B	11	0.33	0.61	187/0/110
CE-W_{stair}B	16.64	0.11	0.62	125/0/172

Table 6
Influence of FS on classification accuracy with AdaBoost classifier. Ten real-world datasets and one artificial dataset.

FS	ALO	BOR	BUR	CHIN	CHO	GOL	GOR	MHD	POM	SIN	TIA
*t*test	87 ± 5	92 ± 13	92 ± 12	84 ± 9	93 ± 5	91 ± 9	93 ± 4	65 ± 1	58 ± 10	92 ± 7	75 ± 5
RELIEF	81 ± 7	94 ± 9	91 ± 18	85 ± 4	93 ± 4	90 ± 6	95 ± 4	72 ± 3	64 ± 12	92 ± 6	**78 ± 7**
RFS	86 ± 9	93 ± 15	91 ± 18	82 ± 8	93 ± 9	93 ± 7	96 ± 3	74 ± 2	**69 ± 8**	87 ± 7	69 ± 8
Pearson	86 ± 5	91 ± 14	94 ± 9	84 ± 9	93 ± 5	91 ± 9	94 ± 3	65 ± 1	58 ± 10	92 ± 7	75 ± 5
MIC	85 ± 11	91 ± 14	93 ± 13	83 ± 1	95 ± 5	95 ± 6	**97 ± 4**	67 ± 2	60 ± 11	**93 ± 6**	72 ± 7
Fisher	86 ± 5	92 ± 11	88 ± 21	84 ± 9	93 ± 5	91 ± 9	93 ± 3	65 ± 1	58 ± 10	92 ± 7	75 ± 5
Gini	85 ± 10	92 ± 12	91 ± 16	82 ± 4	95 ± 5	**96 ± 5**	96 ± 4	73 ± 1	62 ± 4	90 ± 8	77 ± 4
ANOVA	85 ± 5	94 ± 12	90 ± 13	84 ± 9	93 ± 5	92 ± 8	94 ± 4	65 ± 1	58 ± 10	92 ± 7	75 ± 5
E-min	84 ± 13	90 ± 16	**96 ± 8**	85 ± 5	95 ± 5	93 ± 4	97 ± 4	69 ± 2	60 ± 14	94 ± 7	76 ± 7
CE-min	83 ± 15	94 ± 12	92 ± 15	85 ± 5	95 ± 5	92 ± 4	97 ± 4	69 ± 2	60 ± 14	94 ± 7	76 ± 7
E-max	81 ± 12	87 ± 18	88 ± 18	84 ± 7	96 ± 3	92 ± 7	95 ± 3	69 ± 2	61 ± 9	91 ± 7	77 ± 4
CE-max	81 ± 12	87 ± 14	90 ± 16	84 ± 7	96 ± 3	91 ± 8	95 ± 3	69 ± 2	62 ± 9	91 ± 8	78 ± 4
E-median	79 ± 17	93 ± 10	88 ± 21	86 ± 6	91 ± 7	92 ± 7	93 ± 3	67 ± 3	64 ± 10	92 ± 5	74 ± 7
CE-median	74 ± 17	93 ± 10	93 ± 11	85 ± 5	94 ± 7	93 ± 6	96 ± 3	72 ± 3	66 ± 9	92 ± 8	74 ± 7
E-mean	80 ± 10	89 ± 17	92 ± 14	83 ± 7	96 ± 5	92 ± 8	95 ± 4	70 ± 2	62 ± 12	89 ± 4	77 ± 7
CE-mean	83 ± 13	85 ± 17	90 ± 11	84 ± 6	96 ± 2	92 ± 7	95 ± 3	69 ± 2	61 ± 11	90 ± 5	77 ± 5
E-STV	81 ± 12	93 ± 12	93 ± 12	**86 ± 5**	92 ± 6	94 ± 6	95 ± 2	74 ± 3	63 ± 11	90 ± 7	76 ± 6
CE-STV	81 ± 11	88 ± 23	89 ± 19	84 ± 7	94 ± 5	91 ± 5	94 ± 3	**75 ± 2**	62 ± 11	90 ± 7	75 ± 6
E-plu	87 ± 4	88 ± 21	94 ± 10	84 ± 9	93 ± 5	92 ± 8	93 ± 3	65 ± 1	58 ± 10	93 ± 6	75 ± 5
CE-plu	**87 ± 5**	93 ± 12	92 ± 12	85 ± 9	92 ± 4	92 ± 8	94 ± 3	66 ± 2	59 ± 9	92 ± 7	74 ± 4
E-Borda	86 ± 13	92 ± 13	90 ± 16	86 ± 6	95 ± 5	94 ± 6	94 ± 3	73 ± 2	62 ± 9	93 ± 6	75 ± 8
CE-Borda	87 ± 11	96 ± 7	90 ± 20	86 ± 6	95 ± 5	95 ± 5	94 ± 3	73 ± 2	62 ± 9	93 ± 6	75 ± 8
E-W_{power}B	87 ± 10	92 ± 13	93 ± 11	83 ± 6	**97 ± 4**	94 ± 5	97 ± 4	67 ± 3	64 ± 11	94 ± 7	75 ± 7
CE-W_{power}B	84 ± 13	88 ± 17	92 ± 13	86 ± 6	96 ± 4	94 ± 5	97 ± 4	69 ± 3	65 ± 8	**95 ± 6**	76 ± 7
E-W_{step}B	86 ± 6	90 ± 17	94 ± 12	86 ± 8	93 ± 5	91 ± 10	93 ± 3	65 ± 2	60 ± 10	93 ± 7	75 ± 5
CE-W_{step}B	82 ± 11	92 ± 13	95 ± 9	85 ± 6	95 ± 5	94 ± 5	97 ± 4	71 ± 3	64 ± 12	92 ± 6	74 ± 5
E-W_{stair}B	83 ± 13	91 ± 15	91 ± 17	83 ± 6	95 ± 5	93 ± 6	96 ± 3	71 ± 3	68 ± 10	94 ± 4	76 ± 5
CE-W_{stair}B	78 ± 18	**97 ± 5**	92 ± 15	85 ± 5	95 ± 5	94 ± 5	97 ± 4	69 ± 3	66 ± 13	94 ± 5	74 ± 6

Table 7
Influence of FS on classification accuracy with naive Bayes classifier. Ten real-world datasets and one artificial dataset.

FS	ALO	BOR	BUR	CHIN	CHO	GOL	GOR	MHD	POM	SIN	TIA
*t*test	84 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	98 ± 3	61 ± 1	70 ± 16	**93 ± 6**	76 ± 8
RELIEF	86 ± 11	89 ± 18	89 ± 17	88 ± 4	88 ± 6	**95 ± 5**	99 ± 2	63 ± 5	67 ± 9	91 ± 7	76 ± 10
RFS	77 ± 7	78 ± 23	78 ± 23	86 ± 5	88 ± 5	94 ± 3	99 ± 2	63 ± 2	69 ± 9	90 ± 7	72 ± 7
Pearson	84 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	98 ± 3	61 ± 1	70 ± 16	**93 ± 6**	76 ± 8
MIC	85 ± 12	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 5	94 ± 6	99 ± 2	65 ± 2	69 ± 5	91 ± 9	67 ± 6
Fisher	84 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	98 ± 3	61 ± 1	70 ± 16	**93 ± 6**	76 ± 8
Gini	84 ± 11	**96 ± 8**	**96 ± 8**	88 ± 6	93 ± 11	94 ± 6	99 ± 2	63 ± 2	65 ± 16	92 ± 7	75 ± 9
ANOVA	84 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	98 ± 3	61 ± 1	70 ± 16	**93 ± 6**	76 ± 8
E-min	86 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	95 ± 5	94 ± 6	**100 ± 0**	61 ± 3	66 ± 14	91 ± 8	78 ± 5
CE-min	86 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	95 ± 5	94 ± 6	**100 ± 0**	61 ± 3	66 ± 14	91 ± 8	78 ± 5
E-max	84 ± 12	93 ± 10	93 ± 10	86 ± 5	96 ± 4	92 ± 6	**100 ± 0**	60 ± 2	66 ± 14	93 ± 6	73 ± 10
CE-max	84 ± 12	93 ± 10	93 ± 10	86 ± 5	96 ± 4	92 ± 6	**100 ± 0**	60 ± 2	66 ± 14	93 ± 6	73 ± 10
E-median	82 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 5	94 ± 6	100 ± 1	59 ± 3	69 ± 16	**93 ± 6**	76 ± 8
CE-median	82 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	99 ± 2	59 ± 3	71 ± 11	**93 ± 6**	75 ± 8
E-mean	82 ± 14	**96 ± 8**	**96 ± 8**	86 ± 4	**98 ± 3**	94 ± 6	**100 ± 0**	61 ± 3	65 ± 14	91 ± 7	71 ± 9
CE-mean	82 ± 14	**96 ± 8**	**96 ± 8**	86 ± 5	97 ± 4	94 ± 5	**100 ± 0**	61 ± 2	66 ± 16	91 ± 7	72 ± 9
E-STV	74 ± 13	95 ± 10	95 ± 10	87 ± 5	95 ± 4	92 ± 6	100 ± 1	**68 ± 3**	61 ± 12	86 ± 9	75 ± 10
CE-STV	71 ± 13	95 ± 10	93 ± 10	85 ± 5	93 ± 5	90 ± 7	99 ± 2	66 ± 2	62 ± 13	84 ± 7	75 ± 8
E-plu	84 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	98 ± 3	61 ± 1	70 ± 16	**93 ± 6**	76 ± 8
CE-plu	84 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	98 ± 3	61 ± 1	70 ± 16	**93 ± 6**	76 ± 8
E-Borda	83 ± 13	95 ± 9	**96 ± 8**	89 ± 4	93 ± 5	92 ± 5	**100 ± 0**	67 ± 3	66 ± 9	86 ± 10	75 ± 8
CE-Borda	83 ± 13	95 ± 9	**96 ± 8**	**89 ± 4**	93 ± 5	92 ± 5	**100 ± 0**	67 ± 3	66 ± 9	86 ± 10	75 ± 8
E-W_{power}B	85 ± 15	**96 ± 8**	96 ± 8	87 ± 6	96 ± 4	94 ± 6	**100 ± 0**	61 ± 3	69 ± 13	93 ± 7	75 ± 8
CE-W_{power}B	**86 ± 12**	**96 ± 8**	**96 ± 8**	88 ± 6	95 ± 3	94 ± 6	**100 ± 0**	61 ± 4	68 ± 16	92 ± 7	**79 ± 5**
E-W_{step}B	84 ± 14	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	99 ± 2	61 ± 1	69 ± 15	**93 ± 6**	76 ± 8
CE-W_{step}B	84 ± 11	**96 ± 8**	**96 ± 8**	87 ± 6	96 ± 5	94 ± 6	**100 ± 0**	60 ± 4	**74 ± 14**	92 ± 7	74 ± 9
E-W_{stair}B	85 ± 10	**96 ± 8**	**96 ± 8**	88 ± 6	96 ± 4	94 ± 6	**100 ± 0**	61 ± 2	68 ± 15	93 ± 6	77 ± 7
CE-W_{stair}B	85 ± 12	**96 ± 8**	**96 ± 8**	87 ± 6	97 ± 4	94 ± 6	**100 ± 0**	62 ± 4	69 ± 12	93 ± 6	76 ± 8

Table 8
Influence of FS on *F*1 score with AdaBoost classifier. Ten real-world datasets and one artificial dataset.

FS	ALO	BOR	BUR	CHIN	CHO	GOL	GOR	MHD	POM	SIN	TIA
*t*test	85 ± 5	92 ± 14	91 ± 12	82 ± 10	93 ± 5	91 ± 9	92 ± 5	65 ± 1	48 ± 11	92 ± 7	60 ± 10
RELIEF	78 ± 7	94 ± 10	90 ± 21	83 ± 5	93 ± 4	89 ± 7	94 ± 5	72 ± 3	59 ± 12	92 ± 6	**63 ± 10**
RFS	83 ± 10	92 ± 17	90 ± 21	80 ± 9	93 ± 10	92 ± 7	94 ± 4	74 ± 2	**64 ± 10**	87 ± 7	51 ± 7
Pearson	85 ± 5	91 ± 16	94 ± 9	82 ± 10	93 ± 5	91 ± 9	92 ± 4	65 ± 1	48 ± 11	92 ± 7	60 ± 10
MIC	82 ± 13	90 ± 17	92 ± 14	81 ± 2	**95 ± 6**	94 ± 6	**96 ± 5**	67 ± 2	52 ± 12	93 ± 6	51 ± 8
Fisher	85 ± 6	91 ± 12	87 ± 24	82 ± 10	93 ± 5	91 ± 9	91 ± 4	65 ± 1	48 ± 11	92 ± 7	60 ± 10
Gini	83 ± 12	91 ± 13	89 ± 18	81 ± 4	95 ± 6	95 ± 5	95 ± 5	73 ± 1	48 ± 9	90 ± 8	61 ± 8
ANOVA	84 ± 6	93 ± 13	89 ± 15	82 ± 10	93 ± 5	92 ± 8	92 ± 5	65 ± 1	48 ± 11	92 ± 7	60 ± 10
E-min	81 ± 16	90 ± 17	**96 ± 8**	84 ± 6	95 ± 6	92 ± 4	96 ± 5	69 ± 2	51 ± 12	94 ± 7	58 ± 9
CE-min	80 ± 17	94 ± 13	91 ± 17	84 ± 6	95 ± 6	91 ± 4	96 ± 5	69 ± 2	51 ± 12	94 ± 7	58 ± 9
E-max	79 ± 11	86 ± 20	87 ± 20	82 ± 8	96 ± 3	91 ± 8	93 ± 3	69 ± 2	55 ± 9	91 ± 7	62 ± 8
CE-max	79 ± 11	85 ± 15	88 ± 18	82 ± 8	96 ± 3	89 ± 10	94 ± 4	69 ± 2	55 ± 9	91 ± 8	63 ± 8
E-median	76 ± 19	92 ± 11	86 ± 25	84 ± 6	90 ± 7	91 ± 8	91 ± 3	67 ± 3	57 ± 10	92 ± 5	57 ± 8
CE-median	71 ± 18	93 ± 10	93 ± 12	84 ± 6	93 ± 7	92 ± 6	95 ± 4	72 ± 3	58 ± 14	92 ± 8	56 ± 10
E-mean	78 ± 10	88 ± 19	92 ± 16	81 ± 8	95 ± 5	90 ± 11	93 ± 5	70 ± 2	54 ± 15	89 ± 5	61 ± 8
CE-mean	81 ± 14	82 ± 20	90 ± 12	82 ± 7	96 ± 3	90 ± 8	94 ± 3	69 ± 2	54 ± 13	90 ± 5	61 ± 7
E-STV	77 ± 15	92 ± 14	93 ± 13	**85 ± 5**	91 ± 6	93 ± 7	94 ± 3	74 ± 3	56 ± 11	90 ± 7	57 ± 10
CE-STV	76 ± 15	86 ± 25	88 ± 20	82 ± 7	93 ± 5	89 ± 7	93 ± 4	**75 ± 2**	52 ± 14	90 ± 8	60 ± 9
E-plu	85 ± 5	87 ± 23	93 ± 11	82 ± 10	93 ± 5	91 ± 9	92 ± 4	65 ± 1	48 ± 11	93 ± 6	60 ± 10
CE-plu	**86 ± 5**	92 ± 13	92 ± 12	83 ± 10	92 ± 4	92 ± 8	93 ± 4	66 ± 2	49 ± 11	92 ± 7	60 ± 9
E-Borda	83 ± 14	91 ± 14	89 ± 18	84 ± 6	95 ± 6	94 ± 6	93 ± 4	73 ± 2	53 ± 9	93 ± 6	60 ± 8
CE-Borda	85 ± 13	96 ± 7	89 ± 23	84 ± 6	95 ± 6	94 ± 6	93 ± 4	73 ± 2	53 ± 9	93 ± 6	60 ± 8
E-W_{power}B	85 ± 11	91 ± 14	93 ± 11	81 ± 7	**97 ± 4**	93 ± 6	96 ± 5	67 ± 3	55 ± 13	94 ± 7	60 ± 7
CE-W_{power}B	81 ± 15	87 ± 19	92 ± 14	84 ± 7	96 ± 4	93 ± 6	96 ± 5	69 ± 3	56 ± 9	**95 ± 6**	59 ± 8
E-W_{step}B	85 ± 6	89 ± 19	94 ± 13	84 ± 9	93 ± 5	90 ± 10	92 ± 4	65 ± 2	51 ± 12	93 ± 7	60 ± 10
CE-W_{step}B	80 ± 14	90 ± 14	95 ± 9	83 ± 6	95 ± 6	94 ± 6	96 ± 5	71 ± 3	54 ± 13	92 ± 6	56 ± 7
E-W_{stair}B	81 ± 14	91 ± 15	90 ± 19	81 ± 7	95 ± 6	92 ± 7	94 ± 4	71 ± 3	62 ± 12	94 ± 4	61 ± 8
CE-W_{stair}B	75 ± 19	**97 ± 5**	92 ± 15	83 ± 6	95 ± 6	94 ± 6	96 ± 5	69 ± 3	57 ± 14	94 ± 5	57 ± 7

Table 9
Influence of FS on *F*1 score with naive Bayes classifier. Ten real-world datasets and one artificial dataset.

FS	ALO	BOR	BUR	CHIN	CHO	GOL	GOR	MHD	POM	SIN	TIA
*t*test	83 ± 13	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 4	94 ± 6	98 ± 3	61 ± 1	67 ± 15	**93 ± 6**	66 ± 10
RELIEF	84 ± 12	88 ± 20	88 ± 17	87 ± 5	86 ± 8	**95 ± 5**	99 ± 2	63 ± 5	62 ± 11	91 ± 7	64 ± 12
RFS	75 ± 8	75 ± 28	75 ± 28	85 ± 6	87 ± 6	94 ± 4	99 ± 3	63 ± 2	64 ± 10	90 ± 7	55 ± 7
Pearson	83 ± 13	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 4	94 ± 6	98 ± 3	61 ± 1	67 ± 15	**93 ± 6**	66 ± 10
MIC	84 ± 12	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 5	94 ± 6	99 ± 3	65 ± 2	63 ± 4	91 ± 9	55 ± 8
Fisher	83 ± 13	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 4	94 ± 6	98 ± 3	61 ± 1	67 ± 15	**93 ± 6**	66 ± 10
Gini	83 ± 11	**96 ± 8**	**96 ± 8**	87 ± 7	92 ± 11	94 ± 6	99 ± 3	63 ± 2	62 ± 15	92 ± 7	67 ± 11
ANOVA	83 ± 13	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 4	94 ± 6	98 ± 3	61 ± 1	67 ± 15	**93 ± 6**	66 ± 10
E-min	85 ± 14	**96 ± 8**	**96 ± 8**	87 ± 7	95 ± 5	94 ± 6	**100 ± 0**	61 ± 3	60 ± 14	91 ± 8	67 ± 7
CE-min	85 ± 14	**96 ± 8**	**96 ± 8**	87 ± 7	95 ± 5	94 ± 6	**100 ± 0**	61 ± 3	60 ± 14	91 ± 8	67 ± 7
E-max	83 ± 13	93 ± 10	93 ± 10	84 ± 5	96 ± 4	91 ± 6	**100 ± 0**	60 ± 2	63 ± 13	93 ± 6	63 ± 9
CE-max	83 ± 13	93 ± 10	93 ± 10	84 ± 5	96 ± 4	91 ± 6	**100 ± 0**	60 ± 2	63 ± 13	93 ± 6	63 ± 9
E-median	81 ± 14	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 5	94 ± 6	100 ± 1	59 ± 3	66 ± 14	**93 ± 6**	66 ± 9
CE-median	81 ± 14	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 4	94 ± 6	99 ± 2	59 ± 3	67 ± 12	**93 ± 6**	65 ± 10
E-mean	81 ± 14	**96 ± 8**	**96 ± 8**	85 ± 5	**98 ± 4**	94 ± 6	**100 ± 0**	61 ± 3	61 ± 13	91 ± 7	61 ± 10
CE-mean	81 ± 14	**96 ± 8**	**96 ± 8**	85 ± 5	96 ± 4	93 ± 5	**100 ± 0**	61 ± 2	63 ± 16	91 ± 7	61 ± 9
E-STV	73 ± 13	95 ± 10	95 ± 10	85 ± 6	94 ± 4	92 ± 6	100 ± 1	**68 ± 3**	56 ± 12	86 ± 9	65 ± 12
CE-STV	69 ± 13	95 ± 10	93 ± 10	84 ± 6	92 ± 5	90 ± 8	98 ± 3	66 ± 2	57 ± 13	84 ± 8	65 ± 10
E-plu	83 ± 13	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 4	94 ± 6	98 ± 3	61 ± 1	67 ± 15	**93 ± 6**	66 ± 10
CE-plu	83 ± 13	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 4	94 ± 6	98 ± 3	61 ± 1	67 ± 15	**93 ± 6**	66 ± 10
E-Borda	81 ± 14	95 ± 9	**96 ± 8**	**88 ± 5**	93 ± 5	91 ± 5	**100 ± 0**	67 ± 3	59 ± 12	86 ± 10	63 ± 8
CE-Borda	81 ± 14	95 ± 9	**96 ± 8**	88 ± 5	93 ± 5	91 ± 5	**100 ± 0**	67 ± 3	59 ± 12	86 ± 10	63 ± 8
E-W$_{power}$B	84 ± 15	**96 ± 8**	**96 ± 8**	86 ± 6	96 ± 4	94 ± 6	**100 ± 0**	61 ± 3	65 ± 12	93 ± 7	65 ± 8
CE-W$_{power}$B	**85 ± 13**	**96 ± 8**	**96 ± 8**	86 ± 7	95 ± 3	94 ± 6	**100 ± 0**	61 ± 4	63 ± 15	92 ± 7	**69 ± 6**
E-W$_{step}$B	83 ± 13	**96 ± 8**	**96 ± 8**	87 ± 7	96 ± 4	94 ± 6	99 ± 3	61 ± 1	65 ± 13	**93 ± 6**	66 ± 10
CE-W$_{step}$B	83 ± 11	**96 ± 8**	**96 ± 8**	85 ± 6	96 ± 5	94 ± 6	**100 ± 0**	60 ± 4	**70 ± 14**	92 ± 7	64 ± 10
E-W$_{stair}$B	84 ± 10	**96 ± 8**	**96 ± 8**	86 ± 7	96 ± 4	94 ± 6	**100 ± 0**	61 ± 2	65 ± 13	93 ± 6	68 ± 9
CE-W$_{stair}$B	84 ± 13	**96 ± 8**	**96 ± 8**	86 ± 7	97 ± 4	94 ± 6	**100 ± 0**	62 ± 4	65 ± 11	93 ± 6	67 ± 8

Table 10
Summary of results for prediction performance with AdaBoost classifier. Comparison of different FS methods.

FS method	Accuracy				F1 score			
	mean rank	stand. mean	med	WTL	mean rank	stand. mean	med	WTL
*t*test	18.0	−0.40	91.4	99/23/175	16.6	−0.27	90.8	114/19/164
RELIEF	13.5	0.11	90.5	157/2/138	13.9	0.09	89.0	155/0/142
RFS	14.7	−0.28	87.2	145/3/149	15.2	−0.26	87.2	141/0/156
Pearson	18.1	−0.34	91.3	98/23/176	17.0	−0.23	90.5	113/20/164
MIC	14.0	0.10	91.1	149/11/137	14.0	−0.02	89.9	151/7/139
Fisher	20.5	−0.65	88.5	73/20/204	19.1	−0.52	87.2	87/20/190
Gini	12.7	0.33	90.2	165/6/126	14.5	0.14	89.4	148/1/148
ANOVA	18.1	−0.39	90.4	96/23/178	17.0	−0.30	89.4	112/21/164
E-min	10.8	0.52	90.3	181/16/100	11.7	0.46	89.8	174/10/113
CE-min	11.0	0.38	91.7	179/15/103	12.1	0.32	90.6	170/10/117
E-max	16.8	−0.29	87.4	122/2/173	16.5	−0.24	86.0	126/2/169
CE-max	16.3	−0.27	86.6	128/2/167	16.4	−0.24	85.5	127/2/168
E-median	19.0	−0.58	87.9	98/1/198	18.5	−0.59	85.9	104/0/193
CE-median	12.3	0.20	91.9	172/1/124	12.6	0.13	91.9	169/0/128
E-mean	16.5	−0.34	88.9	126/0/171	17.0	−0.38	87.8	121/0/176
CE-mean	16.5	−0.26	84.5	125/2/170	15.9	−0.30	82.5	133/0/164
E-STV	12.0	0.24	89.9	175/1/121	13.3	0.10	89.8	162/0/135
CE-STV	18.6	−0.51	87.6	102/2/193	19.9	−0.58	86.3	88/2/207
E-plu	17.8	−0.41	87.9	103/19/175	16.7	−0.30	86.9	115/18/164
CE-plu	16.2	−0.20	92.1	128/4/165	14.8	−0.10	91.5	143/2/152
E-Borda	11.8	0.35	90.3	174/8/115	12.1	0.39	89.3	171/7/119
CE-Borda	9.7	0.55	90.4	196/10/91	10.7	**0.55**	88.6	186/8/103
E-W$_{power}$B	10.5	0.44	91.7	189/7/101	**9.7**	0.50	91.1	198/6/93
CE-W$_{power}$B	**9.0**	**0.59**	88.2	205/7/85	10.7	0.50	87.3	187/6/104
E-W$_{step}$B	16.3	−0.17	90.3	125/8/164	15.3	−0.06	89.4	137/6/154
CE-W$_{step}$B	12.1	0.43	91.5	170/10/117	13.0	0.34	90.5	162/6/129
E-W$_{stair}$B	11.7	0.37	91.1	177/5/115	10.9	0.41	90.4	188/0/109
CE-W$_{stair}$B	11.1	0.49	**92.2**	181/9/107	10.8	0.47	**91.9**	186/7/104

Table 11
Summary of results for prediction performance with naive Bayes classifier. Comparison of different FS methods.

FS method	Accuracy				F1 score			
	Mean rank	Stand. mean	Med	WTL	Mean rank	Stand. mean	Med	WTL
*t*test	11.8	0.22	93.1	123/114/60	11.6	0.24	**93.1**	125/110/62
RELIEF	14.5	−0.32	88.5	148/0/149	16.1	−0.39	87.1	131/0/166
RFS	20.3	−1.46	78.5	84/2/211	22.4	−1.69	75.1	61/2/234
Pearson	11.8	0.22	93.1	123/114/60	11.6	0.24	**93.1**	125/110/62
MIC	13.3	0.06	91.1	120/73/104	13.5	0.04	91.1	129/62/106
Fisher	11.8	0.22	93.1	123/114/60	11.6	0.24	**93.1**	125/110/62
Gini	15.4	0.05	92.1	100/77/120	14.0	0.18	92.1	120/69/108
ANOVA	11.8	0.22	93.1	123/114/60	11.6	0.24	**93.1**	125/110/62
E-min	12.0	0.35	91.3	137/78/82	11.6	0.27	91.3	142/77/78
CE-min	12.0	0.35	91.3	137/78/82	11.6	0.27	91.3	142/77/78
E-max	19.0	−0.43	91.7	85/30/182	18.8	−0.32	91.2	90/22/185
CE-max	19.0	−0.43	91.7	85/30/182	18.8	−0.32	91.2	90/22/185
E-median	13.3	0.27	93.1	117/80/100	12.9	0.30	**93.1**	127/79/91
CE-median	13.6	0.26	93.1	119/78/100	14.1	0.21	**93.1**	114/77/106
E-mean	15.9	−0.12	91.5	95/78/124	15.3	−0.03	91.5	101/78/118
CE-mean	17.3	−0.20	91.3	91/53/153	17.0	−0.13	91.3	95/53/149
E-STV	20.6	−0.54	86.6	81/1/215	20.6	−0.49	85.7	81/1/215
CE-STV	23.3	−1.27	85.5	51/1/245	23.2	−1.23	84.1	52/1/244
E-plu	11.8	0.22	93.1	123/114/60	11.6	0.24	**93.1**	125/110/62
CE-plu	11.8	0.22	93.1	123/114/60	11.6	0.24	**93.1**	125/110/62
E-Borda	16.7	−0.05	89.0	103/42/152	17.5	−0.16	87.8	94/42/161
CE-Borda	16.7	−0.05	89.0	103/42/152	17.5	−0.16	87.8	94/42/161
E-W_{power}B	13.1	0.32	92.7	128/71/98	12.9	0.32	92.7	131/71/95
CE-W_{power}B	12.0	**0.45**	92.3	140/73/84	11.7	0.42	92.3	144/71/82
E-W_{step}B	12.7	0.27	**93.1**	121/98/78	12.5	0.29	**93.1**	124/94/79
CE-W_{step}B	13.4	0.29	91.9	120/72/105	13.5	0.28	91.8	123/72/102
E-W_{stair}B	11.4	0.42	92.7	144/81/72	11.0	0.44	92.7	148/78/71
CE-W_{stair}B	**9.8**	0.44	92.7	165/70/62	**9.8**	**0.45**	92.7	165/70/62

section are high dimensional. Even though closer look at their prediction performance for MadelonHD dataset reveals that for this dataset their performance gets behind other methods. Simple ensembles based on max, mean, median achieved lower classification performance.

5. Conclusions

In this paper, we have proposed several ensemble FS methods based on voting aggregation. These methods utilize voting schemes such as the Borda count, STV, or plurality voting to aggregate the output of basic FS methods. In addition, we presented new clustered alternatives for ensemble FS. We conducted a series of experiment to evaluate the performance of FS methods from three different measures: FS sensitivity, stability, and classification performance. These experiments were executed on five artificial and ten high-dimensional real-world datasets. The new FS method based on clustered Borda count, outperformed other methods when we considered all performance metrics. Of the ensembles, (C)E-Borda, (C)E-W_{power}B, (C)E-W_{stair}B, (C)E-min, CE-W_{step}B outperformed other methods in Sen rate and prediction performance. However some of them did not performs well in stability. E-STV and C-STV performed very poorly, indicating that STV schemes are not suitable for building ensembles. Of the conventional methods, RELIEF scored well in FS sensitivity and stability; however, it performed under average in prediction performance. T-test FS rated highly in stability and average in prediction, but was below-average in sensitivity. Fisher FS obtained relatively balanced results in all measures, but these were only average when compared to ensembles.

The proposed concepts of ensemble FS and clustering proved to be efficient, but there are still some issues to consider. First, we plan to investigate the effect of the number of basal FS methods processed by ensemble FS. None of the work on ensemble FS has investigated how the number of inputs influences the result. In our study, we used 8 basic FS methods; however, previous studies usually utilized fewer. The key to determining the optimal number of rankers in an ensemble will be to analyze the similarity of ranking methods and select only those that improve the solution. Clustering achieved superior results in our experiments, so we believe it is a promising approach. The number and selection of FS methods remains a future research direction.

Here, we considered the mean-shift algorithm for clustering, since it has the advantage of determining the number of clusters during clustering. Other types of clustering algorithms influence how rankers are clustered, which in turn influences the resulting performance, so other clustering algorithms may be applied.

One apparent disadvantage of ensemble FS in general is its computational and time complexity. The need to perform multiple FSs poses increased requirements on computational resources. However, this task can be very efficiently distributed across multiple resources such that every FS method is assigned to a different resource and the results are then aggregated. This division significantly reduces computing time.

Even though, there are still some issues to consider, proposed approach confirms that ensemble FS methods are competent candidates for FS in broad range of potential applications.

Acknowledgements

This work was supported by the Slovak Research and Development Agency under contract APVV-16-0211.

References

[1] Tutorial on practical tips of the most influential data preprocessing algorithms in data mining, Knowledge-Based Systems 98 (2016) 1–29. 10.1016/j.knosys.2015.12.006
[2] S. Ahmed, M. Zhang, L. Peng, Prediction of detectable peptides in ms data using genetic programming, in: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO Comp '14, ACM, New York, NY, USA, 2014, pp. 37–38. 10.1145/2598394.2598421
[3] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, A. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. 96 (12) (1999) 6745–6750.
[4] J.C. Ang, A. Mirzal, H. Haron, H.N.A. Hamed, Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection, IEEE/ACM Trans. Comput. Biol. Bioinf. 13 (5) (2016) 971–989, doi:10.1109/TCBB.2015.2478454.
[5] W. Awada, T. Khoshgoftaar, D. Dittman, R. Wald, A. Napolitano, A review of the stability of feature selection techniques for bioinformatics data, in: Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on, 2012, pp. 356–363. 10.1109/IRI.2012.6303031
[6] F. Bach, Breaking the curse of dimensionality with convex neural networks, J. Mach. Learn. Res. 18 (19) (2017) 1–53. http://jmlr.org/papers/v18/14-546.html
[7] J.J. Bartholdi, J.B. Orlin, Single transferable vote resists strategic voting, Soc. Choice Welfare 8 (4) (1991) 341–354, doi:10.1007/BF00183045.
[8] R. Bellman, Adaptive Control Processes. A Guided Tour, Princeton Univ. Press, New Jersey, 1961.
[9] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowl. Inf. Syst. 34 (3) (2013) 483–519.
[10] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, Data classification using an ensemble of filters, Neurocomputing 135 (2014) 13–20, doi:10.1016/j.neucom.2013.03.067. http://www.sciencedirect.com/science/article/pii/S0925231213011405
[11] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J. Benítez, F. Herrera, A review of microarray datasets and applied feature selection methods, Inf. Sci. 282 (Supplement C) (2014) 111–135, doi:10.1016/j.ins.2014.05.042. http://www.sciencedirect.com/science/article/pii/S0020025514006021
[12] F. Borovecki, L. Lovrecic, J. Zhou, H. Jeong, F. Then, H.D. Rosas, S.M. Hersch, P. Hogarth, B. Bouzou, R.V. Jensen, D. Krainc, Genome-wide expression profiling of human blood reveals biomarkers for Huntington's disease,. Proceedings of the National Academy of Sciences of the United States of America 102(31), 11,023–11,028 (2005)
[13] M.E. Burczynski, R.L. Peterson, N.C. Twine, K.A. Zuberek, B.J. Brodeur, L. Casciotti, V. Maganti, P.S. Reddy, A. Strahs, F. Immermann, W. Spinelli, U. Schwertschlag, A.M. Slager, M.M. Cotreau, A.J. Dorner, Molecular classification of crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells, J. Mol. Diagnost. 8 (1) (2006) 51–61.
[14] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective, Neurocomputing 300 (2018) 70–79, doi:10.1016/j.neucom.2017.11.077. http://www.sciencedirect.com/science/article/pii/S0925231218302911
[15] T. Chan, G. Golub, R. LeVeque, Updating formulae and a pairwise algorithm for computing sample variances, Stanford university, 1979 Tech. rep..
[16] K. Chin, S. DeVries, J. Fridlyand, P.T. Spellman, R. Roydasgupta, W.L. Kuo, A. Lapuk, R.M. Neve, Z. Qian, T. Ryder, F. Chen, H. Feiler, T. Tokuyasu, C. Kingsley, S. Dairkee, Z. Meng, K. Chew, D. Pinkel, A. Jain, B.M. Ljung, L. Esserman, D.G. Albertson, F.M. Waldman, J.W. Gray, Genomic and transcriptional aberrations linked to breast cancer pathophysiologies, Cancer Cell 10 (6) (2006) 529–541.
[17] D. Chowdary, J. Lathrop, J. Skelton, K. Curtin, T. Briggs, Y. Zhang, J. Yu, Y. Wang, A. Mazumder, Prognostic gene expression signatures can be measured in tissues collected in RNAlater preservative, J. Mol. Diagnost. 8 (1) (2006) 31–39.
[18] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619, doi:10.1109/34.1000236.
[19] N. Dessi, B. Pes, Similarity of feature selection methods: an empirical study across data intensive classification tasks, Expert Syst. Appl. 42 (10) (2015) 4632–4642, doi:10.1016/j.eswa.2015.01.069. http://www.sciencedirect.com/science/article/pii/S0957417415000925
[20] J.F. Díez-Pastor, J.J. Rodríguez, C.I. García-Osorio, L.I. Kuncheva, Diversity techniques improve the performance of the best imbalance learning ensembles, Inf. Sci. 325 (Supplement C) (2015) 98–117, doi:10.1016/j.ins.2015.07.025. http://www.sciencedirect.com/science/article/pii/S0020025515005186
[21] P. Drotar, J. Gazda, Z. Smekal, An experimental comparison of feature selection methods on two-class biomedical datasets, Comput. Biol. Med. 66 (2015) 1–10, doi:10.1016/j.compbiomed.2015.08.010. http://www.sciencedirect.com/science/article/pii/S0010482515002917
[22] R.O. Duda, P.E. Hart, D.G. Stork, Patern classification, Willey, New York, USA, 2000.
[23] K. Dunne, P. Cunningham, F. Azuaje, Solutions to instability problems with sequential wrapper-based approaches to feature selection, Department of Computer Science, Trinity College, Dublin, Ireland, 2002 Tech. rep. tcd-cs-2002-28.
[24] R. Ge, M. Zhou, Y. Luo, Q. Meng, G. Mai, D. Ma, G. Wang, F. Zhou, Mctwo: a two-step feature selection algorithm based on maximal information coefficient, BMC Bioinform. 17 (1) (2016) 142, doi:10.1186/s12859-016-0990-0.
[25] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.
[26] G.J.G. Gordon, R.V.R. Jensen, L.L.L. Hsiao, S.R.S. Gullans, J.E.J. Blumenstock, S.S. Ramaswamy, W.G.W. Richards, D.J.D. Sugarbaker, R.R. Bueno, Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma, Cancer Res. 62 (17) (2002) 4963–4967.
[27] G. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. Inf. Theory 14 (1) (1968) 55–63, doi:10.1109/TIT.1968.1054102.
[28] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, Knowl. Inf. Syst. 12 (1) (2007) 95–116, doi:10.1007/s10115-006-0040-8.
[29] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1) (1997) 273–324.
[30] L.I. Kuncheva, A stability index for feature selection, in: Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi–Conference: Artificial Intelligence and Applications, AIAP'07, ACTA Press, Anaheim, CA, USA, 2007, pp. 390–395.
[31] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, A. Nowe, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinf. 9 (4) (2012) 1106–1119, doi:10.1109/TCBB.2012.33.
[32] J. Li, H. Liu, Challenges of feature selection for big data analytics, IEEE Intell. Syst. 32 (2) (2017) 9–15, doi:10.1109/MIS.2017.38.
[33] Y. Li, T. Li, H. Liu, Recent advances in feature selection and its applications, Knowl. Inf. Syst. 53 (3) (2017) 551–577, doi:10.1007/s10115-017-1059-8.
[34] L. Morán-Fernández, V. Bolón-Canedo, A. Alonso-Betanzos, Centralized vs. distributed feature selection methods based on data complexity measures, Knowl. Based Syst. 117 (2017) 27–45, doi:10.1016/j.knosys.2016.09.022. http://www.sciencedirect.com/science/article/pii/S0950705116303537

[35] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint $l2,1$-norms minimization, in: J.D. Lafferty, C.K.I. Williams, J. Shawe–Taylor, R.S. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems 23, Curran Associates, Inc., 2010, pp. 1813–1821.
[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[37] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, T.R. Golub, Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (6870) (2002) 436–442.
[38] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of relieff and RRelieff, Mach. Learn. 53 (1) (2003) 23–69, doi:10.1023/A:1025667309714.
[39] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, A. Alonso-Betanzos, Ensemble feature selection: homogeneous and heterogeneous approaches, Knowl. Based Syst. 118 (2017) 124–139, doi:10.1016/j.knosys.2016.11.017. http://www.sciencedirect.com/science/article/pii/S0950705116304749
[40] Y.N. Singh, S.K. Singh, A.K. Ray, Bioelectrical signals as emerging biometrics: issues and challenges, ISRN Signal Process. 2012 (1) (2012) 136–151, doi:10.5402/2012/712032.
[41] P. Somol, J. Novovičová, Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality, Pattern Anal. Mach. Intell., IEEE Trans. 32 (11) (2010) 1921–1939, doi:10.1109/TPAMI.2010.34.
[42] Z.D. Stephens, S.Y. Lee, F. Faghri, R.H. Campbell, C. Zhai, M.J. Efron, R. Iyer, M.C. Schatz, S. Sinha, G.E. Robinson, Big data: astronomical or genomical? PLoS Biol. 13 (7) (2015), doi:10.1371/journal.pbio.1002195. E1002195
[43] J. Tang, H. Liu, Feature selection for social media data, ACM Trans. Knowl. Discov. Data 8 (4) (2014) 19:1–19:27, doi:10.1145/2629587.
[44] E. Tian, F. Zhan, R. Walker, E. Rasmussen, Y. Ma, B. Barlogie, J.D. Shaughnessy Jr., The role of the wnt-signaling antagonist DKK1 in the development of osteolytic lesions in multiple myeloma, N. Engl. J. Med. 349 (26) (2003) 2483–2494.
[45] S.M. Vieira, L.F. Mendonça, G.J. Farinha, J.M. Sousa, Modified binary {PSO} for feature selection using {SVM} applied to mortality prediction of septic patients, Appl. Soft Comput. 13 (8) (2013) 3494–3504, doi:10.1016/j.asoc.2013.03.021. http://www.sciencedirect.com/science/article/pii/S1568494613001361
[46] F. Wu, Y. Han, X. Liu, J. Shao, Y. Zhuang, Z. Zhang, The heterogeneous feature selection with structural sparsity for multimedia annotation and hashing: a survey, Int. J. Multimed. Inf. Retr. 1 (1) (2012) 3–15, doi:10.1007/s13735-012-0001-9.
[47] 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, doi:10.1109/TEVC.2015.2504420.
[48] J. Zhu, H. Zou, S. Rosset, T. Hastie, Multi-class adaboost, Stat. Interface 2 (3) (2009) 349–360.