A Distributed Minimum Redundancy Maximum Relevance Feature Selection Approach

M. Sharifnezhad [1], Phd Student; M. Rahmani[2*], Assistant Professor; H. Ghaffarian[3], Assistant Professor

1- Department of Computer Engineering, Faculty of Engineering, Arak University, Arak, Iran.
Email: m-sharifnezhad@araku.ac.ir
2- Department of Computer Engineering, Faculty of Engineering, Arak University, Arak, Iran.
Email: m-rahmani@araku.ac.ir
*corresponding author
3- Department of Computer Engineering, Faculty of Engineering, Arak University, Arak, Iran.
Email: h-ghaffarian@araku.ac.ir

Received: 2020-07-16
Revised: 2020-11-07
Accepted: 2021-02-01

Abstract Feature selection (FS) is served in almost all data mining applications along with some benefits such as reducing the computation and storage cost. Most of the current feature selection algorithms just work in a centralized manner. However, this process does not apply to high dimensional datasets, effectively. In this paper, we propose a distributed version of Minimum Redundancy Maximum Relevance (mRMR) algorithm. The proposed algorithm acts in six steps to solve the problem. It distributes datasets horizontally into subsets, selects and eliminates redundant features, and finally merges the subsets into a single set. We evaluate the performance of the proposed method using different datasets. The results prove that the suggested method can improve classification accuracy and reduce the runtime.

Keywords: Minimum Redundancy, Maximum Relevance, Classification accuracy, Feature reduction, Distributed processing

1. Introduction

In many learning issues, lots of potential features can be involved in identifying an instance. Most learning methods when using a statistical viewpoint for the identification and classification of instances will lose their performance. Data may accompany irrelevant and redundant features [1, 2, 3]. Therefore, selecting a subset of features is considering a minimum number of features that are necessary and sufficient for output identification. Deciding which features must be kept and which ones must be eliminated are done by reliable methods that can predict effectively the relevance rate between features and output class [4, 5].

Researchers generally divide feature selection methods into three categories:

Filter methods: filter approaches are divided into two different groups: univariate approaches and multivariate approaches. In the univariate approaches, features are evaluated independently from other features by considering their relevance degree to the classes. However, approaches based on multivariate evaluation consider features' redundancy together with their relevance to the classes. Features that may have similar ranks are eliminated as redundant cases. Redundancy examination among the features needs more calculation time and improves the classification accuracy [6, 7].

Wrapper methods: in these methods, feature sets are selected utilizing a classifier that evaluates the possible subsets of features. Consequently, the best subset with the highest classification accuracy is selected [8, 9, 10].

Embedded methods: These methods benefit from the advantages of both previous methods employing different evaluation criteria of them. In embedded methods, searching for an optimal feature set is done during the training of the classifier [11, 12]. Usually, the feature selection methods are used in a centralized way [8]. Authors of [13] propose three local search methods called local search, stochastic local search and variable neighborhood search for feature selection in credit scoring of finance and banking. Then, they merge the search methods with a Support Vector Machine (SVM) classifier separately to examine accuracy performance. Authors of [14] propose a two-step method called Filter-Wrapper Hybrid Method (FWHM). In the first step, datasets are scored by six different methods of univariate filters to reduce relevance related to each criterion. In the second step, some random search strategies such as genetic algorithm and particle swarm optimization have been used as a wrapper method. The average ranking yielded from the first step has been considered to generate the preliminary population. Features with higher ranks have more opportunity to be selected. In [15], authors use Support Vector Machine Recursive Feature Elimination (SVM-RFE) to select genes. This method increases classification accuracy in diagnosing cancer genes. Paper

[16] introduces a hierarchical feature selection method in single-layer neural networks to prune redundant and noisy features. Authors of [6] rank features and evaluate them separately using two different filter methods called fisher ratio and Minimum Redundancy Maximum Relevance (MRMR). Then, they select the common selected features between these two methods as the best features. Subsequently, using the SVM-RFE method, the authors examine features other than those obtained previously. Finally, they use a collection of obtained features in the past steps as selected features.

However, in recent years, as of applying huge data and their distribution in different locations, using distributed feature selection is necessary [2, 8, 11, 12, 17]. Massive data cannot be stored in common memories. Therefore, researchers develop many distributed methods instead of centralized ones. These distributed methods need data distributing approaches too. Major techniques for partitioning and distributing data are horizontal or vertical. In horizontal distribution, data are divided into several partitions, which have all features and each of which includes a subset of instances. In vertical distribution, data are divided into partitions with all instances; each of them has a subset of features [17].

After running a given feature selection algorithm on partitions, the selected features in the partitions must be merged and make a single set. So, several merging methods have been proposed. In [1, 2], the authors calculate the classification accuracy of the first selected feature subset. They consider this value as a baseline. Then, the classification accuracy of other subsets of the remaining features is calculated separately. If they improve the baseline veracity, they will also be part of the final selection. In [2, 12, 17], authors determine a threshold using complexity measure. Its logic is based on the fact that the features which are considered as a good candidate cause complexity reduction and must be kept; while those which are bad candidates must be removed. This method is independent of the classifier and causes calculation time reduction for the threshold. Authors in [11, 18] use classification error and percentage of remaining features for calculating the threshold. Both amounts must be minimized as much as possible. The features' amounts are determined per feature that may receive an omission label in each subset. The minimum amount is considered as the threshold. The features which have more omission labels than the threshold are eliminated.

In this paper, we propose a distributed version of the MRMR feature selection approach, called DMRMR. In MRMR, feature selection is performed based on maximum relevance to class and minimum redundancy among the features. The suggested method includes six steps. In the first step, after determining training and test data, training data is distributed horizontally. All subsets have the same number of features. In the second step, each subset of features is scored using MRMR feature selection. During this process, candidate features for elimination receive a vote. A total number of votes for each feature among all subsets are calculated in the third step and the features with votes higher than a threshold would be removed in the fourth step. Finally, all edited subsets merges in a single set in the fifth step and passed to the classifiers for performance evaluation in the final step. We implement and test the performance of the proposed DMRMR and some other centralized and distributed competitor algorithms in Matlab software.

The rest of the article is organized as follows: Section 2 explains the MRMR algorithm. The DMRMR is presented in section 3. Section 4 consists of experimental results and discussion and finally, section 5 presents the conclusion and further studies. Most learning methods when using a statistical viewpoint for the identification and classification of instances will lose their performance. Data may accompany irrelevant and redundant features [1, 2, 3]. Therefore, selecting a subset of features is considering a minimum number of features that are necessary and sufficient for output identification. Deciding which features must be kept and which ones must be eliminated are done by reliable methods that can predict effectively the relevance rate between features and output class [4, 5].

Researchers generally divide feature selection methods into three categories:

Filter methods: filter approaches are divided into two different groups: univariate approaches and multivariate approaches. In the univariate approaches, features are evaluated independently from other features by considering their relevance degree to the classes. However, approaches based on multivariate evaluation consider features' redundancy together with their relevance to the classes. Features that may have similar ranks are eliminated as redundant cases. A redundancy examination among the features needs more calculation time and improves the classification accuracy [6, 7].

Wrapper methods: in these methods, feature sets are selected utilizing a classifier that evaluates the possible subsets of features. Consequently, the best subset with the highest classification accuracy is selected [8, 9, 10].

Embedded methods: These methods benefit from the advantages of both previous methods employing different evaluation criteria. In embedded methods, searching for an optimal feature set is done during the training of the classifier [11, 12].

A feature selection approach is scalable when it performs efficiently in large datasets. Among the different feature selection methods, the filters only rely on general characteristics of the data, and not on the learning machines; therefore, they are faster, and more suitable for large data sets [13]. As [14] shows, the performance of the mRMR method does not decreases when the amount of data increased. The mRMR scales quadratically with the number of features and grows linearly concerning the sample size [15]. Although, mRMR does not include conditional redundancy in its computations and hence it has been criticized for that, [16] shows the mRMR has good performance and this lack is not important in many problems.

To answer the big data-handling problem, in this paper, we propose a distributed version of the mRMR feature selection approach, called DmRMR. In mRMR, feature selection is performed based on maximum relevance to class and minimum redundancy among the features. The suggested method includes six steps. Firstly, after determining training and test data, the training data is distributed horizontally. All subsets have the same number of features. Then, each subset of features is scored using the mRMR feature selection. During this process, candidate features for elimination receive a vote. In the following, a total number of votes for each feature among all subsets are calculated and the features with votes higher than a threshold would be removed. Finally, all edited subsets merges in a single set and passed to the classifiers for performance evaluation. We implement and test the performance of the proposed DmRMR and some other centralized and distributed competitor algorithms in Matlab software. The comparisons show the advantages of the method in terms of accuracy and time complexity.

The rest of the article is organized as follows: a review of the proposed feature selection is presented in section 2. Section 3 explains the mRMR algorithm. The DmRMR is presented in section 4. Section 5 consists of experimental results and discussion and finally, section 6 presents the conclusion and further studies.

2. Background Review

Usually, the feature selection methods are used in a centralized way [8]. Authors of [17] propose three local search methods called local search, stochastic local search, and variable neighborhood search for feature selection in credit scoring of finance and banking. Then, they merge the search methods with a Support Vector Machine (SVM) classifier separately to examine accuracy performance. Authors of [18] propose a two-step method called Filter-Wrapper Hybrid Method (FWHM). In the first step, datasets are scored by six different methods of univariate filters to reduce relevance related to each criterion. In the second step, some random search strategies such as genetic algorithm and particle swarm optimization have been used as a wrapper method. The average ranking yielded from the first step has been considered to generate the preliminary population. Features with higher ranks have more opportunity to be selected. In [19], authors use Support Vector Machine Recursive Feature Elimination (SVM-RFE) to select genes. This method increases classification accuracy in diagnosing cancer genes. Paper [20] introduces a hierarchical feature selection method in single-layer neural networks to prune redundant and noisy features.

Authors of [6] rank features and evaluate them separately using two different filter methods called fisher ratio and mRMR. Then, they select the commonly selected features between these two methods as the best features. Subsequently, using the SVM-RFE method, the authors examine features other than those obtained previously. Finally, they use a collection of obtained features in the past steps as selected features. Paper [21] introduces a combination of mRMR feature selection and machine learning models for the diagnosis of pneumonia. The convolutional neural network is employed as a feature extractor, and some of the existing convolutional neural network models that are AlexNet, VGG-16, and VGG-19 were utilized to realize this specific task. Then, the number of deep features is reduced by using the minimum redundancy maximum relevance algorithm for each deep model.

Authors of [22] propose a classifier subset selection method based on the mRMR method and diversity measures are proposed for building an efficient classifier ensemble. The disagreement and Q-statistic measures are calculated to estimate the diversity among the members. Furthermore, the authors use relevance as a means to determine the accuracy of the ensemble and its members. Paper [23] introduces a computer-based system as a support to gastrointestinal polyp detection. It can detect and classify gastrointestinal polyps from the endoscopic video. Colour wavelet features and convolutional neural network features of endoscopic video frames are extracted. They use mRMR to scale down the feature vector. Instead of using a single classifier, Bootstrap Aggregating an ensemble classifier is used.

In [24], due to the excellent performance of the forward feature selection method for an effective selection of features, the initial subset of this method has been selected by using a combination of high ranking features in different Filter methods.

A hybrid method for multi-label feature selection problems based on combing filter and wrapper methods is proposed in [25], where meta-heuristic algorithms are employed as the wrapper method.

In recent years, as of applying huge data and their distribution in different locations, using distributed feature selection is necessary [2, 8, 11, 12, 26]. Massive data cannot be stored in common memories. Therefore, researchers develop many distributed methods instead of centralized ones. These distributed methods need data distributing approaches too. Major techniques for partitioning and distributing data are horizontal or vertical. In horizontal distribution, data are divided into several partitions, which have all features, and each of which includes a subset of instances. In vertical distribution, data are divided into partitions with all instances; each of them has a subset of features [26].

After running a given feature selection algorithm on partitions, the selected features in the partitions must be merged and make a single set. So, several merging methods have been proposed. In [1, 2], the authors calculate the classification accuracy of the first selected feature subset. They consider this value as a baseline. Then, the classification accuracy of other subsets of the remaining features is calculated separately. If they improve the baseline veracity, they will also be part of the final selection. In [2, 12, 26], authors determine a threshold using complexity measure. Its logic is based on the fact that the features which are considered as a good candidate cause complexity reduction and must be kept; while those which are bad candidates must be removed. This method is independent of the classifier and causes calculation time reduction for the threshold. Authors in [11, 27] use classification error and percentage of remaining features for calculating the threshold. Both amounts must be minimized as much as possible. The features' amounts are determined per feature that may receive an omission label in each subset. The minimum amount is considered as the threshold. The features which have more omission labels than the threshold are eliminated.

3. Minimum Redundancy Maximum Relevance

MRMR method maximizes the relevance between features and the class and minimizes the redundancy among the selected features simultaneously [6]. This

$$D_i = \{(f_i, c_k) | f \in F, c_k \in C\}, F = \{f_1, f_2, \cdots, f_n\}, C = \{c_1, c_2, \cdots, c_m\} \quad (3)$$

method uses mutual information for analyzing relevance and redundancy. Maximum relevance is obtained by calculating the maximum average amount of all mutual information among all current features and the class vector. Minimum redundancy is examined by calculating the minimum average amount of all mutual information among feature vectors. Assume that *MaxD* shows maximum relevance and *MinR* shows minimum redundancy. To reach the optimal subset of features, *MaxD* and *MinR* can be merged using one of the two methods of *Mutual Information Difference* (*MID*) or *Mutual Information Quotient* (*MIQ*) [28, 29] as follows:

$$MID = Max(MaxD - MinR) \quad (1)$$

$$MIQ = Max(MaxD / MinR) \quad (2)$$

Finally, the feature, which has the least value of *MID* or *MIQ*, is selected for elimination in the feature selection process. Figure 1 summarizes how the mRMR works.

$$\begin{pmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{w1} & \cdots & a_{wn} \end{pmatrix} \Rightarrow \left\langle \begin{array}{c} MaxD \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \Leftrightarrow \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} \\ MinR \begin{pmatrix} f_{i1} \\ \vdots \\ f_{in} \end{pmatrix} \Leftrightarrow \begin{pmatrix} f_{j1} \\ \vdots \\ f_{jn} \end{pmatrix} \end{array} \right\rangle \Rightarrow \min\{MID, MIQ\}$$

Fig1. MRMR at a glance

4. The Proposed method

In In datasets with many samples, using centralized methods to examine relevant features and omitting redundant features is time-consuming and not suitable. Also, distribution in many cases causes classification accuracy improvement. To overcome this issue, here, we propose a distributed version of the mRMR multivariate feature selection method. As discussed in section 3, the mRMR is a well-known high-performance centralized feature selection. However, it cannot be omitted in a distributed manner for large-scale datasets.

The proposed method has six steps. The first step includes the horizontal distribution of training datasets in several subsets. In the second step, we use the mRMR feature selection in each feature subset to calculate maximum relevance and minimum redundancy. Then, the features with maximum relevance and minimum redundancy are merged and scored through *MID* and/or *MIQ*. In the third step, in each subset, each feature with a lower score gives an omission label. In the fourth step, voting is done among the features with omission label in each subset. The features with more votes than the threshold are eliminated. In the fifth step, the subsets of the selected features are merged in a final subset. Finally, classification performance is examined on the final subset of selected features. Figure 2 shows the overall layout of the proposed method.

4.1. Details of DMRMR

In this section, we describe the details of DmRMR. During the first step of the proposed method, the training dataset (*D*) with *n* features, *s* samples, and *m* different class labels is divided horizontally into *k* non-empty separate subsets (D_i) without replacement.The value of K is selected randomly. D_i is defined as follows:

where *F* is a non-empty and limited set of *n* features and *C* is a non-empty and limited set of *m* different classes.

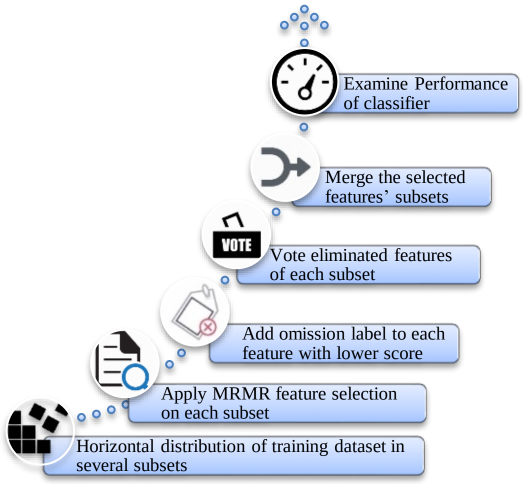
Examine Performance of classifier
Merge the selected features' subsets
VOTE
Vote eliminated features of each subset
Add omission label to each feature with lower score
Apply MRMR feature selection on each subset
Horizontal distribution of training dataset in several subsets

Fig2. The overall layout of the proposed method

Using Horizontal distribution, each subset has a full set of features of the original dataset. During the distribution, all subsets have similar sizes.

In each subset D_i, to calculate the relevance of the feature f_i to class set *C* using the mRMR feature selection method, we use the following equation:

$$MaxD(F_{D_i}, C) = \frac{1}{|F_{D_i}|} \sum_{f_i \in F_{D_i}} I(f_i, C) \quad (4)$$

where *I*(*X*,*Y*) is the mutual information between two *X* and *Y*. More dependency between *X* and *Y* causes a higher amount of mutual information between them. Generally, the mutual information between two variables *X* and *Y*, with probability density functions *p*(*x*), *p*(*y*), and *p*(*x*, *y*) is obtained as follows:

$$I(x, y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log\left(\frac{p(x, y)}{p(x)\, p(y)}\right) \quad (5)$$

Feature f_i would be selected if $I(f_i, C)$ has the highest value within the class *C*. This procedure shows the maximum relevance of feature f_i to class *C*. According to Equation (4), all values of mutual information between the separate feature f_i and class *C* will be obtained. Following the approaches in [6-7, 30-32], we select features with the highest relevance to class.

To minimize redundancy in the mRMR method, we can obtain mutual information between features as follows:

$$MinR(F_{D_i}) = \frac{1}{|F_{D_i}|^2} \sum_{f_i, f_j \in F_{D_i}} I(f_i, f_j) \quad (6)$$

To reach optimal features through maximum relevance and minimum redundancy, we have:

$$\lambda = \arg \quad \min\{MID, MIQ\} \quad (7)$$

$$F_{D_i} = F_{D_i} - \{\lambda\} \quad (8)$$

$$LCR_{D_i} = LCR_{D_i} + \{\lambda\} \quad (9)$$

The features with the lowest ranks are candidates for elimination from the final feature set. They are removed from the feature set of the subset and added to the list of candidates for removing of the subset.

After finishing the distributed candidate selection process in all subsets, in the next step, the candidate features for removing in all subsets are voted, one vote per candidate. The features with votes more than a specific threshold would be removed permanently from all subsets. To find this threshold, we follow the strategy of [33]. This strategy is based on complexity measures. To do this, we first define two new parameters as follows:

$$\varphi = \mu - \frac{\sigma}{2} \qquad (10)$$

$$\Phi = \mu + \frac{\sigma}{2} \qquad (11)$$

Here, μ is the average of votes and σ standard deviation of votes. Then we run a loop with a counter between φ and Φ. In the *i*th round of the loop, we select a subset F_c of all features with the number of votes less than the counter value. Then, we compute the inverse of the Fisher ratio *f* using only features in the F_c subset over the training dataset, based in Equation 2 in Ref. [33]. After that, we compute the following cost function for the F_c in the *i*th round:

$$\vartheta[i] = \alpha f + (1-\alpha)\frac{|F_c|}{m} \qquad (12)$$

where α is a weight factor equals to 0.75 as suggested in [33], giving more influence to to the classification error and |Fc| is the number of features in the subset Fc.

After the end of the loop, the minimum value in ϑ is selected as threshold *Tr*. Those features with votes more than the threshold *Tr* would be eliminated permanently. The pruned subsets are collected in a final set using a merge function. Finally, classification performance is examined using the final train and test datasets. Algorithm 1 shows the pseudo-code of the DmRMR algorithm.

Algorithm 1 Pseudo-code of DMRMR

```
Input:
  - D(m×s): Dataset with m samples and s features
  - D(m1×s): Train Dataset
  - D(m2×s): Test Dataset
  - V: Vector of votes for Removing Features,    Initialize the V to 0
Output:
  - FS: Subset of Final Selected Features
Compute:
  1. Horizontal partition the train data into k non-empty and limited subsets Di
  2. For each Di subset
      a. $\mathrm{MaxD}(F, C) = \frac{1}{|F|}\sum_{fi\epsilon F}^{n} \mathrm{I}(f_i, C)$
      b. MinR(F) = $\frac{1}{|F|^2}\sum_{f_i f_j \epsilon F}^{n} I(f_i, f_j)$
       c. MID: max(MaxD(F,C) – MinR(F))
       d. MIQ: max(MaxD(F,c) / MinR(F))
       d. λ = arg min {MID or MIQ}
       e. F = F – {λ}
        f. LCR = LCR + {λ}
3. Calculate total votes for each feature among all subsets
4. Remove features with votes more than Tr threshold permanently
5. Merge all edited subsets
6. Obtaining accuracy usin
```

5. Performance Evaluation

In this section, first, we explain the experimental setup. Then, we evaluate the results of the suggested method in terms of classification accuracy, time complexity, and the number of features. All experiments have been run on a PC with Intel(R) Core i7-2670QM CPU with 2.2 GHz frequency and 8 GB RAM using MATLAB R2013 software on Windows 7 platform. Also, we have used six datasets to evaluate the suggested method (see table 1). The instances in each dataset are divided into two training and test sets. Normally, for each dataset, this division is considered as $\frac{2}{3}$ training dataset and $\frac{1}{3}$ test dataset.

Table 1. Characteristics of used Dataset [25]

Dataset	# Samples		# Features	# Classes
	Training	Test		
Ozone	1691	845	72	2
Madelon	1600	800	500	2
Spambase	3067	1534	57	2
Connect4	45038	22519	42	3
Spect	178	89	22	2
Isolet	5198	2599	617	26

We compare the DmRMR with Information Gain, ReliefF, and SVM-RFE feature selection strategies.

Information Gain: The IG is one of the most common univariate methods based on filters for evaluating the features. This method scores and ranks the relevance of each feature with a class based on Information Gain. Then, it selects a determined number of features with higher ranks using a threshold [35].

ReliefF: This method is developed based on the Relief algorithm. It is capable to deal with noisy, incomplete, and multi-class data [36]. Instead of finding n instances of nearest hit and nearest miss, the ReliefF selects n instances from each class. The share of each non-classmate in weighting is calculated based on its previous probabilities. Noise in a class or the amount of a feature affects selecting the nearest hit and the nearest miss significantly. To select the nearest hit and the nearest miss more carefully, the ReliefF algorithm uses n nearest hit and the nearest miss. The average share of each one is considered in estimating the quality of each feature [37].

SVM-RFE: The SVM-RFE has been introduced by Guyon et al. [38] as a feature selection method. In this method, the features are eliminated recursively. In each step, the features' weight is calculated by a support vector. Each of them is given one score. The feature with the minimum score will be eliminated [28]. This process will be repeated until all features are eliminated. In the end, we have a ranked list of features. Feature selection can be obtained by selecting a group of superior features.

5.1. Classification Accuracy

In this section, we compare the obtained classification accuracy by K-Nearest Neighbor (KNN), Naïve Bayes (NB), and SVM, on the obtained results of DmRMR, IG, ReliefF, and SVM-RFE. These classifiers are commonly used for evaluating many feature selection algorithms, e.g. in [1, 2, 8, 11, 12, 19, 26, 34].

SVM classifiers were originally designed for solving a binary classification problem. There are some methods for solving a multi-class problem such as One-against-One (OaO), One-against-All (OaA) [39,40], or Directed Acyclic Graph (DAG) [41] and approaches based on building a binary decision tree [42,43].

Table 2. Accuracy results

	Dataset		IG	ReliefF	SVM-RFE	MRMR	Without FS
KNN	Ozone	C	98.75	93.70	98.58	98.58	98.35
		D	77.71	80.18	98.34	98.58	N/A
	Madelon	C	49.37	48.25	49.50	**50.15**	48.25
		D	**88.50**	88.38	50.75	52.12	N/A
	Spambase	C	**77.62**	75.60	72.73	74.49	77.10
		D	77.71	**80.18**	75.53	76.02	N/A
	Spect	C	73.03	71.91	65.17	**74.15**	73.03
		D	73.03	76.40	71.91	**76.50**	N/A
	Isolet	C	73.64	90.62	88.26	**90.06**	85.23
		D	61.83	57.01	89.14	**91.35**	N/A
	Connect4	C	50.53	37.26	**67.54**	66.35	62.06
		D	54.52	57.01	69.51	**70.12**	N/A
	Average	C	70.49	69.56	73.63	**75.63**	74.00
		D	72.22	73.19	75.86	**77.45**	N/A
SVM	Ozone	C	**98.70**	98.22	**98.70**	**98.70**	98.12
		D	**98.70**	**98.70**	90.18	98.70	N/A
	Madelon	C	**49.62**	**49.62**	**49.62**	**49.62**	48.62
		D	66.75	**67.25**	49.62	49.62	N/A
	Spambase	C	**95.56**	77.69	94.65	92.37	91.65
		D	83.38	83.77	91.86	**96.41**	N/A
	Spect	C	**68.54**	66.29	64.04	65.17	62.92
		D	64.04	64.04	**70.79**	65.17	N/A
	Isolet	C	92.90	**95.17**	94.25	91.02	90.42
		D	80.12	81.98	**95.30**	93.35	N/A
	Connect4	C	66.42	60.42	71.22	**72.26**	65.58
		D	60.42	60.42	**74.72**	73.12	N/A
	Average	C	78.62	74.57	**78.75**	78.19	76.22
		D	75.57	76.03	78.75	**79.40**	N/A
NB	Ozone	C	70.41	71.84	71.24	**73.67**	71.95
		D	78.46	66.86	**79.53**	73.50	N/A
	Madelon	C	47.25	46.87	**48.85**	46.50	46.87
		D	70.50	**72.25**	49.50	47.0	N/A
	Spambase	C	76.53	**92.85**	80.12	78.88	93.20
		D	66.95	92.05	89.68	**93.22**	N/A
	Spect	C	73.03	**76.50**	71.91	74.15	74.16
		D	74.16	**75.28**	74.16	71.91	N/A
	Isolet	C	79.98	73.10	80.46	**81.25**	80.34
		D	66.77	53.69	**82.42**	**82.42**	N/A
	Connect4	C	**60.58**	60.30	60.42	60.10	58.62
		D	60.20	**60.50**	60.42	60.42	N/A
	Average	C	67.96	70.24	68.83	69.09	**70.86**
		D	69.51	70.11	**72.62**	71.41	N/A

In this paper, we use an SVM classifier, utilizing the hierarchy binary decision tree for solving multiclass problems.

We present the results of two different implementation strategies for each feature selection method: the distributed implementation, shown by prefix D at the beginning of the method names in the tables, and the centralized implementation, shown by prefix C at the beginning of the method names in the tables. For performance evaluation, we have implemented the proposed distributed versions of IG and ReliefF based on the proposed strategy in [2] and distributed version of SVM-RFE based on the proposed method in [44].

Table 2 shows the results of classification accuracy. The obtained results are variable because of the relevance to datasets and classifier kind. The best-reported results for each dataset have been shown as highlighted. In the last column of the table, the results of classification accuracy have been calculated without using feature selection algorithms. Also, the last row of the table includes the average results. It can be seen that among the results of classification accuracy, with or without using centralized or distributed feature selection algorithms, in half of the datasets, the best accuracy with the KNN is related to the DmRMR method. In the average row, still, we can see that DmRMR achieves the best results. It is interesting that in this row, in all cases, the distributed feature selection algorithms act better than their centralized versions.

In the SVM classifier, although DmRMR could not achieve the best results in most cases, it has the best performance in the average row. Also, we can see that, on average, the SVM classifier receives the best accuracy results. However, we can see the worst results in the NB. Even in the centralized form, if we do not use feature selection, the NB acts more accurately than using feature selection strategies. On average, distributed SVM-RFE followed by DmRMR has the best performance results using the NB.

Figure 3 shows the differences between the achieved results in different classifiers. As this figure shows, the DmRMR achieves the best results in mean accuracy compared with the other feature selection methods in a distributed state with KNN and SVM. Also, mostly, mean classification accuracy in the different classifiers in a distributed state has improved comparing a centralized state in all tested feature selection methods. Only, SVM-RFE with IG is the exception in this case.

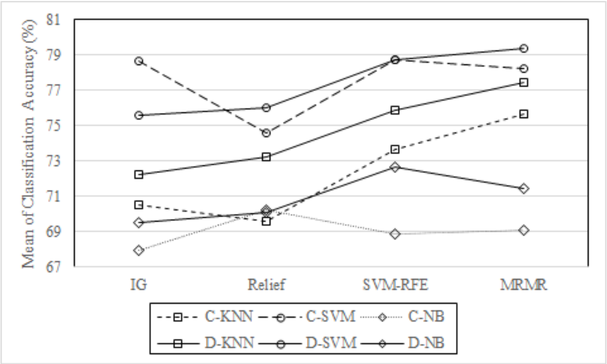

Fig3. Mean accuracy of different classifiers

The results of the study were compared with the related studies as shown in Table3. Different methods have been implemented using the same dataset. The average of Classification Accuracy results have been compared for every classifer. As can be seen in this table, our proposed algorithm reaches the best results. In this table, Ref. [11] does not prepare the accuracy results of IG and ReliefF methods with the KNN classifier. So, their values are missing in the table.

5.2. *Runtime Result*

Table 4 shows the runtime of feature selection algorithms in two distributed state and centralized state. The time showed in this table is the maximum time needed for the feature selection algorithm on each subset that has been made in the division stage; because in a distributed approach, all datasets can be processed at the same time.

Table 3: Comparison Results with Related works

Algorithm	Classifier	Classification Accuracy
IG[2]	KNN,NB,SVM	75.47,68.58, 77.87
IG[11]	KNN ,NB,SVM	- , 73.15, 77.86
IG[12]	KNN,NB,SVM	67.5,70.5,75.5
IG[44]	KNN,NB,SVM	73.25, 56.82,73.13
ReliefF [2]	KNN,NB,SVM	75.57,69.03,78.42
ReliefF [11]	NB,SVM	68.25,78.51
ReliefF [12]	KNN,NB,SVM	74.0,70.0,75.0
ReliefF [44]	KNN,NB,SVM	70.99, 56.98,75.17
SVM-RFE [44]	KNN,NB,SVM	74.03,54.22,78.30
Proposed method	KNN,NB,SVM	**77.64, 71.31,82.24**

In this experiment, all the subsets were processed in a machine; however, the suggested algorithm can be run on several processors.

As can be seen in Table 3, the required runtime on all datasets using distributed methods rather than centralized ones has decreased. The minimum runtime in the distributed state is related to the Spect dataset with the ReliefF feature selection method. This dataset has the minimum number of features and instances comparing the others. Among the other datasets, the minimum distributed runtime is related to SVM-RFE feature selection. Also, the average runtime in both distributed and centralized states is reported in the table. According to this, the maximum difference between centralized and distributed runtime is related to the suggested method, decrease from 1826.217 seconds in centralized mode to 61.762 seconds in distributed mode. The results of this table show that mRMR is the most time-consuming feature selection strategy. On the other hand, SVM-RFE is the fastest method. For more clarification, we divide runtime in different cases into runtimes of SVM-RFE (in both centralized and distributed states). The result is interesting. Distribution in IG is not effective. Its effect on ReliefF is low. In contrast, the structure of mRMR has good potentials for parallelization.

5.3. Number of Features

Table 5 reports the number of selected features in two states of distributed (D) and centralized (C) using four different feature selection methods. In all states, the threshold 25 percent has been used for eliminating the features.

As you can see, in most datasets, the number of selected features by centralized approaches is more than the distributed state. In cases that classification accuracy in a centralized state is more than a distributed one, it can be concluded that no significant decrease occurred for classification accuracy through data distribution.

Table 4. Runtime results (in second)

FS Method		Ozone	Madelon	Spect	Spambase	Connect4	Isolet	Average	% of Improvement
mRMR	C	40.450	3631.541	0.646	18.917	1565.512	5700.237	1826.217	256.64
	D	31.561	98.21	0.288	13.052	62.820	164.64	61.762	102.94
SVM-RFE	C	4.728	6.960	0.680	8.962	12.250	9.146	7.121	1
	D	0.245	0.712	0.224	0.645	0.828	0.947	0.600	1
IG	C	2.116	22.008	0.316	3.077	7.847	30.742	11.018	1.55
	D	1.429	16.037	0.096	2.737	2.183	20.05	7.089	11.82
ReliefF	C	10.284	61.802	0.711	15.920	1502.104	603.497	365.720	51.36
	D	0.805	15.310	0.024	2.712	8.195	123.17	25.036	41.37

Table 5. Number of selected features

FS Method		Ozone	Madelon	Spect	Spambase	Connect4	Isolet
Full Set		72	500	22	57	42	617
mRMR	C	55	185	21	42	30	226
	D	31	52	16	29	32	124
SVM-RFE	C	41	175	16	42	30	226
	D	16	40	7	30	33	102
IG	C	41	175	16	42	30	226
	D	37	155	10	37	30	168
ReliefF	C	40	174	16	42	30	226
	D	17	68	10	21	34	175

The maximum decrease in the number of features in the distributed state is related to Madelon and Isolet datasets in which not only their classification accuracies have not decreased, but also they improved. Because the removal of noise data improves the final classification result.

6. Conclusion

In this article, we present a distributed version of the mRMR feature selection algorithm, called DmRMR. The proposed DmRMR algorithm distributes datasets horizontally, does a feature selection process on each subset, and merges the results in a subset. We evaluate the suggested method using six datasets in terms of classification accuracy, time complexity, and number of features. The results show that in most tested datasets, classification accuracy has improved. The runtime of the suggested feature selection process in the distributed state has decreased compared with the centralized state. Also, the reduction of feature number did not cause the reduction of classification accuracy and in most cases, it has improved comparing centralized state.

7. References

[1] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A Distributed Wrapper Approach for Feature Selection", In Computational Intelligence and Machine Learning Conference(ESANN), April 2013, Bruges, Belgium pp. 24-26.

[2] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A Distributed Feature Selection Approach Based on a Complexity Measure", Advances in Computational Intelligence, pp. 15-28, 2015.

[3] G. Chandrashekar and F. Sahin, "A Survey on Feature Selection Methods", Journal of Computers and Electrical Engineering, vol. 40, pp.16–28, 2014.

[4] I. Guyon, A. Elisseeff, "An Introduction to Variable and Feature Selection", Journal of Machine Learning Research, vol.3, pp.1157–1182, 2003.

[5] I.Guyon, S.Gunn, M.Nikravesh and L.A.Zadeh, "Feature Extraction: Foundations and Applications", Springer, vol. 207, 2006.

[6] L. Yu, H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy", Journal of Machine Learning Research, vol. 5, pp. 1205–1224, 2004.

[7] C. Ding, H. Peng, "Minimum Redundancy Feature Selection From Microarray Gene Expression Data", Journal of Bioinformatics and Computational Biology, vol. 3, no. 2, pp.185–205, 2005.

[8] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Distributed Feature Selection: An Application to Microarray Data Classification", Applied Soft Computing, vol. 30, pp. 136-150, 2015.

[9] R. Kohavi, GH. John, "Wrappers for Feature Subset Selection", Artificial Intelligence, vol. 97, pp. 273–324, 1997.

[10] J.Li, K.Cheng, S.Wang, F. Morstatter, and R. P. Trevino, "Feature Selection: A Data Perspective", Journal of ACM Computing Surveys, vol. 50, no 6, 2018.

[11] V. Bolón-Canedo, N. Sánchez-Maroño, and J. Cerviño-Rabuñal, "Scaling up Feature Selection: a Distributed Filter Approach", Advances in Artificial Intelligence, pp. 121-130, 2013.

[12] L. Mor´an-Fern´andez, V. Bol´on-Canedo, and A. Alonso-Betanzos, "A Time Efficient Approach for Distributed Feature Selection Partitioning by Features", Lecture Notes in Computer Science book series (LNCS), vol. 9422, pp.245–254, 2015.

[13] Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (Eds.), "Feature extraction: foundations and applications", Springer, Vol. 207, 2008.

[14] Rego-Fernández, D., Bolón-Canedo, V., & Alonso-Betanzos, A., "Scalability Analysis of mRMR for Microarray Data", In ICAART (1), pp. 380-386,2014.

[15] Ramírez‐Gallego, S., Lastra, I., Martínez‐Rego, D., Bolón‐Canedo, V., Benítez, J. M., Herrera, F., & Alonso‐Betanzos, A., "Fast‐mRMR: Fast minimum redundancy maximum relevance algorithm for high‐dimensional big data", International Journal of Intelligent Systems, vol.32(2), pp.134-152,2017.

[16] Brown, G., Pocock, A., Zhao, M. J., & Luján, M.,"Conditional likelihood maximisation: a unifying framework for information theoretic feature selection", The journal of machine learning research, vol.13(1),pp. 27-66, 2012.

[17] D. Boughaci and A.A Alkhawaldeh (2018), "Three Local Search-Based Methods for Feature Selection in Credit Scoring", Vietnam Journal of Computer Science, vol. 5, no 2, pp. 107–121, 2018.

[18] H. Min and W. Fangfang "Filter-Wrapper Hybrid Method on Feature Selection", In Second WRI Global Congress on Intelligent Systems (GCIS), Dec 2010, Wuhan, Chinapp, pp.98-101.

[19] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification using Support Vector Machines", Machine Learning, vol. 46, no 1-3, pp. 389–422, 2002.

[20] Q. Wang, J. Wan, F. Nie, B. Liu, C.Yan, and X. Li, "Hierarchical Feature Selection for Random Projection", IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 5, pp. 1581–1586, 2019.

[21] Toğaçar, M., Ergen, B., Cömert, Z., & Özyurt, F., "A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models", IRBM, 41(4), pp.212-222, 2020.

[22]Cheriguene, S., Azizi, N., Dey, N., Ashour, A. S., & Ziani, A., "A new hybrid classifier selection model based on mRMR method and diversity measures", International Journal of Machine Learning and Cybernetics, 10(5), pp.1189-1204, 2019.

[23]Billah, M., & Waheed, S.," Minimum redundancy maximum relevance (mRMR) based feature selection from endoscopic images for automatic gastrointestinal polyp detection", Multimedia Tools and Applications, pp.1-11,2020.

[24] F. Alighardashi, M. A. Zare Chahooki, "The Effectiveness of the Combination of Filter and Wrapper Feature Selection Methods to Improve Software Fault Prediction", Tabriz Journal of Electrical Eng., vol. 47, no. 1, 2017.

[25]S. Kashef, H. Nezamabadi-pour, "A Hybrid Method to Find Effective Subset of Features in Multi-label Datasets", Tabriz Journal of Electrical Engineering, vol. 48, no. 3, 2018

[26] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos, "Centralized vs. Distributed Feature Selection Methods Based on Data Complexity Measures", Journal of Knowledge-Based Systems, vol. 117, pp.27–45, 2016.

[27] A.De Haro Garc´ıa, “Scaling Data Mining Algorithms. Application to Instance and Feature Selection”, Ph.D. Thesis, University of Granada, 2011.

[28] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene Selection for Cancer Classification using Support Vector Machines”, Journal of Machine Learning, vol.46, pp.389–422, 2002.

[29] H. Peng, F. Long, and C. Ding, “Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min Redundancy”, IEEE Transaction on Pattern Analysis and Machine Intelligence., vol. 27, no. 8, pp. 1226–1238, 2005.

[30] Y.Lu, W.Liu, and Y.Li, “A Feature Selection Based on Relevance and Redundancy”, JCP, vol. 10, no. 4, pp. 284-291, 2015.

[31] PH. Taylor et al., “Redundant Feature Selection for Telemetry Data”, ADMI, vol. 8316, pp.53- 65, 2013.

[32] M. Radovic et.al, “Minimum Redundancy Maximum Relevance Feature Selection Approach for Temporal Gene Expression Data”, BMC Bioinformatics, vol. 18, no. 9, 2017.

[33]. V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “A Distributed Feature Selection Approach Based on a Complexity Measure”, In 13th International Work-Conference on Artificial Neural Network, Palma de Mallorca, Spain, pp 15-28, 2015.

[34] http://archive.ics.uci.edu/ml/datasets/

[35] M.A. Hall, L.A. Smith, “Practical Feature Subset Selection for Machine Learning”, Computer Science, vol. 98, pp.181–191, 1998.

[36] I. Kononenko, “Estimating Attributes: Analysis and Extensions of RELIEF”, Machine Learning: ECML-94, vol. 784, pp 171-182, 1994.

[37] M. Robnik-Šikonja and I. Kononenko, “Theoretical and Empirical Analysis of ReliefF and RReliefF”, Machine learning, vol. 53, no. 1-2, pp. 23-69, 2003.

[38] H. Djellali, N. Ghoualmi Zine and N. Azizi, “Two Stages Feature Selection Based on Filter Ranking Methods and SVM-RFE on Medical Applications”, Modelling and Implementation of Complex Systems, pp. 281-293, 2016.

[39] Y. Liu and Y.F.Zheng, “One-against-all multi-class SVM classification using reliability measures”, IEEE international joint conference on neural network (IJCNN), vol. 2, pp. 849-854, 2005.

[40] M. Arun Kumar, M. Gupta, “Fast multiclass SVM classification using decision tree based one-against-all method”, Springer, neural process letter, vol. 32, pp. 311-323, 2010.

[41] J. C. Platt, N. Cristianini and J. Shahere-Taylo, “Large margin DAGs for multiclass classification”, Advances in neural information processing system, vol. 12, no. 3, pp. 547-553, 2000.

[42] G. madzarov, D. gjorgjevikj, and I. chorbev, “A multi-class SVM classifier utilizing binary decision tree”, An international journal of computing and informatics, Informatica, vol. 33 number 2, ISSN0350-5596, Slovenia; pp.233-241, 2009.

[43] A. Meshram, R. Gupta and S. Sharma, “Advanced Probabilistic Binary Decision Tree Using SVM for large class problem”, International Journal of Computer Science and Information Technologies(IJCSIT), Vol. 6 (2), pp. 1660-1664,2015.

[44] E. Afshari, “Proposing a New Embedded Method for Feature Reduction in Big Data”, Master of Science Thesis, Arak University, 2017.