Journal of Soft Computing and Artificial Intelligence

Journal homepage: https://dergipark.org.tr/en/pub/jscai

International Open Access

Volume 05
Issue 01

June, 2024

Research ***Article***

A New Fast Filter-based Unsupervised Feature Selection Algorithm Using Cumulative and Shannon Entropy

Samet Demirel[1*], ***Fatih Aydın***[2]

[1]*Distance Education Application and Research Center. Balıkesir University, 10145, Balıkesir, Türkiye*
[2]*Department of Computer Engineering, Faculty of Engineering. Balıkesir University, 10145, Balıkesir, Türkiye*

ARTICLE INFO

Article history:
Received **April 03, 2024**
Revised **May 05, 2024**
Accepted **May 27, 2024**

Keywords:
Machine Learning
Unsupervised Feature Selection
Univariate-filter Approach
Cumulative Entropy
Shannon Entropy

ABSTRACT

The feature selection process is indispensable for the machine learning area to avoid the curse of dimensionality. Hereof, the feature selection techniques endeavor to handle this issue. Yet, the feature selection techniques hold several weaknesses: (i) the efficacy of the machine learning methods could be quite different on the chosen features (ii) by depending on the selected subset, substantial differences in the effectiveness of the machine learning algorithms could also be monitored (iii) the feature selection algorithms can consume much time on massive data. In this work, to address the issues above, we suggest a new and quick unsupervised feature selection procedure, which is based on a filter and univariate technique. The offered approach together regards both the Shannon entropy computed by the symmetry of the distribution and the cumulative entropy of the distribution. As a consequence of comparisons done with some cutting-edge feature selection strategies, the empirical results indicate that the presented algorithm solves these problems in a better way than other methods.

1. Introduction

Machine learning algorithms suffer from high-dimensional data sets. In this respect, the Feature Selection (FS) algorithms would be a supporting element for reconstructing the model fast and increasing its performance. FS is the task of determining features that allow preserving or, in some data sets, enhancing the model performance without needing the use of all original features [1]. FS is beneficial in learning tasks such as classification, regression, or clustering since as well as decreasing the storage and computing requirements, it affords to dismiss the curse of dimensionality [2] and allows to form of models that have better generalization ability [3]. Thus, the feature subset that best represents the original data set is selected. The selected features refer to information that affects the model outcome and cannot be provided by other features.
Figure **1** describes this process.

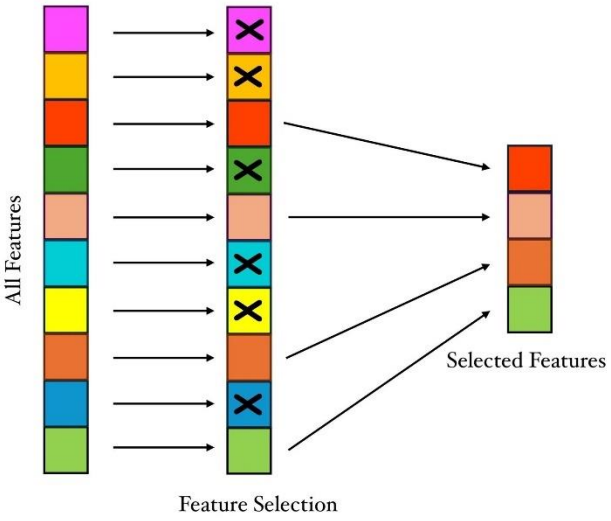

Figure 1 The feature selection scheme.

[1] * Corresponding author
e-mail: sametdemirel@balikesir.edu.tr
DOI: 10.55195/jscai.1464638

FS algorithms are divided into three as supervised, semi-supervised, and unsupervised, in terms of the use of class information. Unsupervised Feature Selection (UFS) algorithms have three significant supremacies: (i) they are unbiased, (ii) they can process data even when prior knowledge is unavailable, and (iii) they can decrease over-fitting in contrast to supervised ones [1]. FS algorithms are separated into four basic approaches: filter, wrapper, hybrid, and embedded, according to the selection strategy of features [4].

Figure 2 shows the categorization of the feature selection methods in terms of the use of class information and the selection strategies.

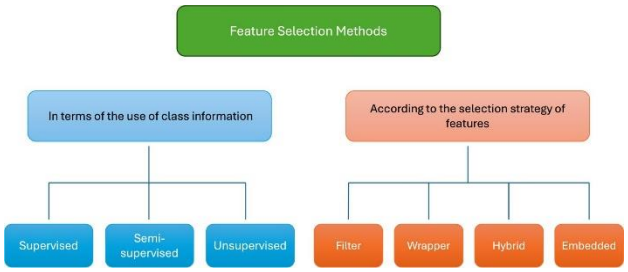

Figure 2 The categorization of the feature selection methods.

The filter approach focuses on the intrinsic and statistical properties of data sets. Hence, they are rather fast in comparison to the other approaches. The wrapper technique is based on the machine learning algorithm selected. Therefore, they are slower than the filter approach. The hybrid strategy incorporates the filter and wrapper approaches. Finally, the embedded methods simultaneously perform the related learning task and feature selection.

In the last decades, hundreds of remarkable UFS algorithms have been introduced. These unsupervised feature selection algorithms address troubles in the subfields such as big data, heterogeneous attributes, high-dimensional data sets, image processing, data clustering, categorical data sets, rule induction, text mining, and biomarker discovery. Besides, there exist various approaches employed to develop feature selection algorithms in the literature. These techniques are adaptive graph-based approach, adaptive similarity learning, autoencoder, bio-inspired approach, clustering, differential evolution, Dirichlet process, discriminative analysis, extreme learning machine, graph representation, Gravitational Search Algorithm, hidden Markov model, Hilbert-Schmidt independence criterion, integer programming, Kolmogorov-Smirnov test, k-nearest neighbors, Laplace score, latent representation, Local Sensitive Dual Concept Learning, local structure learning, Locality Preserving Projection, manifold learning, matrix factorization, Maximal Information Compression Index, metaheuristic algorithms, metric learning, mutual information, nonparametric Bayesian mixture model, particle swarm optimization, principal component analysis, regression-based approach, self-representation learning, sparse learning, spectral learning, statistical learning, subspace learning, and symmetrical uncertainty.

We categorize the unsupervised feature selection algorithms in the literature in terms of the techniques they have applied. Accordingly, in the context of neighborhood relationships, LS (Laplacian score for unsupervised feature selection) [5] uses the locality-preserving capability by finding the nearest neighbors of each feature and thereby selects features. RNE (Robust Neighborhood Embedding) characterizes the local geometry of the data by linear coefficients that rebuild each point via k-nearest neighbors to get the weight matrix and it solves the model based on the Taxicab-norm through the alternation direction method of multipliers [6]. According to clustering approaches, MCFS (Multi-Cluster Feature Selection) [7] conserves the multi-cluster structure of the data by solving a sparse eigenproblem and a least-squares problem and thus selects relevant features.

As for self-representation approaches, RSR (unsupervised feature selection method based on Regularized Self-Representation) [8] selects features by inducing low-rank representation in subspace clustering where any feature can be reproduced as the linear combination of other convenient features. DISR (feature selection method via Diversity-Induced Self-Representation) [9] selects features by reducing redundant features based on diversity and the internal self-representation characteristic of features. In respect of the use of information-theoretic approaches, IUFS (Information-theoretic Unsupervised Feature Selection) [10] aims to maximize the cooperation information between features selected by solving an optimization problem, searching local optima by a greedy approach. DUFS (Pairwise Dependence-based Unsupervised Feature Selection) [11] selects the dependent features by measuring the mutual information between features via a joint entropy and by solving an optimization problem. In terms of spectral learning, SPEC (the SPECtrum decomposition of the Laplacian matrix) [12] suggests a unified framework that relies on spectral graph theory for both unsupervised and supervised tasks. In point of random subspace

learning, SRCFS (unsupervised Feature Selection approach based on multi-Subspace Randomization and Collaboration) [13] carries out feature assessment in each random subspace by generating lots of them and subsequently merges the information from multiple subspaces to obtain an entire feature ranking vector.

In terms of utilizing feature similarity, EUFSFC (Efficient Unsupervised Feature Selection method through Feature Clustering) [14] performs feature selection by extending the Fitness Proportionate Sharing clustering by two feature similarity criteria such as Maximal Information Compression Index and Symmetrical Uncertainty.

With respect to the use of pseudo-labels, USFS (Unsupervised Soft-label Feature Selection) [15] focuses on alleviating the effect of noisy data and outliers, and the use of soft labels to be consistent with inexplicit data distribution. It uses an iterative approach to solve optimization problems.

In this research, we present a fast and simple unsupervised feature selection algorithm. The proposed algorithm jointly considers the cumulative effect, symmetry, and deviation of the distribution, and it has obtained significant results on the training data used in the experiments. Finally, the prominent contributions of this paper are as follows:

- The suggested algorithm runs quickly compared to the other methods and it is easy to implement.
- Regardless of the classifiers and data domains, the offered method largely keeps yielding the highest classification accuracy on average as the number of selected features rises.
- The presented method requires no parameter to operate.

The rest of the sections are organized in the following: in Section 2, we describe our algorithm. In Section 3, we explain the experimental setup. We report in detail the results in Section 4. Lastly, we explain the conclusions of the paper in Section 5.

2. Proposed Method

In this section, we introduce our method based on the cumulative entropy [16] and Shannon entropy [17].

2.1 Description of the algorithm

The proposed algorithm is composed of three stages. Given a training set $X = \{x_i\}_{i=1}^{m} \Rightarrow x_i = \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)}\right) \in \mathbb{R}^d, i = 1, \dots, m$, where m is the number of instances and d is the number of features. In the first stage, the cumulative entropy of each feature is computed by Eq (1) after finding their normal cumulative distribution function values given by Eq (2) for a continuous random variable $x^{(k)}$ with a normal probability density function $f_{x^{(k)}}(x)$.

$$CE\left(x^{(k)}\right) = -\sum_i F\left(x_i^{(k)}\right) \log_2 F\left(x_i^{(k)}\right) \quad (1)$$

$$F(x; \mu, \sigma) = \frac{1}{2}\left(1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) \quad (2)$$

where $erf(\cdot)$ denotes the error function of the normal distribution and it is given by

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{z} e^{-t^2} dt \quad (3)$$

Subsequently, the features are sorted in ascending entropy order, $\mathrm{A} = \left\{a_j | a_j \in \{1, \dots, d\}, w_{a_j} \in CE\left(x^{(k)}\right), k \in \{1, \dots, d\}, j \in \{1, \dots, d\}, w_{a_j} \leq w_{a_{j+1}}\right\}$. The entropy of the cumulative distribution function specifies the number of bits required to represent the variables from the probability distribution to a random variable drawn. From another perspective, the frequent occurrence values of X are represented by the least bits, while the sparse ones are expressed by more bits. Thus, the first stage of our algorithm relies on the assumption that a feature owning the least number of bits needed is of the greatest importance.

In the second step, the Shannon entropy of the features is computed in terms of the symmetry in the distribution. To this end, we determine a border, according to the maximum of the three measures of central tendency (i.e., $mean$, $median$, and $mode$) and designate the $mean$, the $median$, and the $mode$ as $\overline{x^{(k)}}$, $\widetilde{x^{(k)}}$, and $\widehat{x^{(k)}}$, respectively and denote the maximum of the three measures of central tendency as

$$\rho^{(k)} = arg\,max\left(\overline{x^{(k)}}, \widetilde{x^{(k)}}, \widehat{x^{(k)}}\right) \quad (4)$$

Next, we transform the original data set into a sparse matrix by using the function given by

$$u\left(x_i^{(k)}\right) = \begin{cases} 0, & x_i^{(k)} < \rho^{(k)} \\ 1, & x_i^{(k)} \geq \rho^{(k)} \end{cases} \quad (5)$$

We compute the entropy of each feature on the transformed data set and sort them in descending entropy order. Thus, the features with the highest entropy are selected. The second stage aims to measure the entropy of the skewness of the distribution by Eq (6).

$$H\left(x^{(k)}\right) = -\sum_{v\in\{0,1\}} \frac{|u(x^{(k)})=v|}{|u(x^{(k)})|} log_2 \frac{|u(x^{(k)})=v|}{|u(x^{(k)})|} \quad (6)$$

According to the assumption in the second stage, the features with the highest entropy are of the greatest importance, namely, $B = \left\{b_j | b_j \in \{1,\dots,d\}, w_{b_j} \in H\left(x^{(k)}\right), k \in \{1,\dots,d\}, j \in \{1,\dots,d\}, w_{b_j} \geq w_{b_{j+1}}\right\}$.

In the last stage, we fuse these two outputs (i.e., A and B sets) obtained from the first two stages. The outputs are in order of the importance of features. We obtain the ultimate order of features through the geometric mean of their positions as shown in Eq (7).

$$w_{j=1,\dots,d} = \sqrt{\sum_j j\mathbf{1}_{A_j}(j) \sum_j j\mathbf{1}_{B_j}(j)} \quad (7)$$

$$\mathbf{1}_{A_j}(j) = \begin{cases} 0, & A_j \neq j \\ 1, & A_j = j \end{cases} \quad (8)$$

Now, we describe the suggested unsupervised feature selection technique in Algorithm 1 and call it the Entropy-based Feature Selection (EFS). We are now ready to calculate the time complexity of the algorithm. In the first stage, the time complexity is $O(2md + dlog_2 d)$ in the average or best case and $O(2md + d^2)$ in the worst case. The second stage is calculated with time complexity $O(3md + dlog_2 d)$ in the average or best case and $O(3md + d^2)$ in the worst case. The last stage is calculated with time complexity $O(1 + dlog_2 d)$ in the best case, $O(d^2 + dlog_2 d)$ in the average case, and $O(2d^2)$ in the worst case. Thus, the overall time complexity of the algorithm is found as $O(5md + 3dlog_2 d + 1)$ in the best case, $O(5md + 3dlog_2 d + d^2)$ in the average case, and $O(5md + 4d^2)$ in the worst case. To sum it up, the time complexity of the algorithm is linear when $m \gg d$, linearithmic when $d \gg m$, and quadratic when $m \approx d$ for the best case. The time complexity of the algorithm is linear when $m \gg d$ and quadratic when $d \gg m$ or $m \approx d$ for the average case. The time complexity of the algorithm is linear when $m \gg d$ and quadratic when $d \gg m$ or $m \approx d$ for the worst case. As a result, the running time of the algorithm ranges from linear to quadratic, bounding up with the input data.

Algorithm 1 Entropy-based Feature Selection (EFS)

Input

$X = \{x_i\}_{i=1}^m \Longrightarrow x_i = \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)}\right)$

Output

I $\in \mathbb{R}^d$: The ranked feature indicator vector

1: Calculate the cumulative distribution function values $P \in \mathbb{R}^{m\times d}$ of the input data X by Eq (2).
2: Calculate the cumulative entropy $c \in \mathbb{R}^d$ of P by Eq (1).
3: Sort c in ascending order and calculate the feature indicator vector $k \in \mathbb{R}^d$ for the first stage.
4: Calculate the maximum of the three measures of central tendency $\boldsymbol{\rho} \in \mathbb{R}^{\boldsymbol{d}}$ by Eq (4).
5: Transform the original input data X into an undirected binary graph $T \in \mathbb{R}^{m\times d}$ by Eq (5).
6: Calculate the Shannon entropy $h \in \mathbb{R}^d$ of T by Eq (6).
7: Sort h in descending order and calculate the feature indicator vector $t \in \mathbb{R}^d$ for the second stage.
8: Calculate the ranking vector $\mathbf{r} \in \mathbb{R}^{\boldsymbol{d}}$ by Eq (7) taking the first feature indicator vector k and the second feature indicator vector t as an argument.
9: return the ranked feature indicator vector I by sorting r in ascending order.

2.2 Determination of the number of selected features

We have derived a lower bound for determining the fitting number of the selected features as assessing the methods. To find the expression, we should make some assumptions. Accordingly, let ϵ be the error rate of a classification algorithm on the whole input data. No classifiers that can learn cannot have a less accuracy rate than a random predictor. Then, let us delimit the accuracy rate of the classification algorithm by the accuracy rate of the random predictor. The accuracy rate of the majority predictor is equal to n/m, where n is the number of the majority class. The error rate of the majority predictor is $1 - \frac{n}{m}$. Also, the error rate of a majority predictor on each feature is $1 - \frac{n}{m}$. Now, let us assume that the features are independent of each other. In that case, the error rate is $\left(1 - \frac{n}{m}\right)^{d'}$ for the first d' features. Accordingly, let us find d' that satisfies the inequality given by Ineq (1).

$$\left(1 - \frac{n}{m}\right)^{d'} \leq \epsilon \quad (1)$$

Since $\left(1 - \frac{n}{m}\right) \leq e^{-n/m}$, we arrive at Ineq (2).

$$-\frac{m}{n} ln\, \epsilon \leq d' \quad (2)$$

The error rate of at least one classifier on an input data with at least d' features that are intentionally

selected is approximately ϵ . Furthermore, the empirical results confirm this outcome, as well. In this respect, it is sufficient to use a few features while evaluating the UFS algorithms.

In addition to the abovementioned situation, let us consider the features that any two UFS algorithms rank in descending importance order and try to calculate the similarity probability of the first k features of these two sets. Accordingly, the number of ordered arrangements of k out of d features is given by Eq (9).

$$P_k^d = \frac{d!}{(d-k)!} \tag{9}$$

The number of ordered arrangements of k features is $k!$. Thus, the similarity probability of the first k features of these two sets is given by Eq (10).

$$P_{similarity} = \frac{k!(d-k)!}{d!} \tag{10}$$

The results show that the similarity probability decreases as the number of features increases. Therefore, at most $d-1$ features can be selected to evaluate the UFS algorithms. Consequently, the number of features can be picked in the range of $-\frac{m}{n}\ln\epsilon$ to $d-1$. In this study, we chose the number of features in the range of 1 to 15 to indicate the change in the lower bound.

3. Experimental Setup

In this section, we explain the methodology followed in this paper for analyzing the UFS methods used in the experiments. We perform the whole tests under ten-fold cross-validation and carry out each test ten times to be able to use different training data within each fold combination. The experiments have been performed in the MATLAB R2021a on an i7-6700HQ CPU at 2.6 GHz with 16 GB of RAM on Windows 10 Pro (64-bit).

In this study, twelve training sets from the different domains are used. Table 1 shows the descriptive information of the training sets.

Table 2 shows a baseline, two conventional, and eight cutting-edge unsupervised feature selection algorithms used in the experiments. In Section 5, we show the empirical results in terms of classification by using Random Forest (RF), Classification and Regression Trees (CART), Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Naive Bayes (NB). Then, we put forth the results in terms of runtime.

Table 1 The characteristics of the data sets used in experiments (m is the number of instances, d is the number of features, c is the number of classes, and r is the imbalance ratio)

#	Data set	m	d	c	r	Domain
1	cardiotocography[2]	2126	21	3	9.40	Medical
2	climate model2	540	18	2	10.73	Climate
3	colon[3]	62	2000	2	1.82	Biological
4	connectionist bench2	208	60	2	1.14	Sonar
5	diabetic retinopathy2	1151	19	2	1.13	Image
6	dna[4]	3186	180	3	2.16	Biological
7	ecoli-uni[5]	336	343	8	71.50	Biological
8	flowmeterA2	87	36	2	1.49	Fault detection
9	madelon2	2000	500	2	1.00	Artificial
10	qsar biodegradation2	1055	41	2	1.96	Chemical
11	vehicle2	846	18	4	1.10	Image
12	wall following robot2	5456	24	4	6.72	Teleinformatics

Table 2 The unsupervised feature selection techniques used in the experiments

#	Method	Approach	Category	Technique
1	All features	—	—	—
2	DISR[6]	Filter	Multivariate	Diversity and the internal self-representation
3	DUFS[7]	Filter	Multivariate	Joint entropy
4	IUFS[6]	Filter	Multivariate	The alternative conditional expectation and the generalized maximal correlation
5	LS[8]	Filter	Univariate	Laplacian eigenmaps and LPP
6	MCFS[8]	Filter	Multivariate	Spectral embedding and sparse learning
7	RNE[9]	Filter	Multivariate	The locally linear embedding
8	RSR[6]	Filter	Multivariate	Regularized self-representation

[2] https://archive.ics.uci.edu/ml/datasets/
[3] https://jundongl.github.io/scikit-feature/datasets.html
[4] https://www.openml.org/d/40670
[5] https://github.com/wang-feifei/USFS-code/tree/master/Datasets
[6] https://github.com/CAU-AIR-Lab/DUFS/tree/main/programs
[7] https://github.com/CAU-AIR-Lab/DUFS
[8] http://www.cad.zju.edu.cn/home/dengcai/Data/MCFS.html
[9] https://github.com/liuyanfang023/KBS-RNE

9	SRCFS[10]	Filter	Multivariate	Balanced multi-subspace randomization
10	SPEC[11]	Filter	Univariate	Spectral graph theory
11	USFS[12]	Filter	Multivariate	Soft-label learning

4. Findings and Discussion

In this section, we assess the performance of the offered algorithm through classification experiments. Figure 3 shows the change in the cumulative entropy of the features, depending on the number of instances. The entropy of the cumulative distribution function specifies the number of bits that need to characterize the variables from the probability distribution to a random variable drawn. From another perspective, the frequent occurrence values of X are represented by the least bits, while the sparse ones are expressed by more bits. Thus, the first stage of our algorithm relies on the assumption that a feature owning the least number of bits needed is of the greatest importance. Figure 4 shows the change in the Shannon entropy of the symmetry of the distribution in each feature, depending on the number of instances.

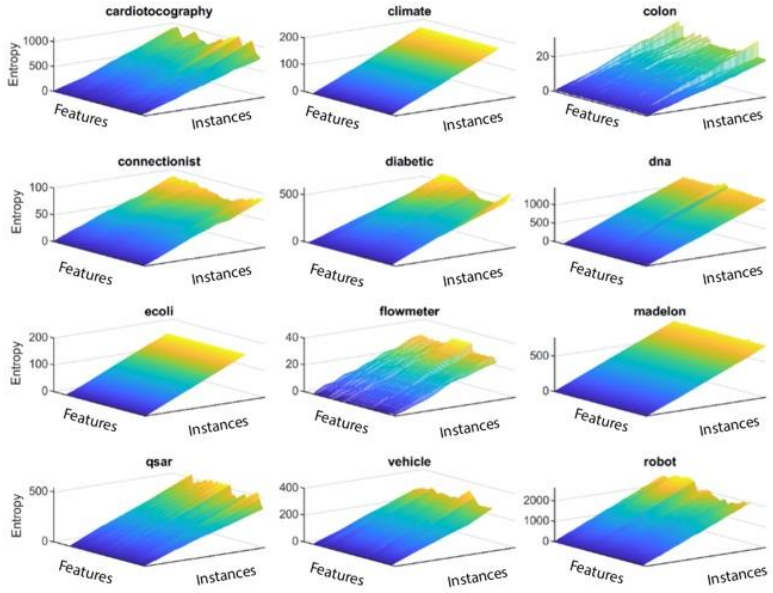

Figure 3 The variation of cumulative entropies of the features, in terms of the number of the instances

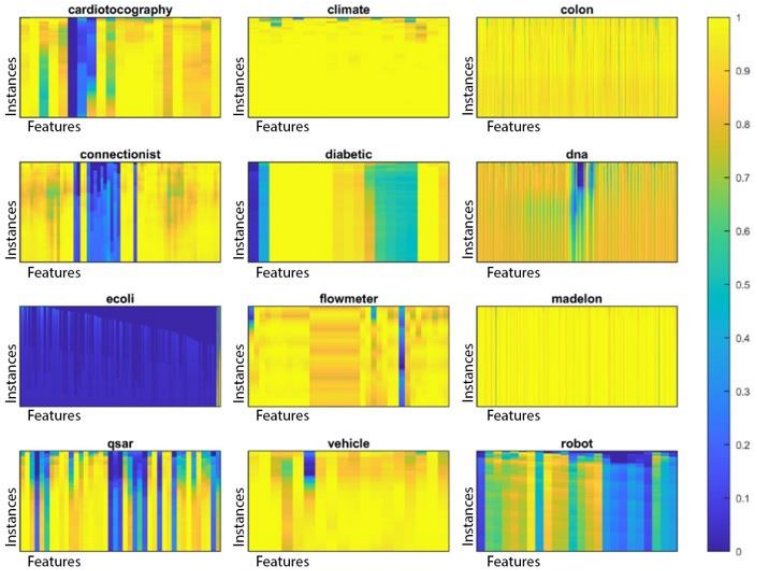

Figure 4 The variation in the Shannon entropy of the symmetry of the distribution in each feature, in terms of the number of the instances

[10] https://github.com/huangdonghere/SRCFS
[11] https://github.com/matrixlover/LSLS
[12] https://github.com/wang-feifei/USFS-code

Figure 5 demonstrates the comparison results of the UFS methods according to the average of five classifiers on all the data sets. According to the results, EFS, IUFS, and LS have the statistically significant highest ACC with 0.783, 0.774, and 0.771, respectively. USFS has the lowest ACC with 0.695. In addition, EFS, IUFS, and LS exceed the baseline that has 0.765 of ACC. Finally, EFS, IUFS, and LS deliver the average highest ACC with statistical significance. Figure 6 shows the average results of the UFS algorithms on all the classification experiments in terms of the average ACC and maximum ACC. From the results, EFS has the highest ACC with 0.748 in terms of Average and the highest ACC with 0.803 in terms of Maximum. Considering all features, the average ACC is 0.777. The second-best results belong to IUFS with 0.725 and 0.783 in terms of Average and Maximum. Finally, the third-best results belong to LS with 0.712 and 0.773 in terms of Average and Maximum. The results of EFS, IUFS, and LS are statistically more significant than others.

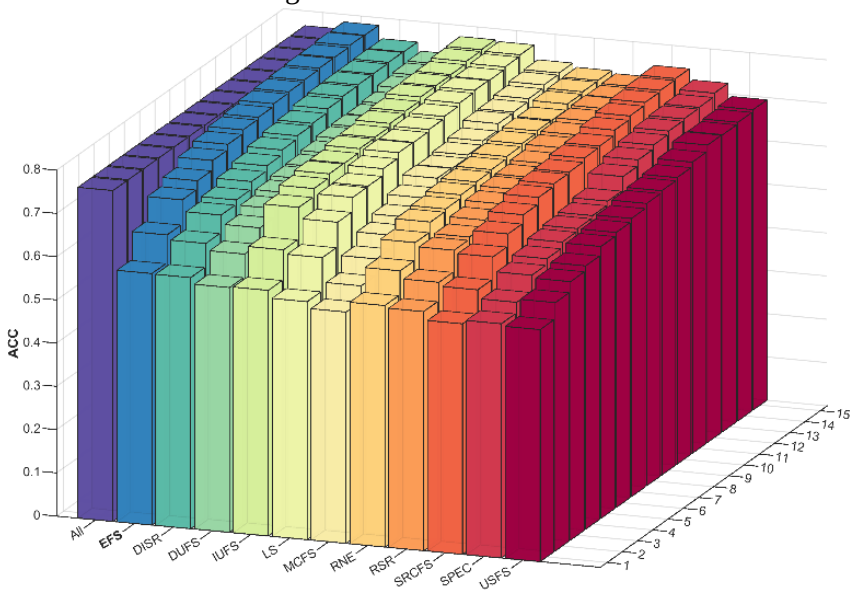

Figure 5 The comparative results of the UFS methods according to the average of five classifiers on all the data sets

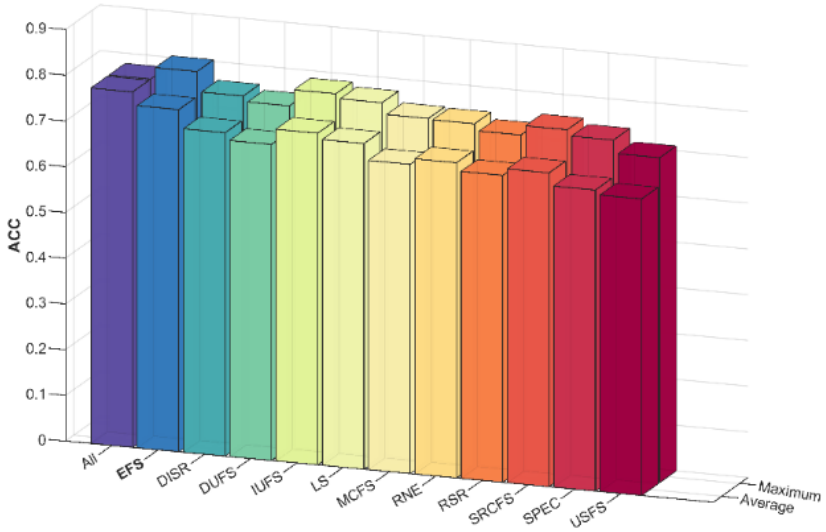

Figure 6 The performance of the UFS algorithms in terms of Maximum and Average, considering the results belonging to the five classifiers on twelve data sets

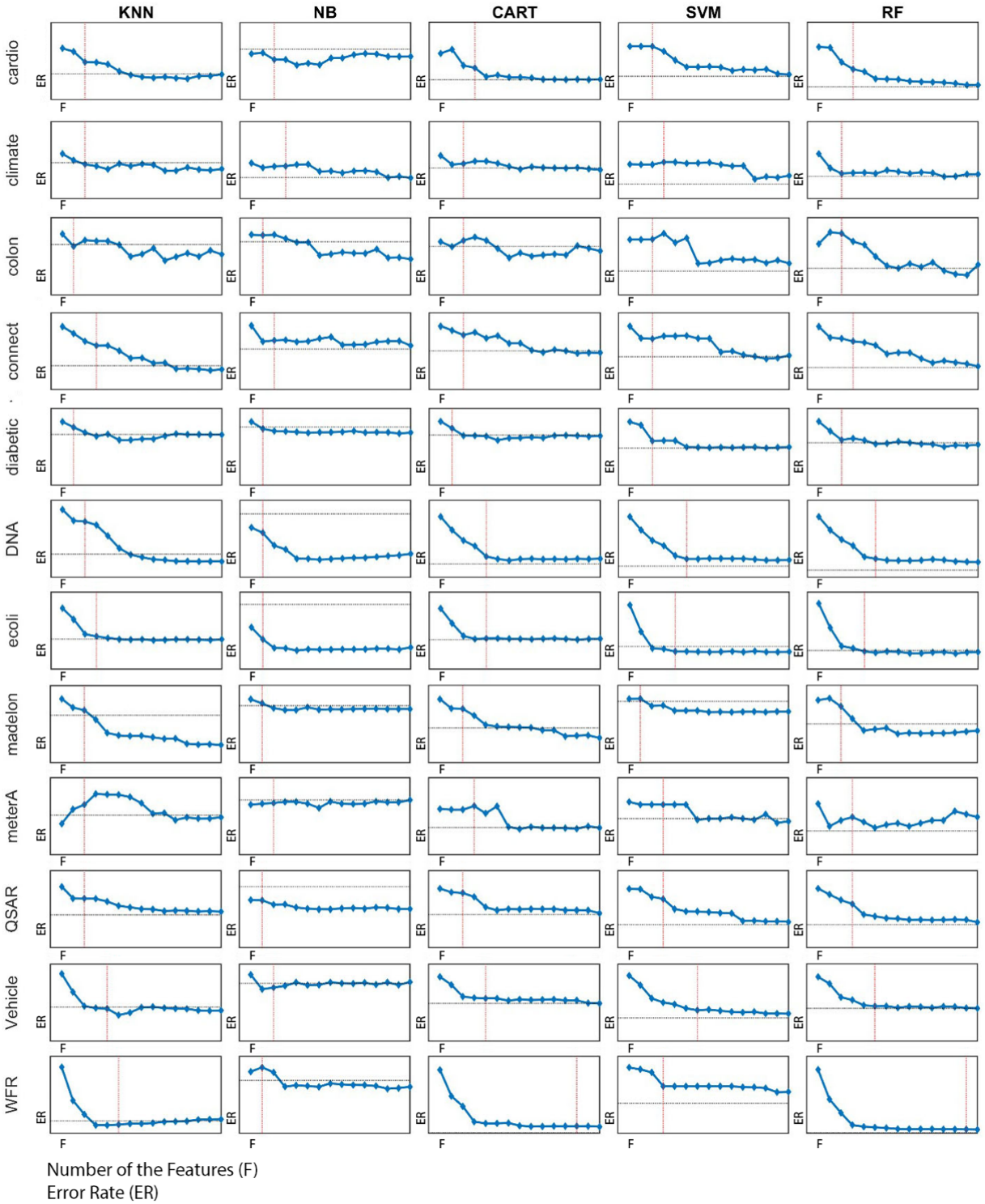

Figure 7 The variation in the minimum error rate, in terms of the number of the features on the data sets (The horizontal black dashed lines denote the error rate of the whole input data. The vertical red dashed lines denote the number of the features obtained by Ineq. (2)).

Figure 7 shows the variation in the minimum error rate, in terms of the number of features on the data sets. From the results, we can observe the d' number of features whose error rate is close to the error rate of the whole input data and larger than the global minimum error rate depending on the number of the selected features. This is a lower bound. Besides, it is difficult to decide an upper bound for the optimum number of features, due to the unpredictable relations formed by the combination of features. But a global error rate can be searched through advancement in a certain step (i.e., an optimized iterative forward search) by starting from the lower bound. Thus, there is no need to check for all possible subsets.

According to the results, we can reach the minimum error rates in several steps by beginning from a lower bound. In other words, there is not mostly necessary to search for lots of features to arrive at the global minimum. The results on three high-dimensional data sets also verify this situation. However, we would like to underline that this situation cannot be generalized to all data sets, as well. Hence, we would like to state that an analysis of many features (e.g., $d-1$) can be performed.

Considering the results in more detail, the numerical results of the suggested algorithm against the state-of-the-art algorithms are shown in Table 3, Table 4, Table 5, Table 6, and Table 7. Table 3 shows the average classification accuracies in terms of the KNN classifier, and it shows that the proposed algorithm reaches the maximum accuracy in 5 out of 12 data sets. Table 4 shows the average classification accuracies of the algorithms in terms of the NB classifier. This table demonstrates too likewise that the proposed algorithm attains the maximum accuracy in 5 out of 12 data sets. Table 5 demonstrates the average classification accuracies of the algorithms using the CART classifier. This table points out that the proposed algorithm achieves the highest accuracy in 3 out of 12 data sets. Table 6 exhibits the average classification accuracies obtained by the algorithms using the SVM classifier. This table also indicates that the proposed algorithm achieves higher accuracy compared to the others in 3 out of 12 data sets. Finally, Table 7 contains the average classification accuracies of the algorithms in terms of the RF classifier. This table also demonstrates that the proposed algorithm obtains higher accuracies than the other algorithms in 4 out of 12 data sets. Considering the whole results in the five tables, the offered algorithm delivers the highest average accuracy in 20 out of 60 experiments. DISR has the highest average accuracy in 9 out of 60 experiments. To sum up, the offered method succeeds the highest total average accuracy over all classifiers.

Table 3 The results of average classification accuracy of the offered and cutting-edge algorithms (KNN classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	**0.869**	0.853	0.855	0.792	0.867	0.831	0.858	0.856	0.854	0.800	0.867
2	0.878	0.861	0.852	**0.882**	0.869	0.865	0.866	0.853	0.878	0.865	0.860
3	**0.657**	0.561	0.599	0.594	0.494	0.650	0.641	0.491	0.545	0.568	0.631
4	**0.748**	0.695	0.736	0.737	0.727	0.555	0.697	0.702	0.680	0.571	0.727
5	**0.618**	0.594	0.597	0.600	0.614	0.573	0.608	0.605	0.606	0.573	0.591
6	**0.654**	0.626	0.322	0.591	0.636	0.321	0.265	0.468	0.264	0.534	0.291
7	0.766	**0.776**	0.755	0.765	0.758	0.635	0.775	0.426	0.768	0.426	0.426
8	0.632	0.526	0.499	0.580	0.722	0.517	0.499	0.504	0.759	**0.761**	0.498
9	0.720	0.702	0.689	0.714	0.734	0.712	0.702	0.688	0.691	**0.772**	0.705
10	0.737	**0.768**	0.745	0.713	0.704	0.724	0.765	0.766	0.530	0.485	0.724
11	0.631	**0.635**	0.615	0.554	0.572	0.569	0.608	0.625	0.628	0.520	0.598
12	0.845	0.837	0.847	0.880	**0.887**	0.858	0.872	0.847	0.859	0.870	0.856

Table 4 The results of average classification accuracy of the offered and cutting-edge algorithms (NB classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	**0.823**	0.811	0.808	0.652	0.807	0.681	0.633	0.793	0.802	0.529	0.567
2	0.922	0.917	0.918	**0.929**	0.914	0.918	0.919	0.913	0.925	0.918	0.920
3	**0.694**	0.606	0.664	0.622	0.521	0.653	0.681	0.540	0.595	0.648	0.688
4	0.616	0.621	0.653	0.640	0.629	0.577	0.580	0.633	0.619	0.586	**0.667**
5	0.593	0.511	0.569	0.556	**0.602**	0.526	0.558	0.541	0.560	0.513	0.495
6	**0.810**	0.804	0.519	0.744	0.807	0.553	0.519	0.752	0.519	0.519	0.519
7	0.794	**0.805**	0.782	0.803	0.745	0.590	**0.805**	0.426	0.769	0.426	0.308
8	0.583	0.512	0.505	0.560	0.592	0.510	0.503	0.515	0.590	**0.594**	0.495
9	**0.769**	0.666	0.669	0.761	0.472	0.711	0.667	0.709	0.542	0.617	0.519
10	**0.748**	0.617	0.649	0.680	0.603	0.737	0.566	0.675	0.541	0.615	0.733
11	0.461	0.467	0.450	0.451	0.404	0.420	0.448	0.423	**0.468**	0.427	0.438
12	0.547	0.490	0.507	0.555	0.503	0.557	0.492	0.482	**0.559**	0.436	0.456

Table 5 The results of average classification accuracy of the offered and cutting-edge algorithms (CART classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.867	0.871	0.860	0.846	**0.877**	0.847	0.871	0.870	0.868	0.838	0.867
2	**0.894**	0.867	0.860	0.888	0.869	0.872	0.872	0.865	0.884	0.876	0.872
3	0.708	0.616	**0.720**	0.669	0.567	0.628	0.696	0.582	0.568	0.644	0.665
4	0.648	0.639	0.668	0.671	0.650	0.559	0.620	0.653	0.626	0.571	**0.682**
5	0.613	0.592	0.597	0.601	0.619	0.573	**0.631**	0.593	0.620	0.581	0.582

6	**0.842**	0.837	0.539	0.796	0.834	0.601	0.516	0.782	0.516	0.692	0.515
7	0.776	**0.784**	0.756	0.768	0.767	0.651	0.770	0.426	0.775	0.426	0.426
8	0.655	0.507	0.498	0.583	0.676	0.508	0.500	0.495	0.715	**0.717**	0.499
9	**0.878**	0.706	0.702	0.854	0.636	0.840	0.730	0.733	0.679	0.762	0.649
10	0.766	0.770	0.760	0.770	0.777	0.776	0.772	**0.788**	0.708	0.715	0.775
11	0.657	**0.663**	0.662	0.654	0.608	0.642	0.653	0.650	0.651	0.596	0.660
12	0.890	0.844	0.871	**0.929**	0.926	0.903	0.903	0.863	0.907	0.895	0.883

Table 6 The results of average classification accuracy of the offered and cutting-edge algorithms (SVM classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.848	0.839	0.828	0.835	0.847	**0.852**	0.851	0.833	0.831	0.841	0.833
2	0.923	0.921	0.921	**0.934**	0.915	0.918	0.920	0.917	0.926	0.917	0.925
3	0.727	0.624	**0.737**	0.602	0.623	0.648	0.732	0.606	0.628	0.666	0.697
4	0.666	0.616	**0.684**	0.657	0.664	0.617	0.580	0.650	0.632	0.646	0.677
5	0.686	0.642	0.669	0.642	**0.687**	0.608	0.683	0.658	0.669	0.612	0.631
6	**0.842**	0.835	0.541	0.802	0.836	0.598	0.519	0.776	0.519	0.692	0.519
7	0.829	**0.834**	0.815	0.823	0.812	0.699	0.831	0.426	0.821	0.426	0.426
8	0.595	0.518	0.505	0.569	0.598	0.515	0.489	0.513	**0.602**	0.601	0.496
9	**0.828**	0.600	0.598	0.794	0.603	0.700	0.697	0.714	0.652	0.689	0.646
10	0.793	0.782	0.783	0.782	0.767	0.801	**0.806**	0.797	0.712	0.698	0.794
11	**0.689**	0.667	0.674	0.674	0.587	0.617	0.670	0.650	0.638	0.607	0.651
12	0.568	0.525	0.552	**0.622**	0.612	0.571	0.576	0.558	0.584	0.587	0.589

Table 7 The results of average classification accuracy of the offered and cutting-edge algorithms (RF classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.897	0.890	0.883	0.879	**0.899**	0.874	0.896	0.897	0.886	0.861	0.889
2	**0.917**	0.909	0.908	0.915	0.903	0.910	0.909	0.905	0.914	0.906	0.910
3	**0.741**	0.629	0.726	0.665	0.575	0.688	0.716	0.576	0.578	0.665	0.673
4	0.717	0.698	0.731	**0.740**	0.723	0.613	0.684	0.694	0.679	0.618	0.728
5	0.663	0.619	0.643	0.648	**0.671**	0.600	0.667	0.634	0.657	0.609	0.622
6	**0.847**	0.841	0.554	0.802	0.837	0.606	0.524	0.786	0.524	0.690	0.532
7	0.808	**0.826**	0.802	0.815	0.806	0.696	0.814	0.426	0.810	0.426	0.426
8	0.706	0.515	0.498	0.618	0.738	0.510	0.503	0.497	0.774	**0.780**	0.496
9	0.866	0.720	0.705	**0.877**	0.648	0.846	0.780	0.779	0.719	0.770	0.654
10	0.807	0.812	0.799	0.812	0.813	0.814	0.818	**0.824**	0.714	0.727	0.814
11	**0.699**	**0.699**	0.696	0.686	0.647	0.674	0.691	0.686	0.689	0.627	0.691
12	0.912	0.887	0.901	**0.937**	0.935	0.918	0.924	0.898	0.922	0.916	0.910

Considering the results in more detail in terms of maximum classification accuracy, the experimental results of the offered method against the cutting-edge algorithms are shown in Table 8, Table 9, Table 10, Table 11, and Table 12. Table 8 contains the maximum classification accuracies in terms of the KNN classifier, and this table also demonstrates that the suggested method and SPEC attain the maximum classification accuracy in 3 out of 12 data sets. Table 9 exhibits the maximum classification accuracies in terms of the NB classifier, and it also points out that the offered method and DISR reach the maximum accuracy in 2 out of 12 data sets. Besides, IUFS and USFS have the highest maximum classification accuracy in 5 and 3 out of 12 data sets, respectively. Table 10 shows the maximum classification accuracies of the algorithms using the CART classifier. This table demonstrates that the offered method and LS achieve the highest maximum classification accuracy in 2 out of 12 data sets. In addition, RSR and SRCFS have the highest maximum classification accuracy in 3 out of 12 data sets. Table 11 includes the average classification accuracies obtained by the algorithms using the SVM classifier. This table also indicates that the offered method reaches higher maximum accuracy compared to the others in 4 out of 12 data sets. Finally, Table 12 shows the maximum classification accuracies of the algorithms in terms of the RF classifier. The related table also shows that the proposed algorithm, LS, RNE, and SRCFS obtains higher maximum classification accuracies than the other algorithms in 2 out of 12 data sets. Additionally, IUFS has the highest maximum classification accuracy in 3 out of 12 data sets. Considering all the results in the five tables, the offered algorithm has the highest maximum classification accuracy in 13 out of 60 experiments. IUFS has the highest maximum accuracy in 11 out of 60 experiments. Consequently, the suggested method yields the highest total maximum classification accuracy over all classifiers.

Table 8 The results of maximum classification accuracy of the offered and cutting-edge algorithms (KNN classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.902	0.883	**0.919**	0.858	0.902	0.875	0.894	0.886	0.903	0.872	0.916
2	0.909	0.889	0.910	**0.911**	0.891	0.898	0.892	0.882	0.905	0.885	0.892
3	0.765	0.623	0.706	0.674	0.534	**0.789**	0.729	0.595	0.650	0.677	0.752
4	**0.857**	0.806	0.792	0.826	0.782	0.629	0.830	0.769	0.774	0.600	0.782

5	0.660	**0.675**	0.632	0.645	0.653	0.665	0.652	0.644	0.668	0.636	0.670
6	**0.821**	0.777	0.530	0.743	0.815	0.452	0.369	0.731	0.368	0.745	0.415
7	0.809	0.817	0.815	0.814	0.815	0.749	**0.820**	0.426	0.818	0.426	0.426
8	0.700	0.572	0.510	0.590	0.823	0.551	0.521	0.533	0.865	**0.867**	0.532
9	0.809	0.737	0.766	0.791	0.798	0.766	0.764	0.741	0.787	**0.838**	0.779
10	0.776	**0.809**	0.798	0.758	0.783	0.788	0.806	0.806	0.746	0.754	0.807
11	**0.727**	0.704	0.678	0.639	0.650	0.668	0.673	0.694	0.709	0.640	0.686
12	0.924	0.888	0.896	0.935	0.929	0.928	0.923	0.891	0.934	**0.937**	0.924

Table 9 The results of maximum classification accuracy of the offered and cutting-edge algorithms (NB classifier)

Data set	**Algorithm**										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.845	**0.850**	0.829	0.796	0.838	0.808	0.794	0.829	0.821	0.756	0.758
2	0.939	0.935	0.939	0.949	0.923	0.949	0.947	0.934	0.946	0.943	**0.950**
3	0.766	0.635	0.726	0.645	0.645	0.697	0.708	0.645	0.645	0.706	**0.789**
4	0.655	0.663	0.679	**0.736**	0.681	0.620	0.620	0.690	0.636	0.632	0.694
5	0.610	0.543	0.607	**0.644**	0.626	0.553	0.604	0.573	0.597	0.568	0.559
6	**0.865**	**0.865**	0.520	0.826	0.861	0.606	0.519	0.800	0.519	0.519	0.519
7	**0.848**	0.840	0.847	**0.848**	0.841	0.769	0.842	0.426	0.843	0.426	0.426
8	0.615	0.548	0.515	0.571	**0.619**	0.543	0.511	0.546	0.609	0.616	0.536
9	0.791	0.684	0.686	**0.820**	0.597	0.808	0.707	0.799	0.602	0.752	0.613
10	0.763	0.734	0.720	0.737	0.696	0.767	0.728	0.745	0.667	0.692	**0.784**
11	0.512	0.512	0.511	0.505	0.428	0.468	0.482	**0.526**	0.515	0.462	0.468
12	0.599	0.536	0.591	**0.622**	0.573	0.619	0.574	0.533	0.598	0.502	0.522

Table 10 The results of maximum classification accuracy of the offered and cutting-edge algorithms (CART classifier)

Data set	**Algorithm**										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.914	0.895	0.920	0.898	**0.925**	0.891	0.917	0.917	**0.925**	0.913	0.904
2	0.910	0.906	0.908	0.913	0.890	0.914	0.912	0.906	0.911	0.906	**0.915**
3	**0.808**	0.653	0.800	0.752	0.645	0.739	0.776	0.656	0.645	0.744	0.774
4	0.725	0.703	0.688	0.747	0.681	0.606	0.681	**0.749**	0.666	0.608	0.731
5	0.652	0.642	0.630	0.622	0.654	0.628	0.693	0.623	**0.694**	0.617	0.633
6	**0.894**	0.888	0.626	0.839	0.892	0.637	0.523	0.845	0.521	0.835	0.521
7	0.817	0.818	0.812	0.810	0.818	0.776	**0.820**	0.426	0.818	0.426	0.426
8	0.713	0.542	0.514	0.596	0.758	0.541	0.511	0.519	**0.814**	0.810	0.534
9	0.909	0.738	0.756	0.908	0.684	**0.932**	0.838	0.808	0.807	0.862	0.754
10	0.810	0.806	0.811	0.802	0.820	0.810	0.815	**0.828**	0.805	0.794	0.805
11	0.704	0.710	0.707	0.707	0.706	0.702	0.699	**0.720**	0.717	0.685	0.704
12	0.951	0.896	0.959	0.991	**0.995**	0.972	0.994	0.915	0.966	0.994	0.963

Table 11 The results of maximum classification accuracy of the offered and cutting-edge algorithms (SVM classifier)

Data set	**Algorithm**										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	**0.895**	0.877	0.887	0.890	0.888	0.885	0.890	0.879	0.883	0.884	0.867
2	0.952	0.950	0.949	**0.961**	0.922	0.959	0.944	0.941	0.951	0.950	0.958
3	0.798	0.663	**0.824**	0.694	0.705	0.713	0.811	0.756	0.660	0.805	0.763
4	**0.769**	0.671	0.707	0.762	0.697	0.704	0.683	0.724	0.671	0.704	0.702
5	0.724	**0.729**	0.726	0.706	0.723	0.727	0.723	0.714	0.728	0.724	0.727
6	**0.893**	0.882	0.633	0.836	0.887	0.629	0.519	0.842	0.519	0.830	0.519
7	0.869	**0.874**	0.870	0.869	0.871	0.818	0.873	0.426	0.871	0.426	0.426
8	0.618	0.554	0.515	0.574	0.619	0.547	0.498	0.537	0.619	**0.621**	0.545
9	0.868	0.626	0.598	0.856	0.672	0.845	0.805	**0.885**	0.782	0.793	0.810
10	**0.858**	0.849	0.834	0.822	0.845	0.855	0.857	0.845	0.817	0.813	0.844
11	0.761	**0.794**	0.777	**0.794**	0.774	0.790	0.793	0.765	0.774	0.741	0.793
12	0.640	0.602	0.647	0.728	0.709	0.669	0.673	0.635	0.669	**0.733**	0.715

Table 12 The results of maximum classification accuracy of the offered and cutting-edge algorithms (RF classifier)

Data set	**Algorithm**										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.940	0.922	0.942	0.929	0.945	0.923	**0.947**	0.942	0.941	0.941	0.935
2	0.932	0.925	0.927	**0.936**	0.918	0.928	0.930	0.927	0.931	0.931	0.931
3	**0.863**	0.742	0.798	0.742	0.645	0.798	0.815	0.734	0.645	0.855	0.806
4	0.829	0.791	0.764	**0.837**	0.767	0.709	0.788	0.788	0.767	0.702	0.786
5	0.708	0.682	0.695	0.679	**0.710**	0.683	0.704	0.680	0.703	0.684	0.700
6	**0.904**	0.891	0.663	0.845	0.902	0.659	0.553	0.862	0.548	0.844	0.576
7	0.865	**0.882**	0.875	0.878	0.881	0.820	0.872	0.426	0.875	0.426	0.426
8	0.776	0.567	0.525	0.639	0.844	0.556	0.516	0.533	**0.889**	0.887	0.544
9	0.897	0.770	0.816	**0.943**	0.707	**0.943**	0.874	0.902	0.874	0.879	0.764
10	0.858	0.862	0.864	0.845	0.858	0.855	**0.869**	**0.869**	0.833	0.805	0.855
11	0.746	0.754	0.749	0.752	0.754	0.744	0.750	0.745	**0.762**	0.737	0.754

12	0.971	0.939	0.975	0.995	**0.997**	0.982	0.996	0.951	0.980	0.996	0.983

Figure 8 shows the comparative results of the UFS methods in terms of running time. According to the average running time of the methods, EFS and LS are methods whose running times are under 1 second. In other words, they are the fastest unsupervised feature selection methods in comparison to the other methods. DISR and RNE slowly run on all the high-dimensional data sets. SRCFS and SPEC slowly perform on data sets that have a large amount of data. MCFS, RSR, IUFS, DUFS, and USFS exhibit good performance in terms of running time. Consequently, EFS is the fastest UFS method. LS ranks second. Accordingly, EFS delivers success in terms of both accuracy rate and running time.

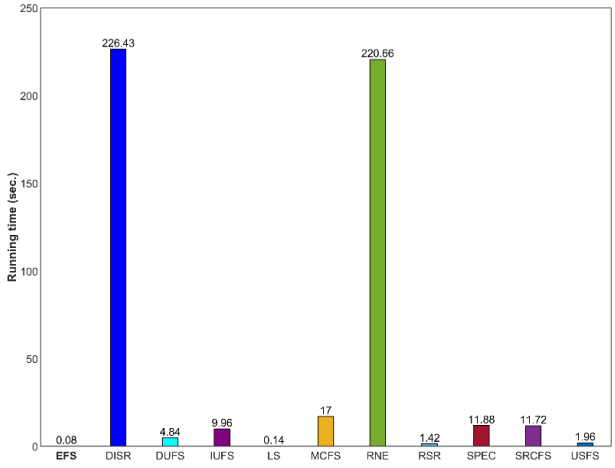

Figure 8 The comparative results of the UFS methods in terms of average running time

5. Conclusion

In this paper, we suggest a new and fast filter-based unsupervised feature selection method called Entropy-based Feature Selection (EFS) based on a single-variable feature selection strategy. The proposed algorithm relies on both the Shannon entropy calculated by the symmetry of the distribution and the cumulative entropy of the distribution.

Unsupervised feature selection algorithms aim to select the most useful features within a dataset. We evaluated the selected features using five well-known classifiers to measure accuracy rates. Among the sixty experiments conducted with features identified by EFS in classification results, the twenty experiments have achieved the highest average accuracy rates. After EFS, the DISR method obtained the highest average accuracy rates on nine datasets.

EFS has an average running time of 0.08 seconds, making it faster than other unsupervised feature selection methods used in the experiments. The LS algorithm follows with an average running time of 0.14 seconds. These low running times demonstrate that the method performs significantly faster on high-dimensional datasets.

Experimental tests on both an artificial dataset and eleven real-world datasets from different domains showed that EFS achieves high accuracy rates. Notably, EFS maintains high average and maximum accuracy rates even as the number of features increases. Future studies can explore EFS's performance in a wider range of data and various application domains. Besides, the next work aims to measure EFS's performance over the clustering problems. Additionally, comparative analyses with other feature selection methods can help better understand the algorithm's competitive advantages. In-depth analyses of the data processed by EFS can provide valuable insights for understanding and improving the algorithm's limitations.

References

[1] S. Solorio-Fernández, J. Ariel Carrasco-Ochoa, J.F. Martínez-Trinidad, A systematic evaluation of filter Unsupervised Feature Selection methods, Expert Syst. Appl. 162 (2020) 113745. https://doi.org/10.1016/j.eswa.2020.113745.

[2] Z.A. Zhao, H. Liu, Spectral Feature Selection for Data Mining, Chapman and Hall/CRC, 2011. https://doi.org/10.1201/b11426.

[3] P. Mitra, S.K. Pal, Pattern Recognition Algorithms for Data Mining, 1st. ed., Chapman & Hall/CRC, 2004.

[4] E. Hancer, B. Xue, M. Zhang, A survey on feature selection approaches for clustering, Artif. Intell. Rev. 53 (2020) 4519–4545. https://doi.org/10.1007/s10462-019-09800-w.

[5] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: NIPS'05 Proc. 18th Int. Conf. Neural Inf. Process. Syst., 2005: pp. 507–514.

[6] Y. Liu, D. Ye, W. Li, H. Wang, Y. Gao, Robust neighborhood embedding for unsupervised feature selection, Knowledge-Based Syst. 193 (2020) 105462. https://doi.org/10.1016/j.knosys.2019.105462.

[7] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '10, ACM Press, New York, New York, USA, 2010: p. 333. https://doi.org/10.1145/1835804.1835848.

[8] P. Zhu, W. Zuo, L. Zhang, Q. Hu, S.C.K. Shiu, Unsupervised feature selection by regularized self-representation, Pattern Recognit. 48 (2015) 438–446. https://doi.org/10.1016/j.patcog.2014.08.006.

[9] Y. Liu, K. Liu, C. Zhang, J. Wang, X. Wang, Unsupervised feature selection via Diversity-induced Self-representation, Neurocomputing. 219 (2017) 350–363. https://doi.org/10.1016/j.neucom.2016.09.043.

[10] S.-L. Huang, L. Zhang, L. Zheng, An information-theoretic approach to unsupervised feature selection for high-dimensional data, in: 2017 IEEE Inf. Theory Work., IEEE, 2017: pp. 434–438. https://doi.org/10.1109/ITW.2017.8277927.

[11] H. Lim, D.-W. Kim, Pairwise dependence-based unsupervised feature selection, Pattern Recognit. 111 (2021) 107663. https://doi.org/10.1016/j.patcog.2020.107663.

[12] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proc. 24th Int. Conf. Mach. Learn. - ICML '07, ACM Press, New York, New York, USA, 2007: pp. 1151–1157. https://doi.org/10.1145/1273496.1273641.

[13] D. Huang, X. Cai, C.-D. Wang, Unsupervised feature selection with multi-subspace randomization and collaboration, Knowledge-Based Syst. 182 (2019) 104856. https://doi.org/10.1016/j.knosys.2019.07.027.

[14] X. Yan, S. Nazmi, B.A. Erol, A. Homaifar, B. Gebru, E. Tunstel, An efficient unsupervised feature selection procedure through feature clustering, Pattern Recognit. Lett. 131 (2020) 277–284. https://doi.org/10.1016/j.patrec.2019.12.022.

[15] F. Wang, L. Zhu, J. Li, H. Chen, H. Zhang, Unsupervised soft-label feature selection, Knowledge-Based Syst. 219 (2021) 106847. https://doi.org/10.1016/j.knosys.2021.106847.

[16] A. Di Crescenzo, M. Longobardi, On cumulative entropies, J. Stat. Plan. Inference. 139 (2009) 4072–4087. https://doi.org/10.1016/j.jspi.2009.05.038.

[17] C.E. Shannon, A Mathematical Theory of Communication, Bell Syst. Tech. J. 27 (1948) 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.