Expert Systems with Applications 41 (2014) 6371–6385

Contents lists available at ScienceDirect

Expert Systems with Applications

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

MIFS-ND: A mutual information-based feature selection method

N. Hoque [a,*], D.K. Bhattacharyya [a,*], J.K. Kalita [b,*]

[a] Department of Computer Science & Engineering, Tezpur University, Napaam, Tezpur 784028, Assam, India
[b] Department of Computer Science, University of Colorado at Colorado Springs, CO 80933-7150, USA

ARTICLE INFO

Keywords:
Features
Mutual information
Relevance
Classification

ABSTRACT

Feature selection is used to choose a subset of relevant features for effective classification of data. In high dimensional data classification, the performance of a classifier often depends on the feature subset used for classification. In this paper, we introduce a greedy feature selection method using mutual information. This method combines both feature–feature mutual information and feature–class mutual information to find an optimal subset of features to minimize redundancy and to maximize relevance among features. The effectiveness of the selected feature subset is evaluated using multiple classifiers on multiple datasets. The performance of our method both in terms of classification accuracy and execution time performance, has been found significantly high for twelve real-life datasets of varied dimensionality and number of instances when compared with several competing feature selection techniques.

1. Introduction

Feature selection, also known as variable, attribute, or variable subset selection is used in machine learning or statistics for selection of a subset of features to construct models for describing data (Arauzo-Azofra, Aznarte, & Benítez, 2011; Cadenas, Garrido, & MartíNez, 2013; Dash & Liu, 1997; Liu & Yu, 2005; Polat & Güneş, 2009). Two important aspects of feature selection are: (i) minimum redundancy and (ii) maximum relevance (Unler, Murat, & Chinnam, 2011). Besides these, people use feature selection for dimensionality reduction and data minimization for learning, improving predictive accuracy, and increasing comprehensibility of models. To satisfy these requirements, two dimensionality reduction approaches are used, i.e., feature extraction and feature selection (Lewis, 1992). A feature selection method selects a subset of relevant features from the original feature set, whereas a feature extraction method creates new features based on combinations or transformations of the original feature set. Feature selection is used to overcome the curse of dimensionality (Hughes, 1968) in a pattern recognition system. The main objective of feature selection is to identify m most informative features out of the d original features, where $m < d$.

In the literature, feature selection approaches are widely used prior to or during classification of data in pattern recognition and data mining in fields as diverse as bioinformatics and network security. Feature selection approaches are classified into four categories, such as filter approach, wrapper approach, embedded approach, and hybrid approach.

1. Filter approach (Guyon & Elisseeff, 2003): This approach selects a subset of features without using a learning algorithm. It is used in many datasets where the number of features is high. Filter-based feature selection methods are faster than wrapper-based methods.
2. Wrapper approach (Blum & Langley, 1997): This approach uses a learning algorithm to evaluate the accuracy produced by the use of the selected features in classification. Wrapper methods can give high classification accuracy for particular classifiers, but generally they have high computational complexity.
3. Embedded approach (Guyon & Elisseeff, 2003): This approach performs feature selection during the process of training and is specific to the applied learning algorithms.
4. Hybrid approach (Hsu, Hsieh, & Lu, 2011): This approach is a combination of both filter and wrapper-based methods. The filter approach selects a candidate feature set from the original feature set and the candidate feature set is refined by the wrapper approach. It exploits the advantages of these two approaches.

Feature selection plays an important role in network anomaly detection. In a network anomaly detection system, anomalies are identified in a network by monitoring the behavior of normal data compared to abnormal ones. The detection system identifies an attack based on behavioral analysis of features of network traffic

* Corresponding authors.
E-mail addresses: tonazrul@gmail.com (N. Hoque), dkb@tezu.ernet.in (D.K. Bhattacharyya), jkalita@uccs.edu (J.K. Kalita).

http://dx.doi.org/10.1016/j.eswa.2014.04.019
0957-4174/

data. Network traffic data objects may contain protocol specific header fields, such as source address, source port, destination address, destination port, protocol type, flags and time to live. Not all this information is equally important to detect an attack. In addition, if the number of features is high, the classifier usually performs poorly. So, the selection of an optimum and relevant set of features is important for the classifier to provide high detection accuracy with low computational cost.

1.1. Applications of feature selection

Feature selection is an important step in most classification problems to select an optimal subset of features to increase the classification accuracy and improve time needed. It is widely used in many applications of data mining and machine learning, network anomaly detection and natural language processing.

1.1.1. In data mining and machine learning

Machine learning is appropriate when a task is defined by a series of cases or examples rather than in terms of an algorithm or rules. Machine learning is useful in many fields including robotics, pattern recognition and bioinformatics. In learning a classifier, a predictor or a learning algorithm is used to extract information from the behavior of the features of a data object. From the acquired information, the classifier can predict the class label of a new data object. So, a feature selection algorithm is used to find an optimal features set that can be used by the predictor to generate maximal information about the class label of the data object.

1.1.2. In network anomaly detection

Anomaly detection in real-time is a challenging problem for network security researchers and practitioners. A network packet contains a large number of features and hence, an anomaly detection system takes a significant amount of time to process features to detect anomaly packets (Khor, Ting, & Amnuaisuk, 2009). To overcome the problem we need a feature selection method that can identify the most relevant features from a network packet. The selected features are used by an intrusion detection system to classify network packets either as normal or anomalous. The accuracy, detection time and effectiveness of the detection system depend on the input feature set along with other hardware constraints. Therefore, feature selection methods are used to determine a minimal feature set which is optimal and does not contain redundant features.

1.1.3. In text categorization

In the recent past, content-based document management tasks have become important. Text categorization assigns a boolean value to a document d and the value determines whether the document belongs to a category c or not. In document classification, appearance of any word in a document may be considered a feature. Feature sets used reflect certain properties of the words and their context within the actual texts. A feature set with most relevant features can predict the category of the document quickly. In automatic text categorization, the native feature space contains unique terms (words or phases) of the document which can number in tens, hundreds or thousands. As a result, operation cost for text categorization is not only time consuming but also intractable in many cases (Yang & Pedersen, 1997). Therefore, it is highly desirable to reduce the feature space for effective classification of a document.

1.1.4. In gene expression data mining

Gene analysis is a topic of great interest associated with specific diagnosis in microarray study. Microarrays allow monitoring of gene expression for thousands of genes in parallel and produce enormous valuable data. Due to large dimension and over-fitting problem (Horng et al., 2009) of gene expression data, it is very difficult to obtain a satisfactory classification result by machine learning techniques. Also, discriminant analysis is used in gene expression data analysis to find effective genes responsible for a particular disease. These facts have given rise to the importance of feature selection techniques in gene expression data mining (Saeys, Inza, & Larrañaga, 2007). Gene expression data is typically high dimensional and is error prone. As a result, feature selection techniques applied in gene expression data analysis especially classification techniques. Feature selection has also been applied as a preprocessing task in clustering gene expression data in search of co-expressed patterns.

1.2. Contribution

The main contribution of this paper is a mutual information-based feature subset selection method to use for complex time series data classification such as network anomaly detection, pattern classification. The method has been established to perform satisfactorily both in terms of classification accuracy and execution time for a large number of real life benchmark datasets. The effectiveness of the method is established in terms of classification accuracy exhibited for several real-life intrusion, text categorization, gene expression and UCI datasets, in association with some well-known classifiers.

The rest of the paper is organized as follows. In Section 2, we explain related work in brief. Section 3 formulates the problem. The concept of mutual information and the proposed method are discussed in Section 4. Experimental results are shown in Section 5. Finally, conclusion and future work are discussed in Section 6.

2. Related work

Many feature selection algorithms (Bhattacharyya & Kalita, 2013; Caruana & Freitag, 1994; Frohlich, Chapelle, & Scholkopf, 2003; Lin, Ying, Lee, & Lee, 2012; Nemati, Basiri, Ghasem-Aghaee, & Aghdam, 2009; Yu & Liu, 2004) have been proposed for classification. The common approach for these algorithms is to search for an optimal set of features that provides good classification result. Most feature selection algorithms use statistical measures such as correlation and information gain or a population-based heuristic search approach such as particle swarm optimization, ant colony optimization, simulated annealing and genetic algorithms. An unsupervised feature subset selection method using feature similarity was proposed by Mitra, Murthy, and Pal (2002) to remove redundancy among features. They use a new measure called *maximal information compression index* to calculate the similarity between two random variables for feature selection. Bhatt and Gopal (2005) use fuzzy rough set theory for feature selection based on natural properties of fuzzy *t-norms* and *t-conorms*. A mutual information-based feature selection algorithm called MIFS is introduced by Battiti (1994) to select a subset of features. This algorithm considers both feature–feature and feature–class mutual information for feature selection. It uses a greedy technique to select a feature subset that maximizes information about the class label.

Kwak and Choi (2002) develop an algorithm called MIFS-U to overcome the limitations of MIFS to obtain better mutual information between input features and output classes than MIFS. Peng, Long, and Ding (2005) introduce a mutual information based feature selection method called mRMR (Max-Relevance and Min-Redundancy) that minimizes redundancy among features and maximizes dependency between a feature subset and a class label. The method consists of two stages. In the first stage, the method

incrementally locates a range of successive feature subsets where a consistent low classification rate is obtained. The subset with the minimum error rate is used as a candidate feature subset in stage 2 to compact further using a forward selection and backward elimination based wrapper method. Estévez, Tesmer, Perez, and Zurada (2009) propose a mutual information-based feature selection method as a measure of relevance and redundancy among features. Vignolo, Milone, and Scharcanski (2013) introduce a novel feature selection method based on multi-objective evolutionary wrappers using genetic algorithm.

To support effective data classification, several attribute evaluation techniques, such as ReliefF (Kira & Rendell, 1992), Chi squared (Liu & Setiono, 1995), Correlation Feature Selection (CFS) (Hall & Smith, 1999) and Principal components analysis (Ke & Sukthankar, 2004), also have been provided in the Weka software platform (Hall et al., 2009). These techniques employ different search techniques such as, BestFirst, ExhaustiveSearch, GreedyStepwise, RandomSearch, and Ranker for feature ranking. To describe the algorithms in this paper, we use symbols and notations given in Table 1.

2.1. Motivation

Feature selection or attribute selection is an important area of research in knowledge discovery and data mining. Due to rapid increase in the availability of datasets with numerous data types, an effective feature selection method is absolutely necessary for classification of data in high dimensional datasets. In many application domains, such as network security or bioinformatics, feature selection plays a significant role in the classification of data objects with high classification accuracy. Although a large number of classification and feature selection techniques have been used in the past, significant reduction of false alarms, especially in the network security domain remains a major and persistent issue. Decrease in the number of irrelevant features leads to reduction in computation time. This has motivated us to design a mutual information based feature selection method to identify an optimal subset of features which gives the best possible classification accuracy.

3. Problem formulation

For a given dataset D of dimension d, with a feature set $F = \{f_1, f_2, \ldots, f_d\}$, the problem is to select an optimal subset of relevant features F' where (i) $F' \subseteq F$ and (ii) for F', a classifier gives the best possible classification accuracy. In other words, we aim to identify a subset of features where for any pair $(f_i, f_j) \in F'$, the feature–feature mutual information is minimum and feature–class mutual information is maximum.

4. Proposed method

The proposed method uses mutual information theory to select a subset of relevant features. The method considers both feature–feature and feature–class mutual information to determine an optimal set of features.

Table 1
Symbols used in our method.

Symbols	Meaning	Symbols	Meaning
D	Dataset	d	Dimension of dataset
F	Original feature set	F'	Optimal feature set
C	Class label	k	Number of features in F'
f_i	Feature no i	f_j	Feature no j
C_d	Domination count	F_d	Dominated count
MI	Mutual information	FFMI	Feature feature mutual information
FCMI	Feature class mutual information	AFFMI	Average feature feature mutual information

4.1. Mutual information

In information theory, mutual information $I(X;Y)$ is the amount of uncertainty in X due to the knowledge of Y (Swingle, 2012; Kraskov, Stögbauer, & Grassberger, 2004). Mathematically, mutual information is defined as

$$I(X;Y) = \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)} \quad (1)$$

where $p(x,y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions for X and Y. We can also say

$$I(X;Y) = H(X) - H(X \mid Y) \quad (2)$$

where $H(X)$ is the marginal entropy, $H(X \mid Y)$ is the conditional entropy, and $H(X;Y)$ is the joint entropy of X and Y. If $H(X)$ represents the measure of uncertainty about a random variable, then $H(X \mid Y)$ measures what Y does not say about X. This is the amount of uncertainty in X after knowing Y and this substantiates the intuitive meaning of mutual information as the amount of information that knowing either variable provides about the other. In our method, a mutual information measure is used to calculate the

Table 2
Domination count of a feature.

Feature	Feature class MI	Average feature–feature MI	Domination count C_d	Dominated count F_d	$C_d - F_d$
f_1	0.54	0.17	2	1	1
f_2	0.76	0.09	3	0	3
f_4	0.17	0.78	0	3	−3
f_5	0.33	0.27	1	2	−1

Table 3
Domination count of a feature.

Feature	Feature class MI	Average feature–feature MI	Domination count C_d	Dominated count F_d	$C_d - F_d$
f_1	0.96	0.43	3	1	2
f_2	0.72	0.45	0	2	−2
f_4	0.85	0.78	1	3	−2
f_5	0.91	0.10	2	0	2

Table 4
Dataset description.

	Dataset	Number of instances	Number of attributes
Intrusion dataset	NSL-KDD 99	125,973	42
	10% KDD 99	494,021	42
	Corrected KDD	311,029	42
	Wine	178	13
Non-intrusion dataset	Monk1	432	6
	Monk2/Monk3	432	6
	IRIS	150	4
	Ionosphere	351	34
	Sonar	209	61
Text categorization	Bloggender-male	3232	101
	Bloggender-female	3232	101
Gene expression	Lymphoma	45	4026
	Colon cancer	62	2000

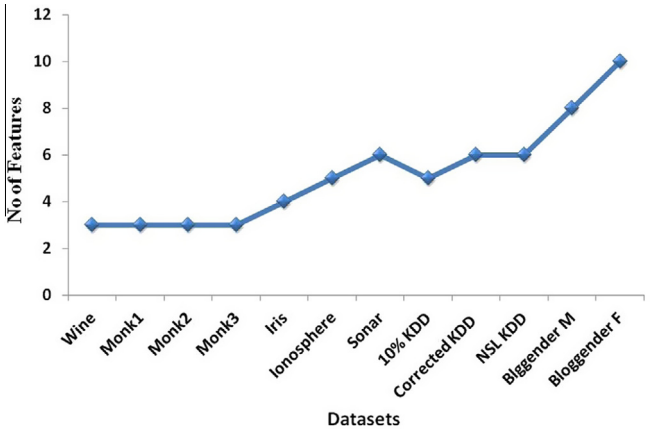


Fig. 1. Optimal range of the size of feature subsets for different datasets.

information gain among features as well as between feature and class attributes. Using a greedy manner, each time we pick a feature from the feature set still left one that provides maximum information about the class attribute with minimum redundancy.

4.2. Our feature selection method

Initially, we compute feature–class mutual information and select the feature that has the highest mutual information. The feature is then put into the selected feature subset and removed from the original feature set. Next, for each of the non-selected features, we compute the feature–class mutual information and then calculate the average feature–feature mutual information for each of the selected features. At this point, each non-selected feature contains feature–class mutual information and average feature–feature mutual information. From these calculated values, it selects a feature that has the highest feature–class mutual information, but minimum feature–feature mutual information using an optimization algorithm known as Non-dominated Sorting Genetic Algorithm-II (NSGA-II) (Deb, Agrawal, Pratap, & Meyarivan, 2000) with domination count C_d and dominated count F_d for feature–class and feature–feature values, respectively. The domination count of a feature represents the number of features that it dominates for feature–class mutual information and dominated count represents the number of features that it dominates for feature–feature mutual information. We select the feature that has the maximum difference of domination count and dominated count.

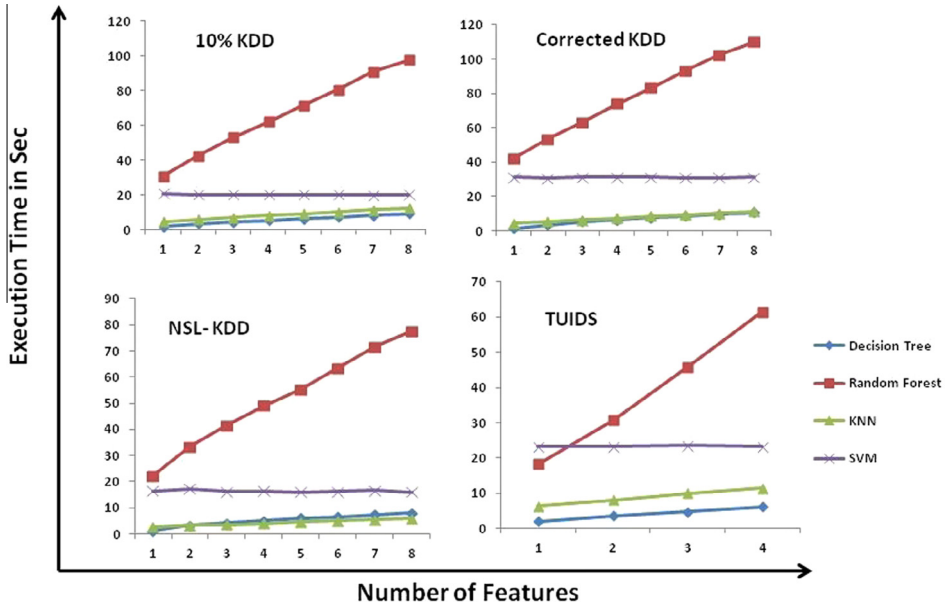
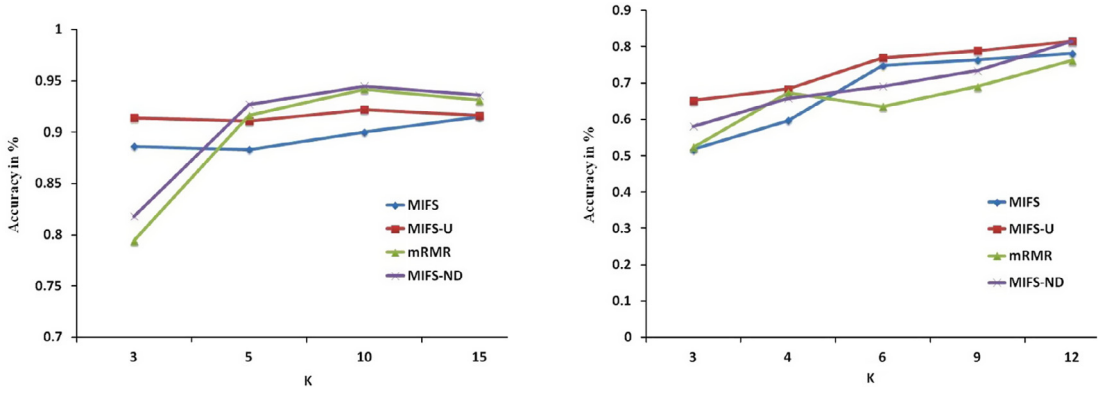
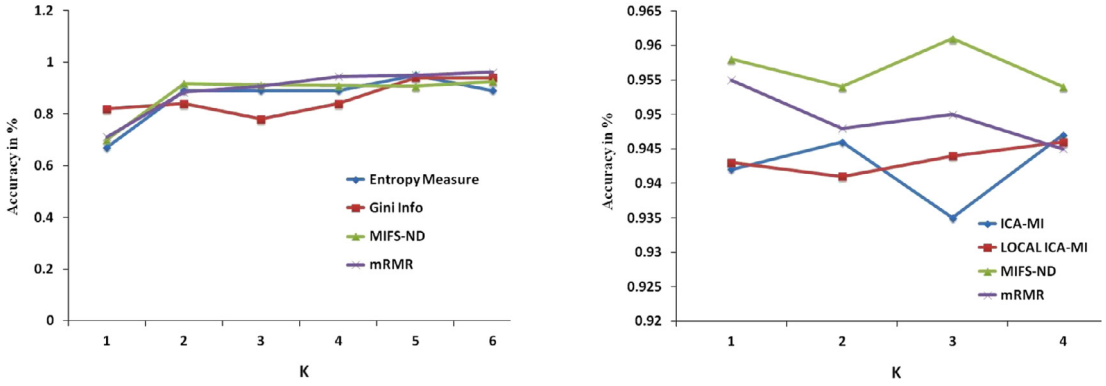


Fig. 2. Execution time performance for different classifiers.



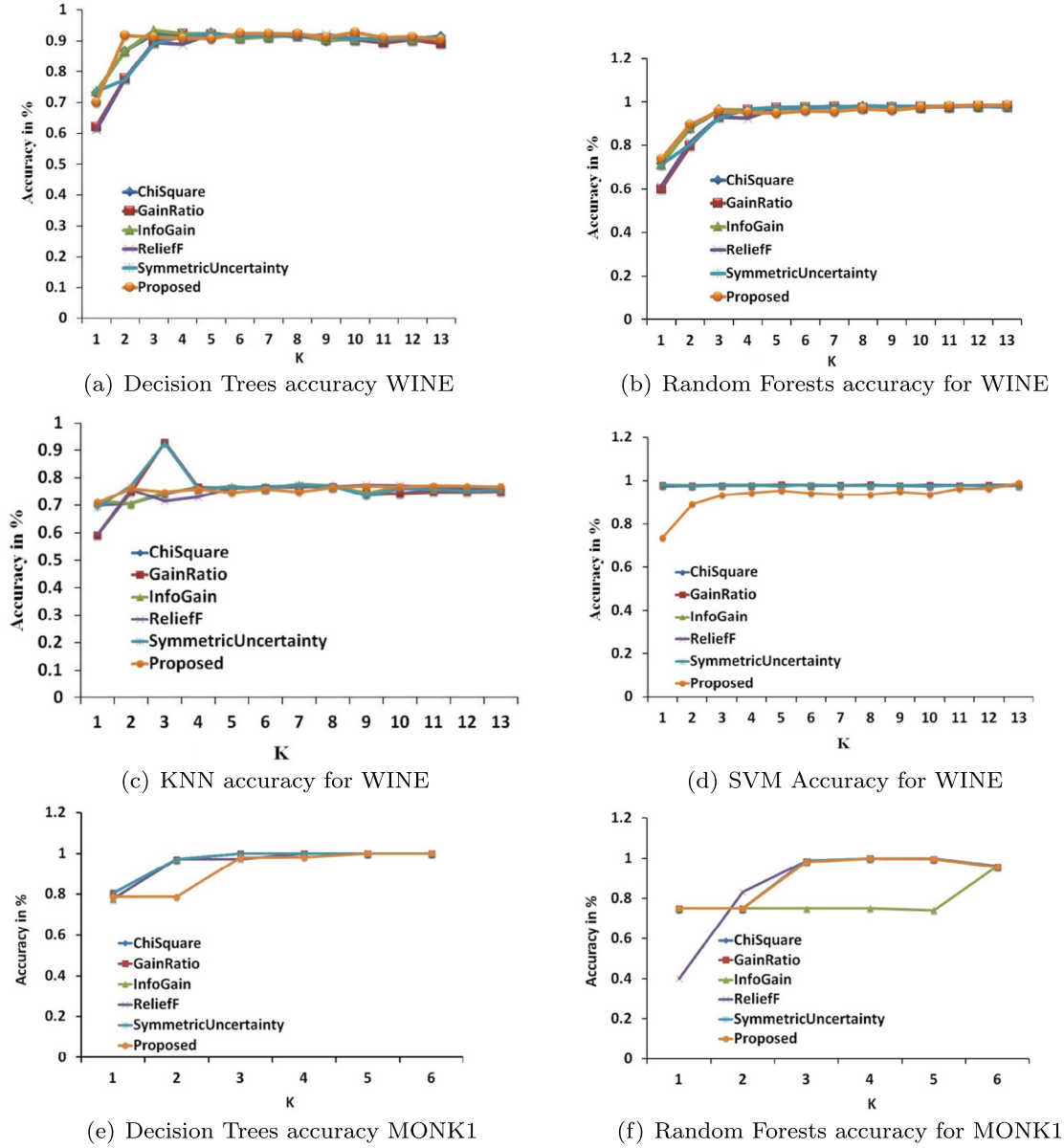
(a) Ionosphere dataset (b) Sonar dataset

Fig. 3. Comparison of MIFS-ND with MIFS, MIFS-U and mRMR for ionosphere and sonar datasets.



(a) Wine dataset (b) Iris dataset

Fig. 4. Comparison of MIFS-ND with entropy measure, GINI info, ICA-MI, local ICA-MI and mRMR.

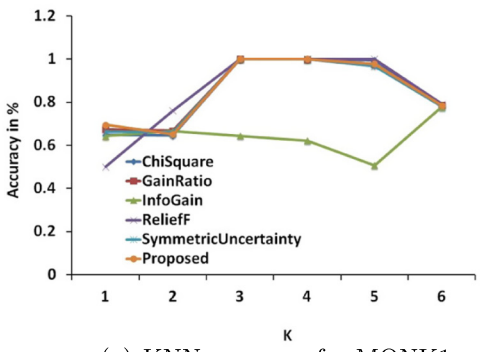


(a) Decision Trees accuracy WINE (b) Random Forests accuracy for WINE

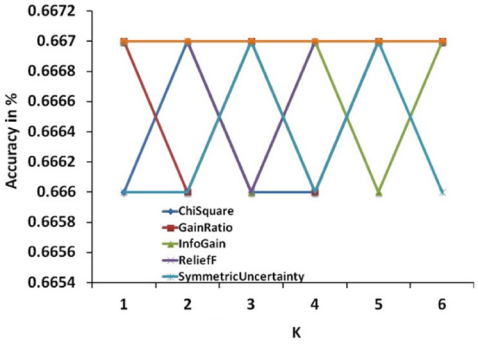
(c) KNN accuracy for WINE (d) SVM Accuracy for WINE

(e) Decision Trees accuracy MONK1 (f) Random Forests accuracy for MONK1

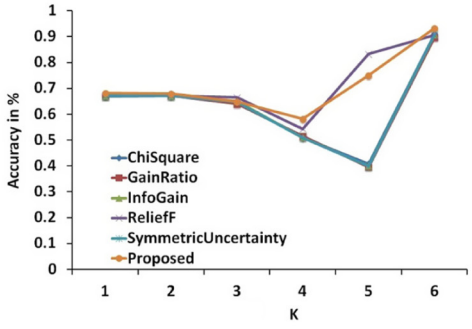
Fig. 5. Accuracy of different classifiers found in non-intrusion datasets.



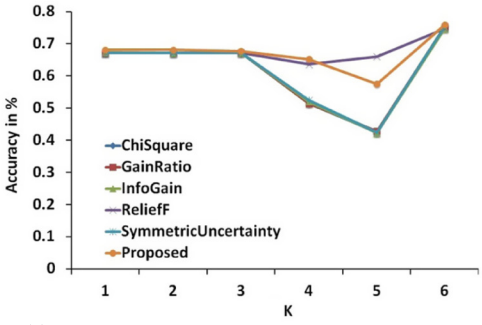
(g) KNN accuracy for MONK1



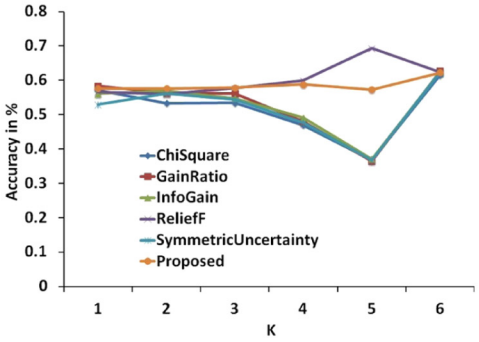
(h) SVM Accuracy for MONK1



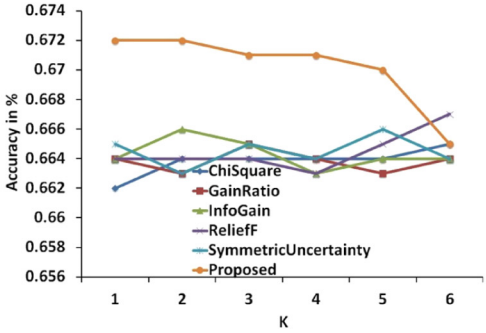
(i) Decision Trees accuracy MONK2



(j) Random Forests accuracy for MONK2



(k) KNN accuracy for MONK2



(l) SVM Accuracy for MONK2

Fig. 5 (*continued*)

4.2.1. Benefits of competitive ranking

The proposed feature selection method selects features based on a ranking procedure using the NSGA-II algorithm. It selects a feature that has the highest feature–class mutual information but minimum feature–feature mutual information, which requires solving an optimization problem. The soundness of the proposed method is derived from the fact that it selects a feature which is either strongly relevant or weakly redundant (explained in Examples 1 and 2). A strongly relevant feature has high feature–class mutual information but low feature–feature mutual information and a weakly relevant feature has high feature–class mutual information but medium feature–feature mutual information (Lutu & Engelbrecht, 2010). In addition, the method handles the tie condition (i.e., if two features have the same rank) by selecting the feature that has high feature–class mutual information.

4.2.2. Example 1

Let $F = \{f_1, f_2, f_3, f_4, f_5\}$ be a set of five features. First, we compute feature–class mutual information for every feature and select the feature that has the maximum mutual information value. Let us assume feature f_3 has the highest mutual information value and hence, f_3 is removed from F and is put in the optimal feature set, say F'. Next, for a feature $f_j \in F$, we compute feature–feature mutual information with every feature $f_i \in F'$ and store the average mutual information value as average feature–feature mutual information for f_j. This way, we compute feature–feature mutual information for f_1, f_2, f_4 and f_5. Again, we compute feature–class mutual information for f_1, f_2, f_4 and f_5. Consider the scenario shown in Table 2. Here, feature f_2 has the maximum difference between C_d and F_d, i.e., 3. Hence feature f_2 will be selected. In case of tie for the values of $(C_d - F_d)$, we pick the feature that has maximum feature–class mutual information. This procedure is continued until we get a subset of k features.

4.2.3. Example 2

For the situation shown in Table 3, the method selects the weakly relevant feature f_1 instead of feature F_5. Though the two features have the same $C_d - F_d$ value, F_1 has higher feature–class mutual information but medium feature–feature mutual information. In such a tie situation, the other methods may select either F_1 or F_5. According to Peng's Peng et al. (2005) and Battiti's Battiti (1994) method a tie condition for non-selected features X_1, X_2 and a selected features set X_k with class label Y is given by the following.

$I(X_1;Y) - \beta\sum_{k=1}^{n-1} I(X_1;X_k) = I(X_2;Y) - \beta\sum_{k=1}^{n-1} I(X_2;X_k)$

$I(X_1;Y) + \sum_{k=1}^{n-1} I(X_2;X_k) = I(X_2;Y) + \sum_{k=1}^{n-1} I(X_1;X_k)$

if $I(X_1;Y) > I(X_2;Y) \Rightarrow \sum_{k=1}^{n-1} I(X_2;X_k) < \sum_{k=1}^{n-1} I(X_1;X_k)$

In this scenario, the proposed method selects the feature X_1, since its feature–class mutual information is higher than X_2.

Definition 1 (*Feature relevance*). Let F be a full set of features, f_i a feature and $S_i = F - \{f_i\}$. Feature f_i is strongly relevant iff $I(C \mid f_i, S_i) \neq I(C \mid S_i)$ otherwise if $I(C \mid f_i, S_i) = I(C \mid S_i)$ and $\exists S'_i \subset S_i$ such that $I(C \mid f_i, S_i) \neq I(C \mid S_i)$, then f_i is weakly relevant to the class C.

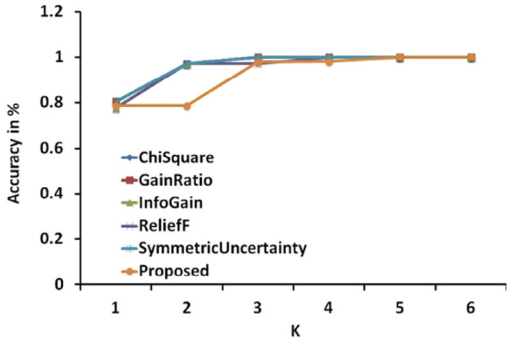
Definition 2 (*Feature–class relevance*). It is defined as the degree of feature–class mutual information for a given class C_i of a given dataset D, where data elements described by d features. A feature $f_i \in F'$, is an optimal subset of relevant features for C_i, if the relevance of $(f_i,\ C_i)$ is high.

Definition 3 (*Relevance score*). The relevance score of a feature f_i is the degree of relevance in terms of mutual information between a feature and a class label. Based on this value, a rank can be assigned to each feature $f_i,\ \forall i = 1, 2, 3, \ldots, d$. For a given feature $f_i \in F'$, the relevance score is high.

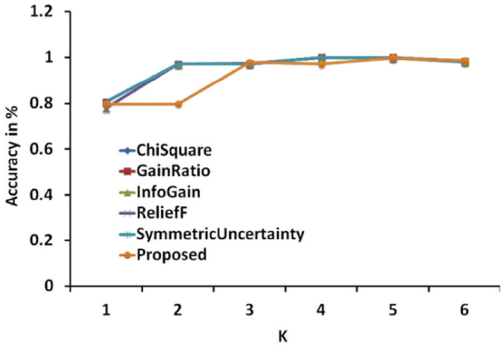
The following properties are trivial based on Peng et al. (2005) and Song, Ni, and Wang (2013).

Property 1. *For a pair of features $(f_i, f_j) \in F'$, the feature–feature mutual information is low, while feature–class mutual information is high for a given class C_i of a given dataset D.*

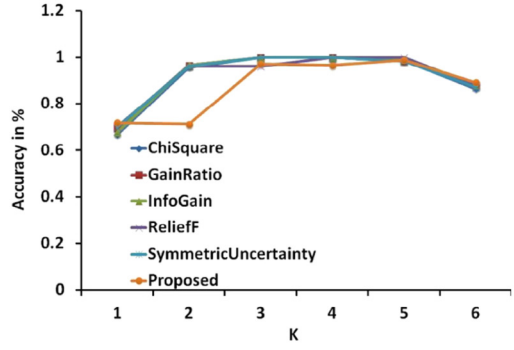
Explanation: For a feature f_i to be a member of an optimal subset of relevant features F' for a given class C_i, the feature–class mutual information or the relevance score (Definition 3) must be high. On the contrary, whenever feature–class mutual information is high, feature–feature mutual information has to be relatively low (Song et al., 2013).



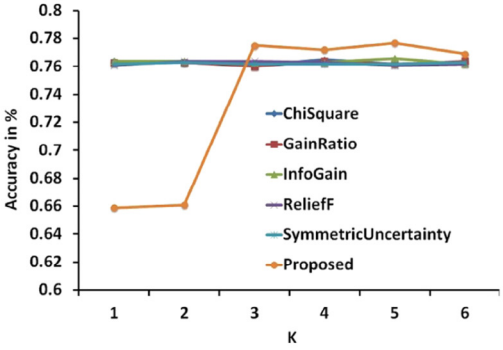
(m) Decision Trees accuracy MONK3



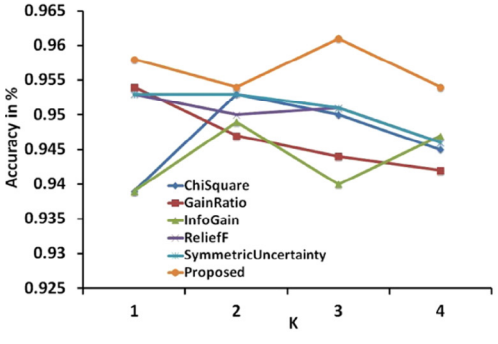
(n) Random Forests accuracy for MONK3



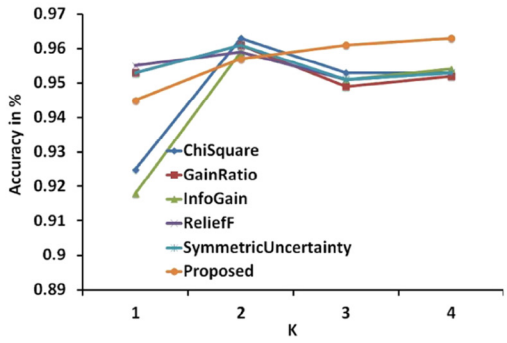
(o) KNN accuracy for MONK3



(p) SVM Accuracy for MONK3



(q) Decision Trees accuracy IRIS



(r) Random Forests accuracy for IRIS

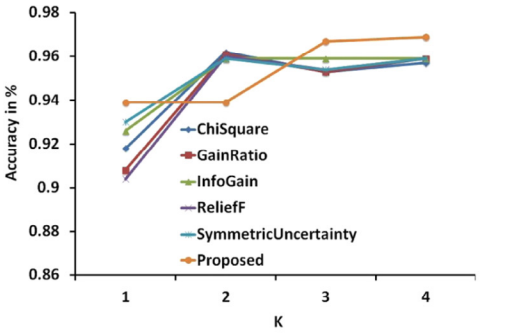
Fig. 5 (*continued*)

Property 2. *A feature $f_i' \notin F'$ has low relevance w.r.t. a given class C_i of a given dataset D.*

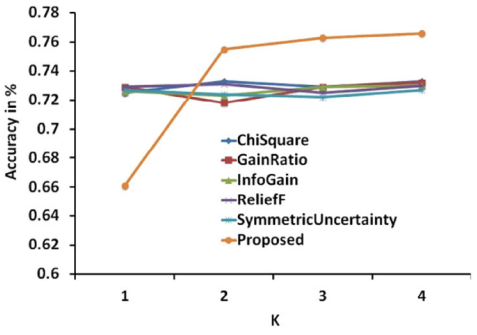
Explanation: Let $f_i' \notin F'$ where F' corresponds to class C_i and let feature–class mutual information between f_i' and C_i be high. If the feature–class mutual information score is high, then as per Definitions 1 and 2, f_i' must be a member of set F', which contradicts.

Property 3. *If a feature f_i has higher mutual information than feature f_j with the class label C, then f_i will have a smaller probability of misclassification (Frénay, Doquire, & Verleysen, 2013).*

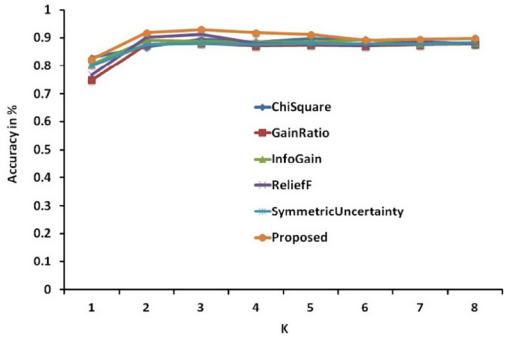
Explanation: Since f_i has higher mutual information score than f_j corresponding to class C, it has more relevance. So, the probability of classification accuracy using f_i as one of the feature will be higher than f_j.



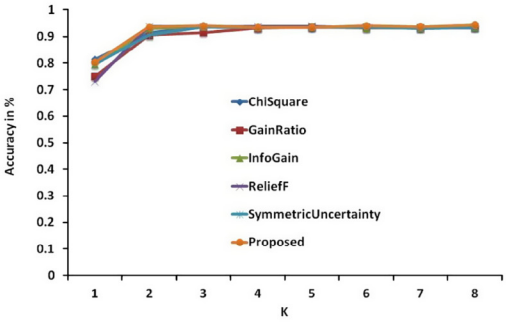
(s) KNN accuracy for IRIS



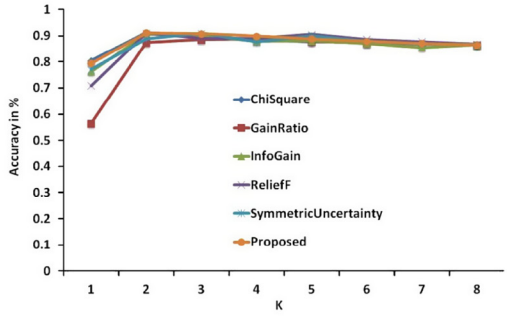
(t) SVM Accuracy for IRIS



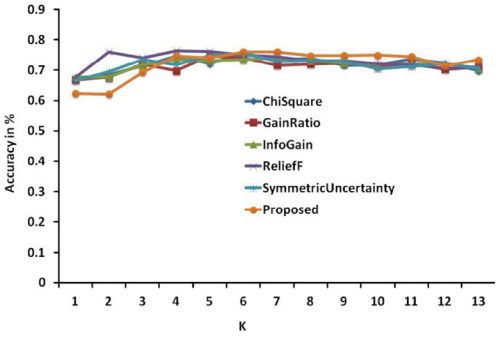
(u) Decision Trees accuracy Ionosphere



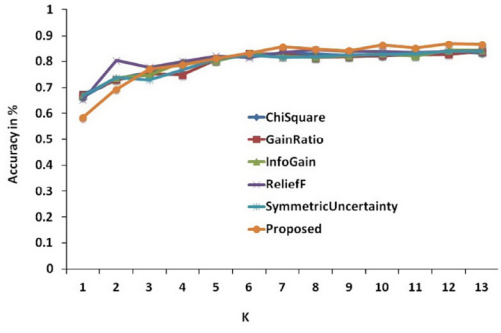
(v) Random Forests accuracy for Iono sphere



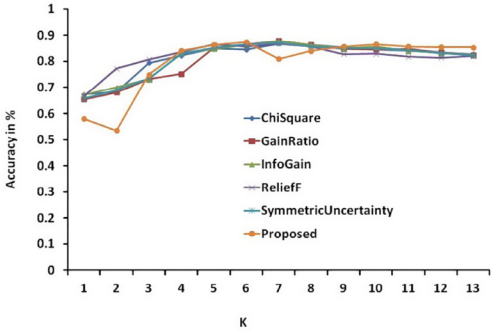
(w) KNN accuracy for Ionosphere



(x) Decision Trees accuracy Sonar



(y) Random Forests accuracy for Sonar



(z) KNN accuracy for Sonar

Fig. 5 (*continued*)

Lemma 1. *For a feature f_i, if the domination count C_d is larger and the dominated count F_d is smaller than all other features f_j, $(i \neq j)$, then the feature f_i has the highest feature–class mutual information and has more relevant.*

Proof. For a feature f_i, if its domination count C_d is larger and the dominated count F_d is smaller than all other features f_j, $(i \neq j)$ then NSGA-II method ensures that feature–class mutual information for F_i is the highest whereas the feature–feature mutual information value is the lowest. Hence, the method selects feature f_i as the strongly relevant feature as shown in Example 1. □

Lemma 2. *For any two features, f_i and f_j, if the difference of C_d and F_d for feature f_i is the same as the difference of C_d and F_d for feature f_j, then the feature with the highest feature–class mutual information is relevant.*

Proof. For any two features f_i and f_j, if the difference of C_d and F_d for feature f_i is as same as the difference of C_d and F_d for feature f_j, then NSGA-II ensures that neither f_i nor f_j satisfies the Lemma 1 and in this situation either f_i or f_j has higher feature–class mutual information. Hence, the method selects a feature that has higher feature–class mutual information as shown in Example 2. □

4.3. MIFS (MI based feature selection) method

Algorithm 1. MIFS-ND

Input: d, the number of features; dataset D;
$F = \{f_1, f_2, \ldots, f_d\}$, the set of features
Output: F', an optimal subset of features
Steps:
for i = 1 to d, **do**
  Compute $MI(f_i, C)$
end
Select the feature f_i with maximum $MI(f_i, C)$
$F' = F' \cup \{f_i\}$
$F = F - \{f_i\}$
count = 1;
while *count* $<= k$ **do**
  for each feature $f_j \in F$, **do**
    FFMI = 0;
    for each feature $f_i \in F'$, **do**
      FFMI = FFMI + compute_FFMI (f_i, f_j)
    end
    AFFMI = Average FFMI for feature f_j.
    FCMI = Compute_FCMI (f_j, C)
  end
  Select the next feature f_j that has maximum AFFMI but minimum FCMI
  $F' = F' \cup \{f_i\}$
  $F = F - \{f_j\}$
  $i = j$
  count = count + 1;
end
Return features set F'

The proposed feature selection method depends on two major modules, namely *Compute_FFMI* and *Compute_FCMI*. We describe working of each of these modules next.

Compute_FFMI (f_i, f_j): For any two features $f_i, f_j \in F$, this module computes mutual information between them, i.e.; f_i and f_j using Eq. 1. It computes marginal entropy for variable f_i and computes mutual information by subtracting conditional entropy of f_i for the given variable f_j from marginal entropy.

Compute_FCMI (f_j, C): For a given feature $f_j \in F$ and a given class label say C, this module is used to find mutual information between f_j and C using Shannon's mutual information formula using Eq. 1. First, it computes marginal entropy for the variable f_j and then subtracts conditional entropy of f_j for the given variable C.

Using these two modules the proposed method picks up a high ranked feature which is strongly relevant but non-redundant. To select a strongly relevant but non-redundant feature it uses NSGA-II method and computes the domination count (C_d) and dominated count (F_d) for every feature. If a feature has the highest difference between C_d and F_d then it selects that feature using Lemma 1 otherwise it uses Lemma 2 to select the relevant feature.

4.4. Complexity analysis

The overall complexity of the proposed algorithm depends on the dimensionality of the input dataset. For any dataset of dimension d, the computational complexity of our algorithm to select a subset of relevance features is $O(d^2)$. However, the use of appropriate domain specific heuristics can help to reduce the complexity significantly.

4.5. Comparison with other relevant work

The proposed feature selection method differs from other methods in the following manners.

1. Like Battiti (1994), our method considers both feature–feature and feature–class mutual information, but Battiti uses an additional input parameter called β to regulate the relative importance of mutual information between a feature to be selected and the features that have already been selected. Instead of the input β, our method calculates the domination count and dominated count, which are used in NSGA-II algorithm to select a relevant feature.
2. Like Kwak and Choi (2002), our method uses a greedy filter approach to select a subset of features. However, like them our method does not consider joint mutual information among three parameters, such as a feature that is to be selected, a set of features that are already selected and the class output.
3. Unlike Peng et al. (2005), who use both filter and wrapper approaches, our method uses only the filter approach to select an optimal feature subset using a ranking statistic, which makes our scheme more computationally cost effective. Similar to the Peng's mRMR method, we use the maximum relevance and minimum redundancy criterion to select a feature from the original feature set. In the subsequent section (Section 5) a detailed comparison of our method with Peng et al.'s method for two high dimensional gene expression datasets and four UCI datasets has been shown. Unlike Brown (2009), our method uses Shannon's mutual information on two variables only but Brown's method uses *multivariate* mutual information. Also, our method selects a feature set using a heuristic *bottom-up* approach and iteratively inserts a relevant but non-redundant feature into the selected feature set. Brown's method follows a *top-down* approach and discards features from the original feature set.
4. Unlike Kraskov et al. (2004), our method computes Shannon's entropy whereas they compute entropy using k-nearest neighbor distance.

5. Experimental result

Experiments were carried out on a workstation with 12 GB main memory, 2.26 Intel (R) Xeon processor and 64-bit Windows 7 operating system. We implement our algorithm using MATLAB R2008a software.

5.1. Dataset description

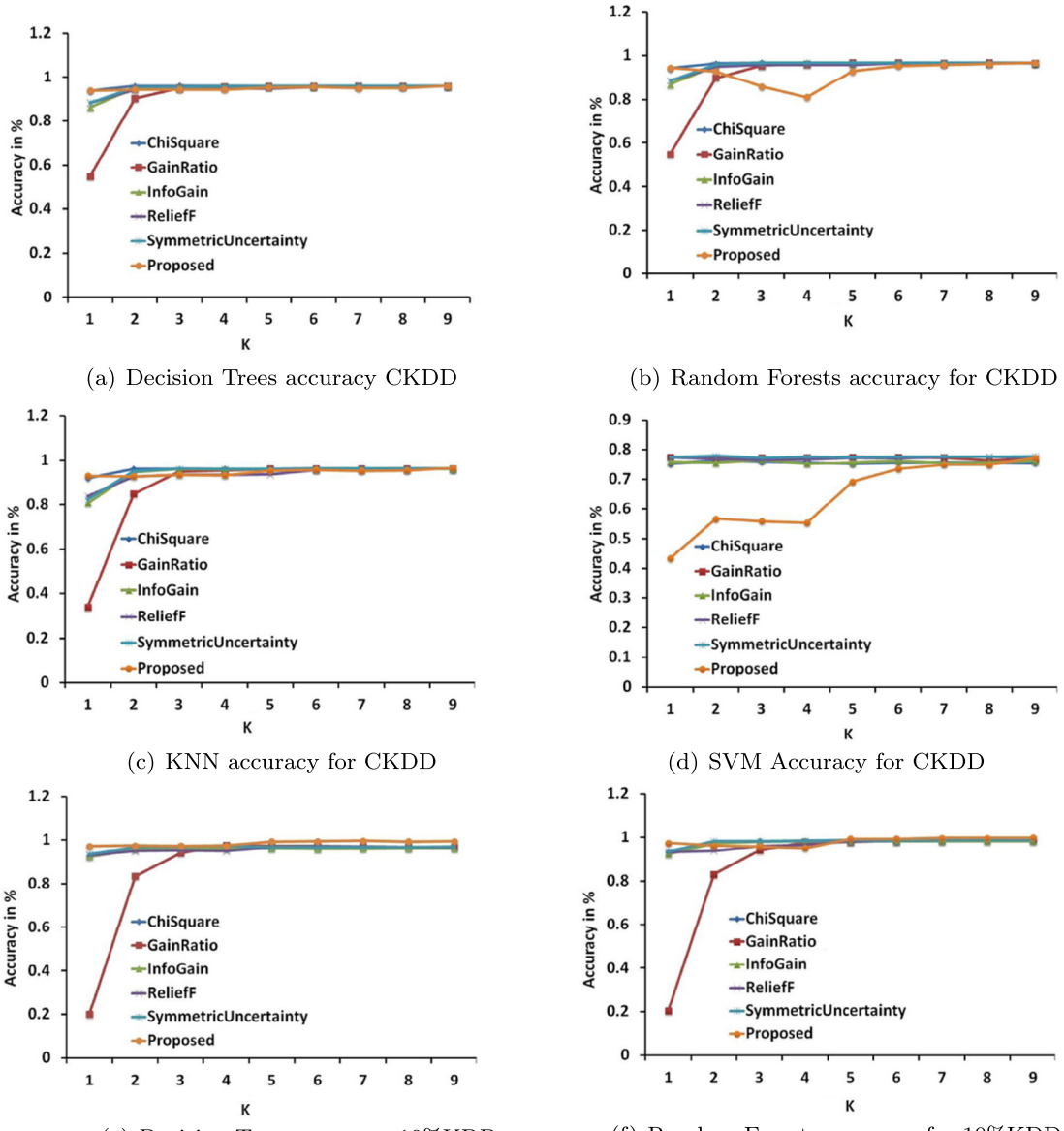
During our experimental analysis, we use several network intrusion, text categorization, a few selected UCI and gene expression datasets. These datasets contain both numerical and categorical values with various dimensionalities and numbers of instances. Descriptions of the datasets are given Table 4.

5.2. Results

The proposed algorithm first selects a subset of relevant features from each dataset. To evaluate the performance of our algorithm we use four well known classifiers, namely, Decision Trees, Random Forests, K-Nearest Neighbor (KNN) and Support Vector Machines (SVM). The performance of our algorithm is compared with five standard feature selection algorithms, namely Chi squared method, Gain Ratio, Information Gain, ReliefF, and Symmetric Uncertainty, which are already available in Weka. We use 10-fold cross validation to evaluate the effectiveness of selected features using different classifiers. The comparison of classification accuracy of our algorithm with all the aforesaid feature selection algorithms is shown here for all the mentioned datasets.

From the experimental results, we observe that the proposed method gives good classification accuracy on most datasets. Since, the network datasets are very large to be handled in MATLAB, we split the network datasets into smaller partitions which contain class labels from all the classes. Finally, we compute the average classification accuracy of all the partitioned datasets.

The proposed feature selection method selects an optimal subset of features for which we achieve the best classification accuracy. However, the cardinality of a feature subset identified by the proposed MIFS-ND method varies for different datasets. This variation in cardinalities of the optimal feature subsets for some of the datasets, namely, Sonar, Ionosphere, Wine and Iris is shown



(a) Decision Trees accuracy CKDD

(b) Random Forests accuracy for CKDD

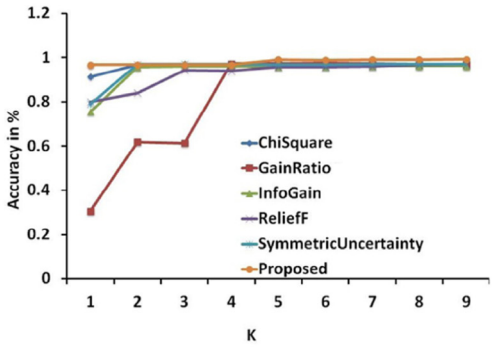
(c) KNN accuracy for CKDD

(d) SVM Accuracy for CKDD

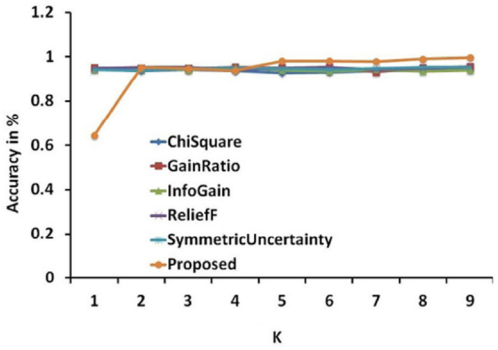
(e) Decision Trees accuracy 10%KDD

(f) Random Forests accuracy for 10%KDD

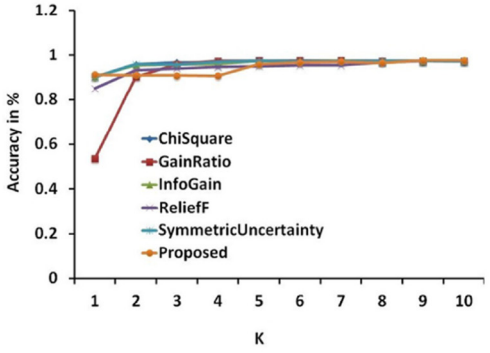
Fig. 6. Accuracy of different classifiers found in non-intrusion datasets.



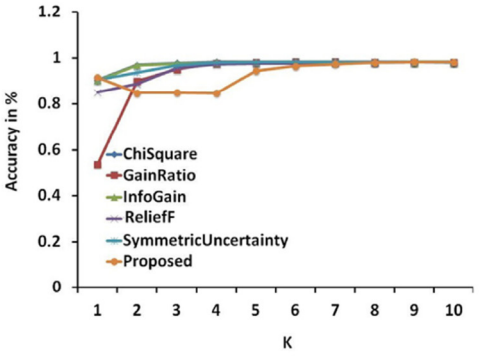
(g) KNN accuracy for 10%KDD



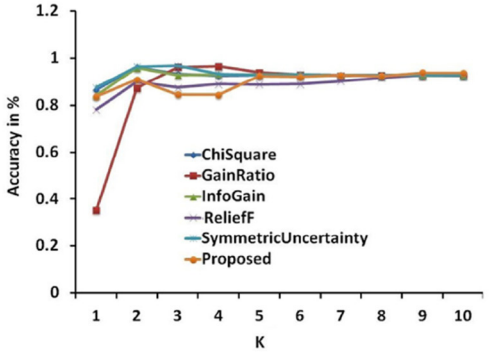
(h) SVM Accuracy for 10%KDD



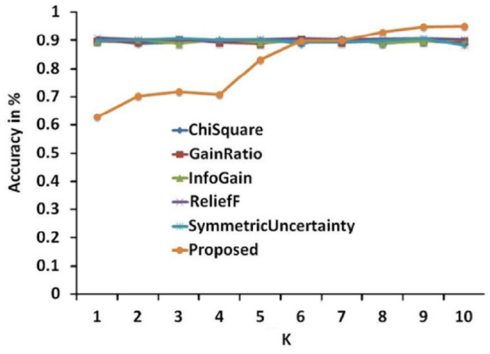
(i) Decision Trees accuracy NSL_KDD



(j) Random Forests accuracy for NSL_KDD



(k) KNN accuracy NSL_KDD



(l) SVM accuracy for NSL_KDD

Fig. 6 (*continued*)

in Figs. 3(a, b) and 4(a, b), respectively. Our observation is that (i) except for the Sonar dataset, the proposed MIFS-ND method performs significantly better for other datasets than the competing methods. The performance of our method suffers in case of Sonar dataset due to lack of sufficient training instances, and (ii) with the increase in size of the original feature sets of different datasets, the cardinality for optimal feature subset identified by MIFS-ND also increases. For example, in case of the UCI datasets, when the size of feature set varies from 4 to 61 (as shown in Table 4) the cardinality of the optimal feature subset varies from 3 to 6, whereas in case of the intrusion and text categorization datasets, the cardinality of the feature subsets varies from 5 to 10 due to the increased size of feature sets. Fig. 1 establishes this fact for 12 datasets.

5.3. Discussion

Feature selection is an unavoidable part for any classification algorithm. We use mutual information theory to select a subset of features from an original feature set based on feature–feature and feature–class information values. We evaluate our feature selection method using network security, text categorization, a few selected UCI and gene expression datasets. From experimental analysis, we observe that for most datasets, the classification accuracy is much better for a subset of features compared to when using the full feature set. Due to significant reduction of the number of features, better computational efficiency is also achieved. As shown in Fig. 2, the computational time increases significantly for Random Forests but computational time is constant for SVMs. The computational time is relatively increased for Decision Trees and KNN classifiers due to the increase in the number of features.

5.3.1. Performance analysis on intrusion datasets

In case of network datasets, we found high classification accuracy for all the classifiers with the 10% KDD dataset as shown in Fig. 6(e)–(h). We found better classification accuracy using Decision Trees, Random Forests and the KNN classifier than the SVM on the corrected KDD dataset for the proposed algorithm as shown in Fig. 6(a)–(d). Similarly, as shown in Fig. 6(i)–(l), our algorithm produces better result on the NSL_KDD dataset when $k \geqslant 6$, where

k is the number of features in a subset and when $k \leqslant 5$, the classification accuracy is relatively lower.

5.3.2. Performance analysis on UCI datasets

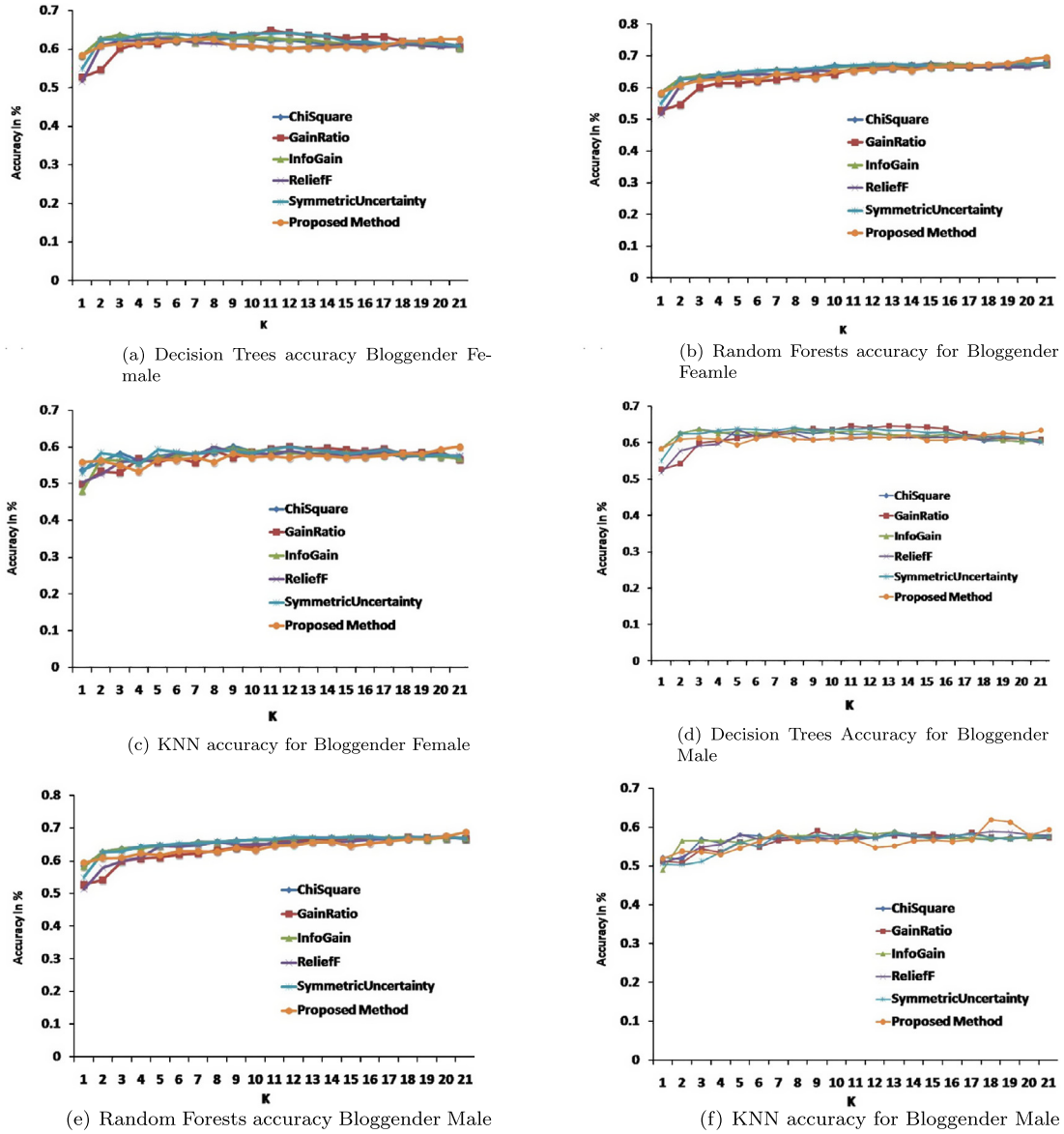
We perform experiments to analyze the classification accuracy of our proposed MIFS-ND method on UCI datasets. With the Wine dataset, Decision Trees, Random Forests and the KNN classifier give better classification accuracy for the proposed method but the accuracy is a bit lower with the SVM classifier, as plotted in Fig. 5(a)–(d), respectively. As shown in Fig. 5(e)–(t), the used classifiers show high classification accuracy with Monk1, Monk2, Monk3 and Iris datasets. We also compare the performance of our method MIFS-ND with MIFS (Battiti, 1994) and MIFS-U (Kwak & Choi, 2002) for Sonar and Ionosphere datasets as shown in Fig. 3(a) and (b), respectively. The comparison shows that the proposed algorithm gives better classification accuracy with these two UCI datasets.

5.3.3. Performance analysis on text categorization datasets

The proposed feature selection method is also applied to text datasets to evaluate its performance. From experimental results, we observe that the Bloggender female and male datasets show a bit poorer classification accuracy for Decision Trees as shown in Fig. 7(a) and (d), respectively. But, the method shows almost equal classification accuracy for Random Forests and KNN as plotted in Fig. 7(b), (c), (e) and (f), respectively.

5.3.4. Performance analysis on gene expression datasets

The proposed MIFS-ND method is also tested on two high-dimensional gene expression datasets, namely, Lymphoma and Colon Cancer. The classification accuracy on the Lymphoma dataset is very good for all the four classifiers as shown in Fig. 8(a)–(d). Compared to mRMR, MIFS-ND gives high classification accuracy for all four classifiers. However, the accuracy is compromised from feature number 7 for SVM classifier only. Similarly, in Colon Cancer dataset, Random Forests give high classification accuracy for MIFS-ND as shown in Fig. 8(f) compared to other feature selection methods. But KNN and SVM give high classification accuracy for the mRMR method as plotted in Fig. 8(g) and (h), respectively. In case of Decision Tree, classification accuracy for mRMR and MIFS-ND is almost similar as shown in Fig. 8(e).



(a) Decision Trees accuracy Bloggender Female

(b) Random Forests accuracy for Bloggender Feamle

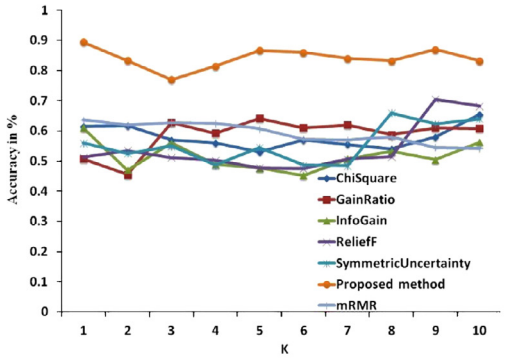
(c) KNN accuracy for Bloggender Female

(d) Decision Trees Accuracy for Bloggender Male

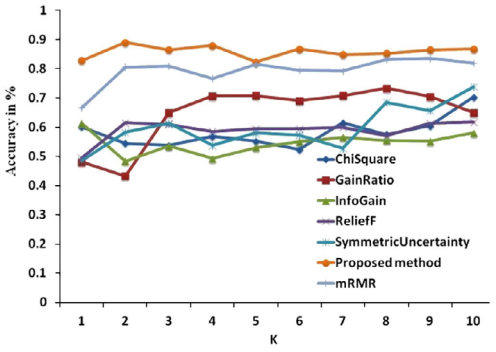
(e) Random Forests accuracy Bloggender Male

(f) KNN accuracy for Bloggender Male

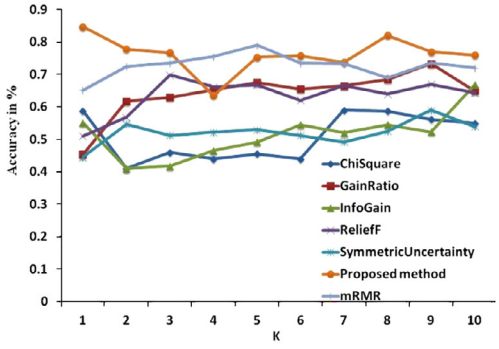
Fig. 7. Accuracy of different classifiers found in text categorization datasets.



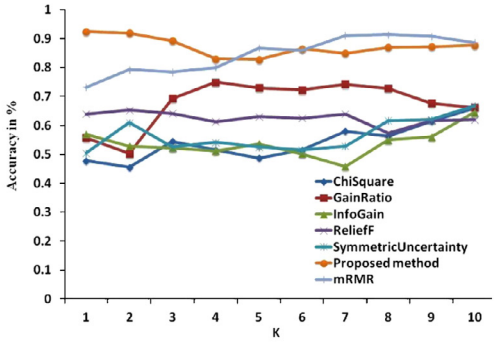
(a) Decision Trees accuracy for Lymphoma



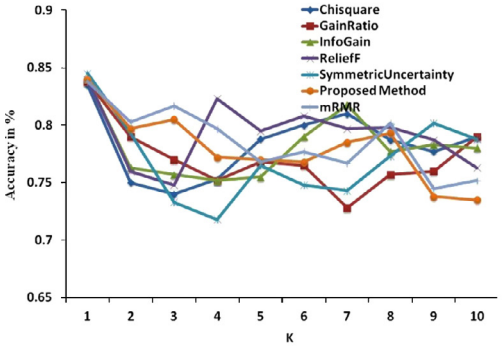
(b) Random Forests accuracy for Lymphoma



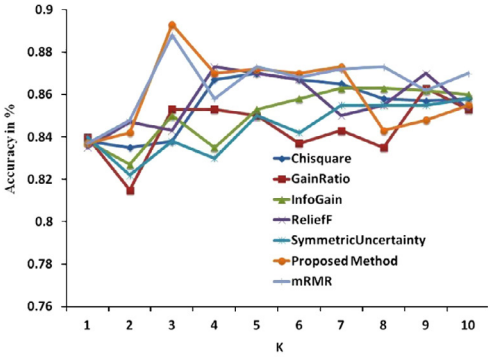
(c) KNN accuracy for Lymphoma



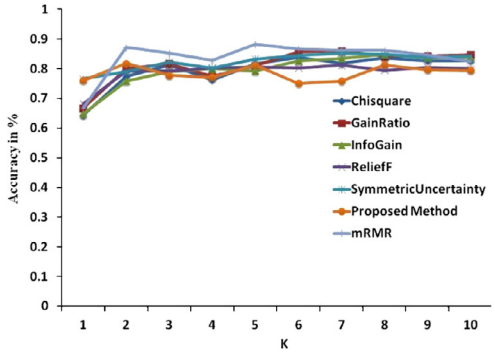
(d) SVM accuracy for Lymphoma



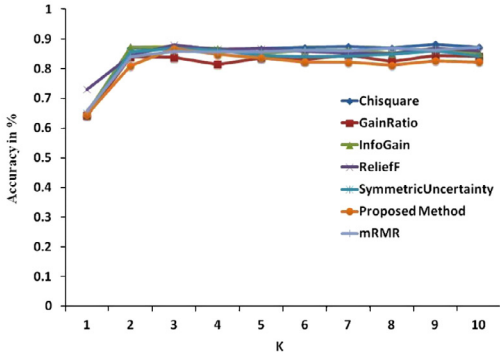
(e) Decision Trees Accuracy for Colon Cancer



(f) Random Forests accuracy Colon Cancer



(g) KNN accuracy for Colon Cancer



(h) SVM accuracy for Colon Cancer

Fig. 8. Accuracy of different classifiers for gene expression datasets.

5.4. Evaluation of classification performance based on t-test

We use the McNemar's test (Dietterich, 1998) to evaluate the performance of our method in terms of classification accuracy using four different classifiers. According to McNemar's test, two algorithms can have four possible outcomes such as *TT*, *TF*, *FT* and *FF*. Here, *T* and *F* stand for true and false respectively. The test computes a value called *z-score*, defined in Eq. (3) that measures how statistically significant the results are. The arrows (←, ↑) indicate which classifier performs better with a given dataset.

$$z = \frac{(|N_{TF} - N_{FT}| - 1)}{\sqrt{N_{TF} + N_{FT}}} \quad (3)$$

where N = number of instances in a dataset.

From the results of t-test, it can be observed that for Wine dataset, the Decision Trees performance is better than KNN and SVM but Random Forests perform better than Decision Trees for all the feature selection methods as shown in Table 5. In Monk1 dataset, KNN and Random Forests, both give better performance than Decision Trees, whereas Decision trees performance is better than SVM as shown in Table 6. In Monk2, Monk3 and Ionosphere dataset, Decision Trees classifier show better performance compared to KNN and SVM whereas Random Forests performance is better than Decision Trees as shown in Tables 7, 8 and 11. As given in Table 9, Decision Tree performance is better than SVM but KNN and Random Forests show better performance than Decision Trees. In Sonar dataset, the performance of KNN and Random Forests is better than Decision Trees whereas Decision Trees performance is better than that of SVM as shown in Table 10.

Table 5
t-Test for wine dataset.

Dataset	Feature selection method		KNN	RF	SVM
Wine	ChiSquar	DT	← 4.3410	1.5539 ↑	← 8.7818
	GainRatio		← 3.1201	2.2159 ↑	← 8.6927
	InfoGain		←3.4758	1.3790 ↑	← 8.8202
	ReliefF		← 3.7773	2.4859 ↑	← 8.6646
	SymmetricU		← 2.9723	1.9829 ↑	← 8.4879
	MIFS-ND		← 2.7097	2.6000 ↑	← 8.4917

Table 6
t-Test for Monk1 dataset.

Dataset	Feature selection method		KNN	RF	SVM
Monk1	ChiSquar	DT	3.1888 ↑	4.2504 ↑	← 12.6320
	GainRatio		3.2269 ↑	4.4338 ↑	← 12.5119
	InfoGain		← 1.8995	1.8943 ↑	←10.7272
	ReliefF		5.5811 ↑	5.5811 ↑	← 12.2406
	SymmetricU		3.7420 ↑	4.3032 ↑	← 12.6151
	MIFS-ND		2.7419 ↑	3.8916 ↑	← 12.4085

Table 7
t-Test for Monk2 dataset.

Dataset	Feature selection method		KNN	RF	SVM
Monk2	ChiSquar	DT	← 2.3840	← 1.0071	← 11.0991
	GainRatio		← 2.2699	← 0.5800	← 10.9840
	InfoGain		← 1.7675	← 1.2980	← 10.9856
	ReliefF		0.7857 ↑	0.2196 ↑	← 11.2472
	SymmetricU		← 2.8095	←1.4083	← 11.2910
	MIFS-ND		0.6572 ↑	0.5255 ↑	← 11.0219

Table 8
t-Test for Monk3 dataset.

Dataset	Feature selection method		KNN	RF	SVM
Monk3	ChiSquar	DT	← 2.6415	← 6.4908	← 13.8191
	GainRatio		← 2.4969	6.4944 ↑	← 13.9999
	InfoGain		← 1.3687	6.0628 ↑	← 13.7995
	ReliefF		0.6791 ↑	6.5169 ↑	← 13.7592
	SymmetricU		← 2.3657	6.2318 ↑	← 13.8910
	MIFS-ND		← 1.2637	← 5.9339	← 13.7014

Table 9
t-Test for Iris dataset.

Dataset	Feature selection method		KNN	RF	SVM
Iris	ChiSquar	DT	0.1215 ↑	1.5815 ↑	← 8.2768
	GainRatio		0.1250 ↑	1.7166 ↑	← 8.2622
	InfoGain		0.1768 ↑	1.6919 ↑	← 8.2618
	ReliefF		← 0.1250	1.2877 ↑	← 8.3372
	SymmetricU		0.6637 ↑	1.4462 ↑	← 8.3071
	MIFS-ND		0.2754 ↑	1.6303 ↑	← 8.2772

Table 10
t-Test for Sonar dataset.

Dataset	Feature selection method		KNN	RF	SVM
Sonar	ChiSquar	DT	0.7063 ↑	1.3416 ↑	← 7.2530
	GainRatio		1.6916 ↑	1.6897 ↑	← 7.5730
	InfoGain		1.8612 ↑	1.4681 ↑	← 6.6444
	ReliefF		1.4308 ↑	1.8329 ↑	← 7.0064
	SymmetricU		1.8296 ↑	2.5176 ↑	← 6.9518
	MIFS-ND		1.7365 ↑	1.8857 ↑	← 7.2041

Table 11
t-Test for Ionosphere dataset.

Dataset	Feature selection method		KNN	RF	SVM
Ionosphere	ChiSquar	DT	← 0.8715	1.4005 ↑	← 9.9160
	GainRatio		← 0.8175	1.3189 ↑	← 9.8643
	InfoGain		← 0.4200	1.6768 ↑	← 9.7896
	ReliefF		← 0.7503	1.7440 ↑	← 9.8320
	SymmetricU		← 0.6462	1.9885 ↑	← 9.7559
	MIFS-ND		1.6916 ↑	1.6897 ↑	← 7.5730

6. Conclusion and future work

In this paper, we have described an effective feature selection method to select a subset of high ranked features, which are strongly relevant but non-redundant for a wide variety of real-life dataset. To select high ranked features, an optimization criterion used in the NSGA-II algorithm is applied here. The method has been evaluated in terms of classification accuracy as well as execution time performance using several network intrusion datasets, a few UCI datasets, text categorization datasets and two gene expression datasets. We compare the classification accuracy for the selected features using Decision trees, Random Forests, KNN and SVM classifiers. The overall performance of our method has been found excellent in terms of both classification accuracy and execution time performance for all these datasets. Development of an incremental feature selection tool based on MIFS-ND is underway to support wide variety of application domains.

Acknowledgment

This work is supported by Department of Information Technology (DIT). The authors are thankful to the funding agency.

References

Arauzo-Azofra, A., Aznarte, J. L., & Benítez, J. M. (2011). Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications, 38*(7), 8170–8177.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks, 5*(4), 537–550.

Bhattacharyya, D. K., & Kalita, J. K. (2013). *Network anomaly detection: A machine learning perspective*. CRC Press.

Bhatt, R. B., & Gopal, M. (2005). On fuzzy-rough sets approach to feature selection. *Pattern Recognition Letters, 26*(7), 965–975.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*(1), 245–271.

Brown, G. (2009). A new perspective for information theoretic feature selection. In *International conference on artificial intelligence and statistics* (pp. 49–56).

Cadenas, J. M., Garrido, M. C., & MartíNez, R. (2013). Feature subset selection filter–wrapper based on low quality data. *Expert Systems with Applications, 40*(16), 6241–6252.

Caruana, R., & Freitag, D. (1994). Greedy attribute selection. In *ICML citeseer* (pp. 28–36).

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(3), 131–156.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-II. *Lecture Notes in Computer Science, 1917*, 849–858.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation, 10*(7), 1895–1923.

Estévez, P. A., Tesmer, M., Perez, C. A., & Zurada, J. M. (2009). Normalized mutual information feature selection. *IEEE Transactions on Neural Networks, 20*(2), 189–201.

Frénay, B., Doquire, G., & Verleysen, M. (2013). Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification. *Neurocomputing, 112*, 64–78.

Frohlich, H., Chapelle, O., & Scholkopf, B. (2003). Feature selection for support vector machines by means of genetic algorithm. In *15th IEEE international conference on tools with artificial intelligence, 2003. Proceedings.* (pp. 142–148). IEEE.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research, 3*, 1157–1182.

Hall, M. A., & Smith, L. A. (1999). Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *FLAIRS conference* (pp. 235–239).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter, 11*(1), 10–18.

Horng, J.-T., Wu, L.-C., Liu, B.-J., Kuo, J.-L., Kuo, W.-H., & Zhang, J.-J. (2009). An expert system to classify microarray gene expression data using gene selection by decision tree. *Expert Systems with Applications, 36*(5), 9072–9081.

Hsu, H.-H., Hsieh, C.-W., & Lu, M.-D. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications, 38*(7), 8144–8150.

Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory, 14*(1), 55–63.

Ke, Y., & Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004* (Vol. 2, pp. II-506). IEEE.

Khor, K.-C., Ting, C.-Y., & Amnuaisuk, S.-P. (2009). A feature selection approach for network intrusion detection. In *International Conference on Information Management and Engineering, 2009. ICIME'09.* (pp. 133–137). IEEE.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In *Proceedings of the ninth international workshop on machine learning* (pp. 249–256). Morgan Kaufman Publishers Inc..

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physics Review E*, American Physical Society, 69.

Kwak, N., & Choi, C.-H. (2002). Input feature selection for classification problems. *IEEE Transactions on Neural Networks, 13*(1), 143–159.

Lewis, D. D. (1992). Feature selection and feature extraction for text categorization. *Proceedings of the workshop on speech and natural language* (pp. 212–217). Association for Computational Linguistics.

Lin, S.-W., Ying, K.-C., Lee, C.-Y., & Lee, Z.-J. (2012). An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. *Applied Soft Computing, 12*(10), 3285–3290.

Liu, H., & Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. In *Seventh international conference on tools with artificial intelligence, 1995. Proceedings* (pp. 388–391). IEEE.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering, 17*(4), 491–502.

Lutu, P. E., & Engelbrecht, A. P. (2010). A decision rule-based method for feature selection in predictive data mining. *Expert Systems with Applications, 37*(1), 602–609.

Mitra, P., Murthy, C., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(3), 301–312.

Nemati, S., Basiri, M. E., Ghasem-Aghaee, N., & Aghdam, M. H. (2009). A novel aco–ga hybrid algorithm for feature selection in protein function prediction. *Expert Systems with Applications, 36*(10), 12086–12094.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1226–1238.

Polat, K., & Güneş, S. (2009). A new feature selection method on classification of medical datasets: Kernel f-score feature selection. *Expert Systems with Applications, 36*(7), 10367–10373.

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics, 23*(19), 2507–2517.

Song, Q., Ni, J., & Wang, G. (2013). A fast clustering-based feature subset selection algorithm for high dimensional data. *IEEE Transactions on Knowledge and Data Engineering, 25*(1), 1–14.

Swingle, B. (2012). Rényi entropy, mutual information, and fluctuation properties of fermi liquids. *Physical Review B, 86*(4), 045109.

Unler, A., Murat, A., & Chinnam, R. B. (2011). *mr^2pso*: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences, 181*(20), 4625–4641.

Vignolo, L. D., Milone, D. H., & Scharcanski, J. (2013). Feature selection for face recognition based on multi-objective evolutionary wrappers. *Expert Systems with Applications, 40*(13), 12086–12094.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML* (Vol. 97, pp. 412–420).

Yu, L., & Liu, H. (2004). Redundancy based feature selection for microarray data. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 737–742). ACM.