Neural Comput & Applic (2015) 26:507–523
DOI 10.1007/s00521-014-1673-2

Hybrid evolutionary algorithms for classification data mining

Mrutyunjaya Panda · Ajith Abraham

Received: 29 November 2013 / Accepted: 13 July 2014 / Published online: 10 August 2014

Abstract In this paper, we propose novel methods to find the best relevant feature subset using fuzzy rough set-based attribute subset selection with biologically inspired algorithm search such as ant colony and particle swarm optimization and the principles of an evolutionary process. We then propose a hybrid fuzzy rough with *K*-nearest neighbor (K-NN)-based classifier (FRNN) to classify the patterns in the reduced datasets, obtained from the fuzzy rough bio-inspired algorithm search. While exploring other possible hybrid evolutionary processes, we then conducted experiments considering (i) same feature selection algorithm with support vector machine (SVM) and random forest (RF) classifier; (ii) instance based selection using synthetic minority oversampling technique with fuzzy rough *K*-nearest neighbor (K-NN), SVM and RF classifier. The proposed hybrid is subsequently validated using real-life datasets obtained from the University of California, Irvine machine learning repository. Simulation results demonstrate that the proposed hybrid produces good classification accuracy. Finally, parametric and nonparametric statistical tests of significance are carried out to observe consistency of the classifiers.

Keywords Evolutionary algorithms · Fuzzy rough · Neural network · Bio-inspired algorithms · Classification · Statistical test

M. Panda (✉)
Gandhi Institute for Technological Advancement (GITA), Bhubaneswar 752054, India
e-mail: mrutyunjaya74@gmail.com

A. Abraham
Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, P.O. Box 2259, Auburn, WA 98071, USA
e-mail: ajith.abraham@ieee.org

1 Introduction

The large amount of data that are stored in databases contains valuable hidden knowledge which helps the user to improve the performance of decision-making process [1]. Feature selection is considered to be an important task inside machine learning with a focus on the most relevant features used in representing the data in order to delete those features considered as irrelevant making the knowledge discovery from data as simple. Feature subset selection represents the problem of finding an optimal subset of features of a dataset according to some criterion of selection, so that a classifier with the highest possible accuracy can be generated by an inductive learning algorithm that is run on data containing only the subset of features [2]. Based on whether a learning algorithm is included in the training process or not, existing feature selection (FS) approaches can be broadly classified into two categories: filter and wrapper approaches. A filter FS approach is a preprocessing procedure, and the search process is independent of a learning algorithm. In wrapper approaches, a learning algorithm is part of the evaluation function to determine the goodness of the selected feature subset. Wrappers can usually achieve better results than filters while filters are more general and computationally less expensive than wrappers [3]. A FS algorithm explores the search space of different feature combinations to reduce the number of features and simultaneously optimize the classification performance. In FS, the size of the search space for n features is $2n$. So in most situations, it is impractical to conduct an exhaustive search [3]. Therefore, the search strategy is the key part in FS. Different search techniques have been applied to FS such as greedy search, but most of them suffer from the problem of becoming stuck in local optima or high computational cost [4, 5]. Therefore, an

efficient global search technique is needed to develop a good FS algorithm. Evolutionary computation techniques are well known for their global search ability and have been applied to the FS problems. These include particle swarm optimization (PSO) [6, 7], genetic algorithms (GAs) [8] and genetic programming (GP) [9]. Compared with GAs and GP, PSO is easier to implement, has fewer parameters, computationally less expensive, and can converge more quickly [10]. Due to these advantages, two versions of PSO, namely continuous PSO and binary PSO, have been used for FS problems [6, 7, 11]. However, no study has been conducted to investigate the difference of using continuous PSO and binary PSO for FS. FS problems have two goals, which are maximizing the classification performance (or minimizing the classification error rate) and minimizing the number of features. These two objectives are usually conflicting, and there is a trade-off between them. However, most of the existing FS approaches, including PSO-based approaches, aim to maximize the classification performance only. Therefore, it is sought to use PSO to develop a multi-objective FS approach to simultaneously minimizing the classification error rate and minimizing the number of features selected. Ant colony optimization is inspired by the behaviors of ants and has many applications in discrete optimization problems. The approach relies on a metaheuristic which is used to guide other heuristics in order to obtain better solutions than those that are generated by local optimization methods; in ACO, a colony of artificial ants cooperates to look for good solutions to discrete problems [12]. ACO is particularly attractive for feature selection since there is no heuristic information that can guide search to the optimal minimal subset every time. On the other hand, if features are represented as a graph, ants can discover the best feature combinations as they traverse the graph [13]. One of the powerful approaches to dealing with the class imbalance problem is synthetic minority over-sampling technique (SMOTE). In this technique, SMOTE generates minority class within the overlapping regions. SMOTE has been widely used to solve imbalanced dataset problems in many medical area, such as medical imaging intelligence [14] and prostate cancer staging [15]. Rough set theory (RST) was proposed by Pawlak [16], which is a valid mathematic tool to handle imprecision, uncertainty and vagueness. As an effective method to feature selection, rough sets can preserve the meaning of the features. The essence of rough set approach to feature selection is to find a subset of the original features. Rough set theory provides a mathematical tool that can be used to find out all possible feature subsets. Unfortunately, the number of possible subsets is always very large when N is large because there are $2N$ subsets for N features. Hence, examining exhaustively all subsets of features for selecting the optimal one is NP-hard [17]. However, most often the values of attributes are continuous, but RST is applicable only on discretized data. In addition, after discretization, it is not possible to judge the extent to which the attribute value belongs to the corresponding discrete levels. This is the source of information loss, and it affects the classification accuracy negatively. Therefore, it is essential to work with real-valued data for combating the information loss, and this can be achieved by combining fuzzy and rough set theory [18]. The success of rough set theory is due in part to three aspects of the theory. First, only the facts hidden in data are analyzed. Second, no additional information about the data is required for data analysis such as thresholds or expert knowledge on a particular domain. Third, it finds a minimal knowledge representation for data. As rough set theory handles only one type of imperfection found in data, it is complementary to other concepts for the purpose, such as fuzzy set theory. The two fields may be considered analogous in the sense that both can tolerate inconsistency and uncertainty—the difference being the type of uncertainty and their approach to it; fuzzy sets are concerned with vagueness, and rough sets are concerned with indiscernibility. Many deep relationships have been established, and therefore, most recent studies have made conclusions about this complementary nature of the two methodologies, especially in the context of granular computing. Therefore, it is desirable to extend and hybridize the underlying concepts to deal with additional aspects of data imperfection. Such developments offer a high degree of flexibility and provide robust solutions and advanced tools for data analysis [19–21]. The K-nearest neighbor (KNN) algorithm is a well-known classification technique that assigns a test object to the decision class most common among test object. An extension of the KNN algorithm to fuzzy set theory (FNN) was introduced in [22]. It allows partial membership of an object to different classes and also takes into account the relative importance (closeness) of each neighbor w.r.t. the test instance. However, as Sarkar correctly argued in [23], the FNN algorithm has problems dealing adequately with insufficient knowledge. In particular, when every training pattern is far removed from the test object and, hence, there are no suitable neighbors, the algorithm is still forced to make clear-cut predictions. This is because the predicted membership degrees to the various decision classes always need to sum up to 1. Currently, the system based on neural network methods is one of the most accurate of all prediction systems; however, it poses some drawbacks [24]. Firstly, the black box nature of the neural network makes it difficult to understand how the networks predict the structure. Secondly, the systems based on neural net perform well if the query has many possibilities. On the other hand, the classification data mining using Nearest Neighbor methods does not suffer from any of such

drawbacks and is considered to be sub-optimal. Random forest is an ensemble of unpruned classification or regression trees, induced from bootstrap samples of the training data, using random feature selection in the tree induction process. Prediction is made by aggregating (majority vote for classification or averaging for regression) s the predictions of the ensemble [25]. Support vector machines (SVM) are becoming increasingly popular in the machine learning and computer vision communities. Training a SVM requires the solution of a very large quadratic programming (QP) optimization problem. In this paper, we use a variant of SVM for fast training using sequential minimal optimization (SMO) [26]. SMO breaks this large QP problem into a series of smallest possible QP problems avoiding large matrix computation. The amount of memory required for SMO is linear in the training set size, which allows SMO to handle very large training sets. SMO's computation time is dominated by SVM evaluation; hence, SMO is fastest for linear SVMs and sparse datasets. SVM ensembles can improve the limited classification performance of the SVM.

The rest of the paper is organized as follows: Sect. 2 provides literature search, Sect. 3 introduces about various evolutionary algorithms considered followed by the proposed methodologies used in Sect. 4. A brief discussion on the real-life dataset is provided in Sect. 5. Section 6 presents the experimental evaluations with discussions. Finally, Sect. 7 concludes the paper with future directions of research.

2 Related work

A hybrid adaptive particle swarm optimization aided learnable Bayesian classifier is proposed [27]. The objective of the classifier is to solve some of the fundamental problems associated with the pure naive Bayesian classifier and its variants with a larger view toward maximization of the classifier accuracy. In [28], the authors develop a PSO-based multi-objective FS approach to selecting a set of non-dominated feature subsets and achieving high classification performance and conclude to investigate the multi-objective PSO-based FS approach to better exploring the Pareto front of non-dominated solutions in FS problems as a future work. In [29], the authors present the biological motivation and some of the theoretical concepts of swarm intelligence with an emphasis on particle swarm optimization and ant colony optimization algorithms. The basic data mining terminologies are explained and linked with some of the past and ongoing works using swarm intelligence techniques. The paper [30] introduced the theoretical foundations of swarm intelligence with a focus on the implementation and illustration of particle swarm optimization and ant colony optimization algorithms. They provided the design and implementation methods for some applications involving function optimization problems, real-world applications and data mining. In [31], a bee colony optimization algorithm hybrid with rough set theory to find minimal reducts is proposed, which do not require any random parameter assumption. All these methods are analyzed using medical datasets. The authors argued that their proposed method exhibits consistent and better performance than the other methods with a saying that in future, the same approach can be extended to categorical attributes and also to handle missing values. In [32], the authors summarize that the rough neural networks (RNNs) are the neural networks based on rough set approaches which is a hot research area in the artificial intelligence in recent years, for the advantage of rough set to process uncertainly question: attributes reduce by none information losing then extract rule, and the neural networks have the strongly fault tolerance, self-organization, massively parallel processing and self-adapted. So that, RNNs can process the massively and uncertainly information, which is widespread applied in our life. Feature selection based on the fuzzy rough feature selection and tolerance-based feature selection on a number of benchmarks from the UCI repository was used in [33]. The authors [34] proposed a hybrid method by combining SMOTE and artificial immune recognition system (AIRS) to handle the imbalanced data problem that are prominent in medical data. They used the Wisconsin breast cancer (WBC) and Wisconsin diagnostic breast cancer (WDBC) datasets to compare the proposed method with other popular classifiers, i.e., AIRS, CLONALG, C4.5 and BPNN. In [35], a rough set attribute reduction algorithm that employs a search method based on particle swarm optimization (PSO) is proposed and compared with other rough set reduction algorithms and finally concluded that reducts found by their proposed algorithm are more efficient and can generate decision rules with better classification performance. The rough set rule-based method can achieve higher classification accuracy than other intelligent analysis methods such as neural networks, decision trees and a fuzzy rule extraction algorithm based on fuzzy min–max neural networks (FRE-FMMNN). In [36], a hybrid algorithm for instance and feature selection is discussed, where evolutionary search in the instances' space is combined with a fuzzy rough set-based feature selection procedure. The preliminary results, contrasted through nonparametric statistical tests, suggest that the proposal can improve greatly the performance of the preprocessing techniques in isolation. In [37], a granular neural network for identifying salient features of data, based on the concepts of fuzzy set and a newly defined fuzzy rough set, is proposed. The effectiveness of the proposed network, in evaluating

selected features, is demonstrated on several real-life datasets. The results of FRGNN are found to be statistically more significant than related methods in 28 instances of 40 instances, i.e., 70 % of instances, using the paired t test. The paper [38] has presented an approach to deal with system modeling and function approximation where the authors conclude that due to unknown relations in the condition attributes of some information systems, employing a neural network could not be helpful for approximating functionalities presented by such information systems. In some cases, it is easy to extract features and relations hided on data in an information system, but in many cases it could be impossible when we have no idea about what we are looking for. In [39], a modified feature selection technique based on fuzzy rough set theory and differential evolution is proposed. Here, the experimental results are carried out using binary and multiclass datasets taken from UCI machine learning repository.

3 Evolutionary algorithms

In this section, we will discuss on evolutionary algorithms that are used for the study of the classification data mining through out this paper.

3.1 Particle swarm optimization (PSO)

PSO is a population-based search algorithm and is initialized with a population of random solutions, called particles [40]. Unlike in the other evolutionary computation techniques, each particle in PSO is also associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles have the tendency to fly toward the better and better search area over the course of search process. The PSO was first designed to simulate birds seeking food which is defined as a 'cornfield vector' [41–45]. Assume the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. The birds do not know where the food is. But they know how far the food is and their peers' positions. So what is the best strategy to find the food? An effective strategy is to follow the bird which is nearest to the food. PSO learns from the scenario and uses it to solve the optimization problems. In PSO, each single solution is like a 'bird' in the search space, which is called 'particle.' All particles have fitness values which are evaluated by the fitness function to be optimized and have velocities which direct the flying of the particles (the particles fly through the problem space by following the particles with the best solutions so far). PSO is initialized with a group of random particles (solutions) and then searches for optima by updating each generation. The main PSO algorithm as described by Pomeroy [46] is given below: (Fig. 1).

3.2 Ant colonies optimization (ACO)

Ant colonies optimization (ACO) algorithms were introduced around 1990 [47–49]. These algorithms were inspired by the behavior of ant colonies. Ants are social insects, being interested mainly in the colony survival rather than individual survival. Of interest is ants' ability to find the shortest path from their nest to food. This idea was the source of the proposed algorithms inspired from ants' behavior. When searching for food, ants initially explore the area surrounding their nest in a random manner. While moving, ants leave a chemical pheromone trail on the ground. Ants are guided by pheromone smell. Ants tend to choose the paths marked by the strongest pheromone concentration. When an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. The indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources. As given by Dorigo et al. [50], the main steps of the ACO algorithm are given below:

1. *pheromone trail initialization*
2. *solution construction using pheromone trail*

Each ant constructs a complete solution to the problem according to a probabilistic state transition rule. The state transition rule depends mainly on the state of the pheromone [64].

3. *pheromone trail update.*

A global pheromone updating rule is applied in two phases. First, an evaporation phase where a fraction of the pheromone evaporates and then a reinforcement phase where each ant deposits an amount of pheromone which is proportional to the fitness of its solution [51]. This process is iterated until a termination condition is reached.

3.3 Synthetic minority over-sampling technique (SMOTE)

Synthetic minority over-sampling technique (SMOTE) [52] consists of the following steps:

Step 1 Take majority vote between the feature vector under consideration and its k-nearest neighbors for the

Pseudo code for PSO

```
/* set up particles' next location */
for each particle p do
{
for d = 1 to dimensions do
{
p.next[d] = random()
p.velocity[d] = random(deltaMin, deltaMax)
}
p.bestSoFar = initialFitness
}
/* set particles' neighbors */
for each particle p do
{
for n = 1 to numberOfNeighbors do
{
p.neighbor[n]=getNeighbor(p,n)
}
}
/* run Particle Swarm Optimizer */
while iterations ≤ maxIterations do
{
/* Make the "next locations" current and then*//* test their fitness. */
for each particle p do
{
for d = 1 to dimensions do
{
p.current[d] = p.next[d]
}
fitness = test(p)
if fitness > p.bestSoFar then do
{
p.bestSoFar = fitness
for d = 1 to dimensions do
{
p.best[d] = p.current[d]
}
}
if fitness = targetFitness then do
{
.../* e.g., write out solution and quit */
}
}/* end of: for each particle p */ for each particle p do
{
h = getNeighborWithBestFitness(p)
for d = 1 to dimensions do {
iFactor = iWeight * random(iMin, iMax)
sFactor = sWeight * random(sMin, sMax)
pDelta[d] = p.best[d] - p.current[d]
nDelta[d] = n.best[d] - p.current[d]
delta = (iFactor * pDelta[d]) + (sFactor * nDelta[d])
delta = p.velocity[d] + delta
p.velocity[d] = constrict(delta)
p.next[d] = p.current[d] + p.velocity[d]
}
}
/* end of: for each particle p */
}
/* end of: while iterations ≤ maxIterations */
end /* end of main program */ /* Return neighbor n of particle p */
function getNeighbor(p, n)
{
...return neighborParticle
}
/* Return particle in p's neighborhood */ /* with the best fitness */
function getNeighborWithBestFitness(p)
{
...return neighborParticle
}
/* Limit the change in a particle's */ /* dimension value */
function constrict(delta)
{
if delta < deltaMin then
return deltaMin
else
if delta > deltaMax then
return deltaMax
else
return delta
}
```

Fig. 1 Pseudo-code for PSO

nominal feature value. In the case of a tie, choose at random.

Step 2 Assign that value to the new synthetic minority class sample. Next step is to classify the data using FRS-NN, SVM or random forest classifier.

3.4 Rough set theory (RS)

Rough set theory is a mathematical approach for handling vagueness and uncertainty in data analysis. Objects may be indiscernible due to the limited available information. A rough set is characterized by a pair of precise concepts, called lower and upper approximations, generated using object indiscernibility. Here, the most important problems are the reduction of attributes and the generation of decision rules. In rough set theory, inconsistencies are not corrected or aggregated. Instead, the lower and upper approximations of all decision concepts are computed and rules are induced. The rules are categorized into certain and approximate (possible) rules depending on the lower and upper approximations, respectively. From the previous work [53, 54], rough set theory has been proved as a successful filter-based feature selection technique that performs better in data reduction, and it can be applied to many real-time problems. The three main aspects of the rough set theory are as follows:

- *Hidden facts in dataset are analyzed*
- *No additional information about the data is required*
- *Minimal knowledge is represented*

In real-time applications, there are many cases where the feature values are crisp and real-valued. Therefore, most traditional feature selection algorithms fail to perform well. To overcome this issue, an actual dataset is discretized before constructing a new dataset using crisp values. Here, the degrees of membership of the feature values to the discretized values are not examined and it leads to an inadequacy. So, it is clear that there is a prerequisite for feature selection techniques that can reduce the real-valued and crisp attributed datasets. Fuzzy theory and concept of fuzzification are the feature selection techniques that have emerged to provide an effective solution for real-valued features. This technique allows the feature values that belong to more than one class label with different degrees of membership and models the vagueness in the dataset. Again, it is exploited with fuzzy concepts, i.e., it enables an uncertainty in reasoning the dataset. To overcome the vagueness and indiscernibility in feature values, fuzzy and rough set theory is encapsulated to remove uncertainty in datasets. Fuzzy rough set theory [54] is an extended version of the crisp rough set theory. It takes the degree of membership values within the range of [0, 1]. It gives higher

Input: A set of instances
Output: A subset of features (B)
$B \leftarrow \{\ \}$;
repeat
$T \leftarrow \{B\}, best \leftarrow -1$;
For each $a \in \left(A \middle/ B\right)$ do
$\gamma_{B\cup\{a\}} > best$ then
$T \leftarrow B \cup \{a\},\ best \leftarrow \gamma_{B\cup\{a\}}$;
end
end
$B \leftarrow T$;
until $\gamma_B \geq Max\gamma$;

Fig. 2 Fuzzy rough QUICKREDUCT algorithm

flexibility when compared to crisp rough sets where it deals only with zero or full set membership. Fuzzy rough set is described by two fuzzy sets. They are lower and upper approximation. Fuzzy rough feature selection (FRFS) [55] can be effectively used to reduce the discrete and real-valued noisy attributes without any user information. In addition, this technique applies to both classification and regression problems that take the input value as continuous or nominal values. Information that is required to partition the fuzzy sets for feature vectors is automatically obtained from the datasets. It can also be replaced by other searching mechanisms such as swarm intelligence and ant colony optimization. The fuzzy rough QUICKREDUCT algorithm is provided in Fig. 2.

In FRFS, FR Quickreduct [56] is the basic algorithm that has been developed to find a minimal subset of feature vectors and it is represented in Fig. 1. It uses the fuzzy rough dependency function γ to select and add the feature values to reduct candidate. If adding any feature value to the reduct candidate fails to increase the degree of dependency, then the FR Quickreduct algorithm stops with the particular iteration. The FR Quickreduct algorithm calculates a reduct candidate with all possible subsets of feature values, but it lacks in comprehensiveness. It starts the iteration with an empty set and adds a feature value one by one after checking the constraint that fuzzy rough set dependency should be increased or else it should produce a maximum value for the actual dataset. Thus, the dependency of each feature value is ascertained using FR quick reduct algorithm, and the feasible candidate is chosen (Fig. 2).

3.5 Fuzzy nearest neighbor classification (FRKNN)

The fuzzy *K*-nearest neighbor (FNN) algorithm [57] was introduced to classify test objects based on their similarity to a given number *K* of neighbors (among the training objects), and these neighbors' membership degrees to (crisp or fuzzy) class labels. For the purposes of FNN, the extent $C(y)$ to which an unclassified object y belongs to a class C is computed as:

$$C(y) = \sum_{x \in N} R(x,y)C(x) \tag{1}$$

where N is the set of object y's K-nearest neighbors, and $R(x, y)$ is the [0, 1]-valued similarity of x and y.

Assuming crisp classes, Fig. 3 shows an application of the FNN algorithm that classifies a test object y to the class with the highest resulting membership.

Initial attempts to combine the FNN algorithm with concepts from fuzzy rough set theory were presented in [58, 59]. In these papers, a fuzzy rough ownership function is constructed that attempts to handle both "fuzzy uncertainty" (caused by overlapping classes) and "rough uncertainty" (caused by insufficient knowledge, i.e., attributes, about the objects). The pseudo-code is given in Fig. 4.

3.6 Support vector machine (SVM)

Sequential minimal optimization (SMO) is an algorithm for solving the optimization problem which arises during the training of support vector machines. It was invented by John Platt in 1998 at Microsoft Research [60]. SMO is

FNN (U, C, y, K).
U: the training data; C: the set of decision classes;
Y: the object to be classified; K: the number of nearest neighbors.
$N \leftarrow getNN(y, K)$;
$\forall C \in c$
$C(y) = \sum_{x \in N} R(x,y)C(x)$
Output $\arg\max_{C \in c}\left(C(y)\right)$

Fig. 3 The fuzzy KNN algorithm

FRNN(U, C, y)
U: the training data;
C: the set of decision classes;
y: the object to be classified
$N \leftarrow$ get Nearest Neighbors(y,K)
$\tau \leftarrow 0, Class \leftarrow \varnothing$
$\forall C \in c$
$if\left(\left(\left(R \downarrow C\right)(y) + \left(R \uparrow C\right)(y)\right)\Big/2 \geq \tau\right)$
$Class \leftarrow C$
$\tau \leftarrow \left(\left(R \downarrow C\right)(y) + \left(R \uparrow C\right)(y)\right)\Big/2$
Output Class

Fig. 4 The fuzzy rough nearest neighbor algorithm – classification

widely used for training support vector machines and is implemented by the popular LIBSVM tool. The publication of the SMO algorithm in 1998 has generated a lot of excitement in the SVM community, as previously available methods for SVM training were much more complex and required expensive third-party QP solvers.

3.7 Random forest (RF)

Random forests [61] grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

Each tree is grown as follows:

1. If the number of cases in the training set is N, sample N cases at random—but *with replacement*, from the original data. This sample will be the training set for growing the tree.
2. If there are M input variables, a number $m \ll M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
3. Each tree is grown to the largest extent possible. There is no pruning.

In the original paper on random forests, it was shown that the forest error rate depends on two things:

- The *correlation* between any two trees in the forest. Increasing the correlation increases the forest error rate.
- The *strength* of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.

Reducing m reduces both the correlation and the strength. Increasing it increases both. Somewhere in between is an "optimal" range of m—usually quite wide. Using the oob error rate (see below), a value of m in the range can quickly be found. This is the only adjustable parameter to which random forests is somewhat sensitive.

Features of random forests

- It is unexcelled in accuracy among current algorithms.
- It runs efficiently on large databases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in class population unbalanced datasets.
- Generated forests can be saved for future use on other data.
- Prototypes are computed that give information about the relation between the variables and the classification.
- It computes proximities between pairs of cases that can be used in clustering, locating outliers or (by scaling) give interesting views of the data.
- The capabilities of the above can be extended to unlabeled data, leading to unsupervised clustering, data views and outlier detection.
- It offers an experimental method for detecting variable interactions.

When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases is left out of the sample. These *oob* (*out-of-bag*) *data* are used to get a running unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance. After each tree is built, all of the data are run down the tree, and *proximities* are computed for each pair of cases. If two cases occupy the same terminal node, their proximity is increased by one. At the end of the run, the proximities are normalized by dividing by the number of trees. Proximities are used in replacing missing data, locating outliers and producing illuminating low-dimensional views of the data.

The out-of-bag (oob) error estimate

In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally, during the run, as follows:

- Each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases is left out of the bootstrap sample and not used in the construction of the kth tree.
- Put each case left out in the construction of the kth tree down the kth tree to get a classification. In this way, a test set classification is obtained for each case in about one-third of the trees. At the end of the run, take j to be the class that got most of the votes every time case n was oob. The proportion of times that j is not equal to the true class of n averaged over all cases is the oob error estimate. This has proven to be unbiased in many tests.

3.8 Statistical test of significance

3.8.1 Sign test

Sign test [62] is used to test whether one random variable in a pair tends to be larger than the other random variable in the pair. Given n pairs of observations. Within each pair, either a plus, tie or minus is assigned. The plus corresponds to that one value is greater than the other, the minus corresponds to that one value is less than the other, and the tie means that both equal to each other. The null hypothesis is that the number of pluses and minuses are equal. If the null hypothesis test is rejected, then one random variable tends to be greater than the other.

3.8.2 Wilcoxon's signed-rank test

This test is appropriate for matched pairs data, that is, for testing the significance of the relationship between a dichotomous variable and a continuous variable with related samples. It does assume that the difference scores are rankable, which is certain if the original data are interval scale.

Sign test only makes use of information of whether a value is greater, less than or equal to the other in a pair. Wilcoxon's signed-rank test [63, 64] calculates differences of pairs. The absolute differences are ranked after discarding pairs with the difference of zero. The ranks are sorted in ascending order. When several pairs have absolute differences that are equal to each other, each of these several pairs is assigned as the average of ranks that would have otherwise been assigned. The hypothesis is that the differences have the mean of 0.

3.8.3 Student's paired t test

A t test is used for testing the mean of one population against a standard or comparing the means of two populations if you do not know the populations' standard deviation and when you have a limited sample ($n < 30$). If you know the populations' standard deviation, you may use a z test [65].

3.8.4 z test

A z test is used for testing the mean of a population versus a standard, or comparing the means of two populations, with large ($n \geq 30$) samples whether you know the population standard deviation or not. It is also used for testing the proportion of some characteristic versus a standard proportion, or comparing the proportions of two populations.

4 Description of real-life dataset used

We evaluated the proposed method using a few of the benchmarks medical datasets from University of California, Irvine (UCI) repository [66].

4.1 Fisher's Iris data

Fisher is perhaps the best-known database to be found in the pattern recognition literature. The dataset contains 3 classes of 50 instances each, where each class refers to a type of Iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. The database contains the following attributes:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
    - Iris Setosa
    - Iris Versicolour
    - Iris Virginica

4.2 Pima Indians diabetes

The Pima Indians diabetes dataset is a well-known challenging pattern recognition problem from the UCI machine learning repository [66]. The dataset has 768 cases, all with the following numeric attributes:

1. Number of times pregnant
2. Plasma glucose concentration a 2 h in an oral glucose tolerance test
3. Diastolic blood pressure (mmHg)
4. Triceps skin fold thickness (mm)
5. 2-h serum insulin (mu U/ml)
6. Body mass index [weight in kg/(height in m)^2]
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

The class variable (9) is treated as a boolean: 0 (false) and 1 (true—tested positive for diabetes).

4.3 Tic-Tac-Toe

The popular children's game Tic-Tac-Toe pits two players against each other on a three-by-three grid, each player attempting to place three marks—X or O—in a straight line. If all nine spaces are filled with neither player creating a line of three marks, the game results in a draw, or what is commonly known as a Cat's game. While the gameplay is

straightforward enough, applying data mining techniques to the possible game boards provides some interesting information regarding the basic underlying strategies. Tic-Tac-Toe benchmark dataset encodes the complete set of possible board configurations at the end of Tic-Tac-Toe games, where "x" is assumed to play first. The target concept is "win of x" (i.e., true when "x" has one of 8 possible ways to create a "three-in-a-row"). In Tic-Tac-Toe dataset, there are total 958 instances/examples (626 positive examples and 332 negative examples), number of classes: 2 (positive and negative) and the number of attributes: 9 (each attribute corresponding to one Tic-Tac-Toe square and has 3 attribute values *x*, *o* and *b*), which follows:

1. A1 = top-left-square: {*x*, *o*, *b*}
2. A2 = top-middle-square: {*x*, *o*, *b*}
3. A3 = top-right-square: {*x*, *o*, *b*}
4. A4 = middle-left-square: {*x*, *o*, *b*}
5. A5 = middle-middle-square: {*x*, *o*, *b*}
6. A6 = middle-right-square: {*x*, *o*, *b*}
7. A7 = bottom-left-square: {*x*, *o*, *b*}
8. A8 = bottom-middle-square: {*x*, *o*, *b*}
9. A9 = bottom-right-square: {*x*, *o*, *b*}

4.4 Heart-C

This dataset was obtained from Cleveland database. Cleveland dataset concerns classification of person into normal and abnormal person regarding heart diseases.

- Data representation:

  Number of instances: 414.
  Number of attributes: 13 and a class attribute
  Class:
  Class0: normal person.
  Class1: first stroke
  Class2: second stroke
  Class3: end of life

- Attribute description:

  (i) Attribute description range

  Age: age in years continuous
  Sex: (1 = male; 0 = female) 0, 1
  Cp-value 1: typical angina 1, 2, 3, 4
  -Value 2: atypical anginal
  -Value 3: non-anginal pain
  -Value 4: asymptotic
  Trestbps: resting blood pressure (in mmHg) continuous
  Chol: serum cholesterol in mg/dl Continuous
  fbs: (Fasting blood sugar .120 mg/dl) 0, 1 (1 = true; 0 = false)
  restecg: electrocardiography results 0, 1, 2
  -Value 0: normal
  -Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of >0.05 mV)
  -Value 2: showing probable or definite left Ventricular: hypertrophy by Estes' criteria
  Thalach: maximum heart rate achieved continuous
  Exang: exercise induced angina (1 = yes; 0 = no) 0, 1
  Oldpeak: ST depression induced by exercise relative to rest Continuous
  Slope: the slope of the peak exercise
  ST: segment 1, 2, 3
  Value 1: up sloping
  Value 2: flat
  Value 3: down sloping
  Ca: number of major vessels (0–3)
  Colored by fluoroscopy continuous
  Thal: normal, fixed defect, reversible defect 3, 6, 7

  (ii) Linear data scaling

  Here, each value is converted into the range between 0 and 1 using the following formulae
  $\Delta = X_{\max} - X_{\min}$
  $Y =$ intercept $C = (X - X_{\min})/\Delta$
  Slope $= m =$ 1/Delta
  So, we calculate Y for a given X as $Y = mX + C$.

4.5 Hepatitis dataset

The dataset contains 155 instances distributed between two classes die with 32 instances and live with 123 instances. There are 19 features or attributes, 13 attributes are binary while 6 attributes with 6–8 discrete values. The goal of the dataset is to forecast the presence or absence of hepatitis virus. Table 1 lists information about the features.

4.6 Breast cancer-W dataset

The dataset breast cancer-Wisconsin currently contains 699 instances with 2 classes (malignant and benign), 9 integer-valued attributes and the following Attribute Information as given in Table 2.

These data contain 16 missing attribute values. There are 16 instances in Groups 1–6 that contain a single

Table 1 Information about the features of the hepatitis dataset

Number	Name of features	The values of features
1	Age	10, 20, 30, 40, 50, 60, 70, 80
2	Sex	Male, female
3	Steroid	Yes, no
4	Antivirals	Yes, no
5	Fatigue	Yes, no
6	Malaise	Yes, no
7	Anorexia	Yes, no
8	Liver big	Yes, no
9	Liver firm	Yes, no
10	Spleen palpable	Yes, no
11	Spiders	Yes, no
12	Ascites	Yes, no
13	Varices	Yes, no
14	Bilirubin	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
15	Alk phosphate	33, 80, 120, 160, 200, 250
16	SGOT	13, 100, 200, 300, 400, 500
17	Albumin	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
18	Protime	10, 20, 30, 40, 50, 60, 70, 80, 90
19	Histology	Yes, no

Table 2 Breast cancer-W dataset attribute information

# Attribute	Domain
1. Sample code number	Id number
2. Clump thickness	1–10
3. Uniformity of cell size	1–10
4. Uniformity of cell shape	1–10
5. Marginal adhesion	1–10
6. Single epithelial cell size	1–10
7. Bare nuclei	1–10
8. Bland chromatin	1–10
9. Normal nucleoli	1–10
10. Mitoses	1–10
11. Class	2 for benign,

missing (i.e., unavailable) attribute value, now denoted by "?". Class distribution: benign: 458 (65.5 %) and malignant: 241 (34.5 %)

5 Proposed methodology

The proposed system design is diagrammatically presented in Fig. 5. The data mining framework for the classifier is viewed from the perspective of both the training/learning phase and the test phase. The dataset is visualized and preprocessed before applying any of the data mining techniques The training phase then makes the learning process complete by generating all possible rules for classification after performing feature relevance followed by classification. The test phase determines the accuracy of the classifier when presented with a test data and by viewing the returned class label.

Our experiments were conducted in two ways. First, with a supervised attribute filter that can be used to select attributes. It is very flexible and allows various search and evaluation methods to be combined. The proposed bio-inspired algorithm search such as ACO and PSO determines the search method while the evaluator determines how attributes/attribute subsets are evaluated. Secondly, with a supervised instance-based feature subset selection method where a dataset is resample by applying SMOTE. Here, class value is the index of the class value to which SMOTE should be applied with a value of 0 to auto-detect the non-empty minority class and the number of nearest neighbors to use. The original dataset must fit entirely in memory. We then compared the performance of ACOFSS, PSOFSS and SMOTE as feature subset selection methods in order to obtain the best attributes for the classification data mining purpose. In attribute selection methods, we combine ACO and PSO with rough and fuzzy rough-based attribute evaluators to obtain the best possible set of features for efficient classification. On the other hand, in instance-based feature selection method, SMOTE is used with 5-nearest neighbor presentation. Once, the feature selection is done, the next step is to apply the data with the reduced feature set to the classifier where we propose to use support vector machine (SVM), random forest (RF) and fuzzy rough *K*-nearest neighbor (FRKNN) for classification data mining. We used Lukasiewitz fuzzy implication operator in place of Kleene–Dienes implications, for getting a better classification result. At last, the experimental results will be validated with the full training set,

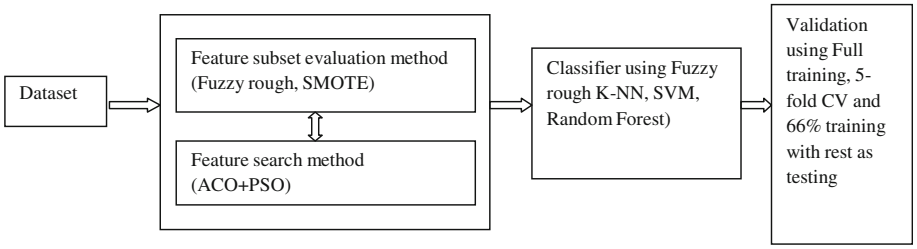

Fig. 5 Proposed classification data mining process

fivefold cross-validation and with using 66 % training set and rest for testing purpose.

Now, in order to test the statistical significance of all the used algorithms on all datasets, we use parametric test (t test and z test) and nonparametric test (Sign test and Winkolson's signed-rank test) for classification accuracy.

6 Experimental results and discussions

In this section, the effectiveness of the various hybrid evolutionary algorithms is demonstrated on different real-life datasets obtained from UCI repository. All the experiments are carried out in Java environment [67] in a Intel PIV, Windows XP, 2.66 GHz CPU, 512 MB RAM with all the default parameters. First, we carried out feature subset selection on the datasets using FRFS and ACO (or PSO) with FRKNN classifier (with $K = 10$) methods. The obtained results are shown in Tables 3 and 4 with the classification accuracy, the reduced feature subset size). Seeing the effectiveness of PSO over ACO, we conduct more experiments using FRSE and PSO with SVM and then with RF classifier, as a hit and trial method for obtaining better classification accuracy. These can be observed from Tables 5 and 6. Further, we used SMOTE as an instance selection method to select the desired instances and then classified the data with FRKNN, SVM and RF classifier. The results are provided in Tables 7, 8 and 9. All the tables are provided with best reduced features, % feature reductions as ratio between total selected features to the total number of features available in the respective dataset, classification accuracy obtained from using (i) whole training set, (ii) fivefold cross-validation and (iii) 66 % training and the rest as testing data for all the methodologies discussed, and lastly with fitness value obtained from the following formula:

$$\text{Fitness Value} = \alpha * \text{Accuracy Rate} + (1 - \alpha)\text{Reduction Rate} \quad (2)$$

where accuracy rate is the accuracy achieved by the classifier during the classification process, and the reduction rate as the ratio between the numbers of features selected to the total number of features available in the dataset. The evaluation criteria using fitness value is a crucial issue in the implementation of stochastic algorithms, as the classification quality parameter outclasses that of subset length. Here, we use the constant $\alpha = 0.5$. From the results, it is quite evident that the fitness values are to be improved further.

Table 3 FRSE + ACO + FRKNN, $K = 10$; average accuracy with fivefold CV = 71.465 %

Dataset	Actual features	Reduced best features	% Reductions	Training accuracy (%)	Fivefold cross-validation accuracy (%)	66 % training and rest testing accuracy (%)	Fitness value
IRIS	5	5	100	100	88.67	90.2	0.951
Tic-Tac-Toe	10	2 (4, 10)	20	65.36	34.7	30.98	0.255
Hepatitis	20	13 (1, 2, 3, 4, 5, 6, 11, 13, 14, 16, 17, 19, 20)	65	99.4	69.1	66.1	0.655
Pima diabetis	9	9	100	100	67.6	70.2	0.851
Wisconsin breast cancer	10	8 (1, 2, 4, 5, 6, 7, 9, 10)	80	100	95.42	95.8	0.879
Heart-C	14	8 (1, 3, 4, 5, 8, 10, 12, 14)	57.14	100	73.3	69.9	0.635

Table 4 FRSE + PSO + FRNN; average accuracy with fivefold CV = 78.52 %

Dataset	Actual features	Reduced Best features	% Reductions	Training accuracy (%)	fivefold cross-validation	66 % training and rest testing	Fitness value
IRIS	5	5	100	100	88.67	90.2	0.95
Tic-Tac-Toe	10	9 (2, 3, 4, 5, 6, 7, 8, 9, 10)	90	100	67.01	63.5	0.767
Hepatitis	20	12 (1, 2, 3, 4, 6, 9, 10, 11, 14, 15, 19, 20)	60	99.36	76.8	79.3	0.684
Pima diabetis	9	9	100	100	67.6	70.2	0.851
Wisconsin breast cancer	10	8 (1, 2, 4, 5, 6, 7, 9, 10)	80	100	95.42	95.8	0.879
Heart-C	14	9 (1, 3, 5, 7, 8, 10, 12, 13, 14)	64.29	100	75.57	68.93	0.666

Table 5 FRSE + PSO + SVM; average accuracy with fivefold CV = 85.38 %

Dataset	Actual features	Reduced best features	% Reductions	Training accuracy (%)	Fivefold cross-validation	66 % training and rest testing	Fitness value
IRIS	5	5	100	96.67	96.67	96.07	0.98
Tic-Tac-Toe	10	9 (2, 3, 4, 5, 6, 7, 8, 9, 10)	90	77.66	76.4	95.46	0.827
Hepatitis	20	12 (1, 2, 3, 4, 6, 9, 10, 11, 14, 15, 19, 20)	60	87.09	81.94	86.79	0.734
Pima diabetis	9	9	100	77.48	77.21	79.31	0.896
Wisconsin breast cancer	10	8 (1, 2, 4, 5, 6, 7, 9, 10)	80	96.71	96.57	96.22	0.881
Heart-C	14	9 (1, 3, 5, 7, 8, 10, 12, 13, 14)	64.28	84.16	83.5	81.56	0.729

Table 6 FRSE + PSO + random forest; average accuracy with fivefold CV = 85.08 %

Dataset	Actual features	Reduced best features	% Reductions	Training accuracy (%)	Fivefold cross-validation	66 % training and rest testing	Fitness value
IRIS	5	5	100	99.33	95.33	96.07	0.98
Tic-Tac-Toe	10	9 (2, 3, 4, 5, 6, 7, 8, 9, 10)	90	99.38	81.63	79.15	0.846
Hepatitis	20	12 (1, 2, 3, 4, 6, 9, 10, 11, 14, 15, 19, 20)	60	98.7	81.94	79.3	0.697
Pima diabetis	9	9	100	98.31	75.26	77.78	0.889
Wisconsin breast cancer	10	8 (1, 2, 4, 5, 6, 7, 9, 10)	90	99.78	97.45	97.19	0.936
Heart-C	14	9 (1, 3, 5, 7, 8, 10, 12, 13, 14)	64.28	98.34	78.87	78.64	0.715

Table 7 SMOTE + FRNN; Average accuracy with fivefold CV = 82.8 %

Dataset	Actual features	Reduced best features	% Reductions	Training accuracy (%)	Fivefold cross-validation	66 % training and rest testing	Fitness value
IRIS	5	5	100	100	92.5	88.3	0.942
Tic-Tac-Toe	10	10	100	100	64.5	63.8	0.819
Hepatitis	20	20	100	99.5	82.9	76.6	0.883
Pima diabetis	9	9	100	100	77.2	77.3	0.887
Wisconsin breast cancer	10	10	100	100	98.2	97.8	0.989
Heart-C	14	9 (1, 3, 5, 7, 8, 10, 12, 13, 14)	64.28	100	81.5	82	0.731

Table 8 SMOTE + SVM; average accuracy with fivefold CV = 88.34 %

Dataset	Actual features	Reduced best features	% Reductions	Training accuracy (%)	Fivefold cross-validation	66 % training and rest testing	Fitness value
IRIS	5	5	100	97.5	97.5	97.1	0.985
Tic-Tac-Toe	10	10	100	87.9	87.9	90.9	0.955
Hepatitis	20	20	100	90.9	85.6	79.68	0.898
Pima diabetis	9	9	100	75.1	74.6	74.8	0.874
Wisconsin breast cancer	10	10	100	97.55	96.92	95.94	0.98
Heart-C	14	9 (1, 3, 5, 7, 8, 10, 12, 13, 14)	64.28	89.6	87.52	84.67	0.745

Table 9 SMOTE + random forest; average accuracy with fivefold CV = 90.04 %

Dataset	Actual features	Reduced best features	% Reductions	Training accuracy (%)	fivefold cross-validation	66 % training and rest testing	Fitness value
IRIS	5	5	100	100	95.5	94.12	0.97
Tic-Tac-Toe	10	10	100	99.85	94.96	90.66	0.953
Hepatitis	20	20	100	98.39	88.76	73.44	0.867
Pima diabetis	9	9	100	98.94	77.89	75	0.875
Wisconsin breast cancer	10	10	100	99.79	97.87	96.88	0.985
Heart-C	14	9 (1, 3, 5, 7, 8, 10, 12, 13, 14)	64.28	99.1	85.26	86	0.751

Table 10 Comparison of the classification accuracy

Dataset	SMOTE + RF [ours]	EIS + RFS [68]	S-AIRS [69]	FR + SVM [70]	RSES [71]	HEA [72]	Fuzzy [73]
IRIS	95.5	96.00	–	–	–	**96.60**	–
Tic-Tac-Toe	94.96	78.29	–	–	**100**	–	73.64
Hepatitis	**88.76**	82.58	–	56.14	–	86.12	–
Pima diabetis	77.89	74.80	–	–	–	**78.26**	75.03
Wisconsin breast cancer	97.87	96.42	96.91	**98.71**	94.4	76.2	98.05
Heart-C	**85.26**	55.16	–	83.33	–	84.07	–

The significance of bold indicates highest accuracy for that algorithm over all others, for that dataset

Table 11 Summary of z test between any two of the compared classification methods

	FRSE + ACO + FRKNN	FRSE + PSO + FRNN	SMOTE + FRNN	SMOTE + SVM	FRSE + PSO + RF	FRSE + PSO + SVM	SMOTE + RF
FRSE + ACO + FRKNN	–						
FRSE + PSO + FRNN	0.473	–					
SMOTE + FRNN	0.253	0.523	–				
SMOTE + SVM	0.07	0.09	0.35	–			
FRSE + PSO + RF	0.148	0.271	0.708	0.520	–		
FRSE + PSO + SVM	0.139	0.249	0.672	0.559	0.954	–	
SMOTE + RF	0.043	0.039	0.207	0.713	0.305	0.336	–

In order to provide the efficacy of the proposed approaches, a comparison with the other existing literature is provided in Table 10.

Further, the consistency attainment of the algorithms used with parametric and nonparametric tests are provided in Tables 11, 12, 13 and 14 with their P value at 95 % confidence level (0.05 significance level).

It can also be observed that the feature reduction is done from 20 to 100 %, but the important objective of the feature selection algorithm is to select as less as possible while ensuring classification accuracy. In that point, even though the SMOTE-based classifiers could achieve more classification accuracy, the features could not be reduced, hence taking more time to build the model, in comparison with FRSE + PSO-based classifiers.

From Table 10, we can conclude that even though not a single algorithm performs better in all six datasets considered in this study, our SMOTE + RF performs best in Hepatitis and Heart-c dataset, close to the best in other datasets.

From Tables 11 and 12 using parametric statistical test of significance (z test and t test), the following observations are obtained.

- With Z Test
  - (i) SMOTE + RF is the only one among all the compared classification algorithms that outperform FRSE + ACO + FRKNN and FRSE + PSO + FRNN both.

Table 12 Summary of Student *t* test between any two of the compared classification methods

	FRSE + ACO + FRKNN	FRSE + PSO + FRNN	SMOTE + FRNN	SMOTE + SVM	FRSE + PSO + RF	FRSE + PSO + SVM	SMOTE + RF
FRSE + ACO + FRKNN	–						
FRSE + PSO + FRNN	0.49	–					
SMOTE + FRNN	0.279	0.537	–				
SMOTE + SVM	0.1	0.121	0.372	–			
FRSE + PSO + RF	0.179	0.297	0.716	0.534	–		
FRSE + PSO + SVM	0.17	0.276	0.681	0.572	0.955	–	
SMOTE + RF	0.071	0.066	0.235	0.721	0.329	0.359	–

Table 13 Summary of Sign test between any two of the compared classification methods

	FRSE + ACO + FRKNN	FRSE + PSO + FRNN	SMOTE + FRNN	SMOTE + SVM	FRSE + PSO + RF	FRSE + PSO + SVM	SMOTE + RF
FRSE + ACO + FRKNN	–						
FRSE + PSO + FRNN	0.25	–					
SMOTE + FRNN	0.061	0.219	–				
SMOTE + SVM	0.031	0.031	0.688	–			
FRSE + PSO + RF	0.031	0.031	0.688	0.688	–		
FRSE + PSO + SVM	0.031	0.031	0.688	0.219	1.0	–	
SMOTE + RF	0.031	0.031	0.219	0.688	0.031	0.219	–

Table 14 Summary of Wilcoxon's signed-rank test between any two of the compared classification methods

	FRSE + ACO + FRKNN	FRSE + PSO + FRNN	SMOTE + FRNN	SMOTE + SVM	FRSE + PSO + RF	FRSE + PSO + SVM	SMOTE + RF
FRSE + ACO + FRKNN	–						
FRSE + PSO + FRNN	0.181	–					
SMOTE + FRNN	0.031	0.063	–				
SMOTE + SVM	0.031	0.031	0.156	–			
FRSE + PSO + RF	0.031	0.031	1.0	0.156	–		
FRSE + PSO + SVM	0.031	0.031	0.313	0.156	0.787	–	
SMOTE + RF	0.031	0.031	0.063	0.313	0.031	0.094	–

(ii) No sufficient evidence is found in order to conclude whether anyone among SMOTE + FRNN, SMOTE + SVM, FRSE + PSO + RF, FRSE + PSO + SVM and SMOTE + RF is statistically significant than the other.

- With *t* test

(i) The observation is such that no algorithm is found to outperform the other.

As parametric statistical test of significance works with assumptions about the shape of the distribution (i.e., assume a normal distribution) in the underlying population and about the form or parameters (i.e., means and standard deviations) of the assumed distribution, sometimes they may provide misleading results. Hence, we propose to use some nonparametric methods such as sign test and Wilcoxon's signed-rank test for obtaining the consistency in operation of classifiers irrespective of dataset types. The results obtained with sign test and Wilcoxon's signed-rank test on classification accuracy with 5 % significant level and 95 % confidence levels are highlighted in Tables 13 and 14, respectively, with following observations:

- Sign test
  - (i) SMOTE + SVM, FRSE + PSO + RF, FRSE + PSO + SVM and SMOTE + RF outperform FRSE + ACO + FRKNN and FRSE + PSO + FRNN.
  - (ii) SMOTE + RF outperforms FRSE + PSO + RF, FRSE + ACO + FRKNN and FRSE + PSO + FRNN; but does not statistically significantly better than SMOTE + FRNN, SMOTE + SVM and FRSE + PSO + SVM.
  - (iii) No sufficient evidence supports that whether SMOTE + RF, FRSE + PSO + SVM, SMOTE + SVM and SMOTE + FRNN outperform one another in anyway.

As sign test cannot distinguish between 0.01 and 10.0, consider both the same, we propose to explore the Wilcoxon signed-rank test.

- Wilcoxon's signed-rank test
  - (i) SMOTE + FRNN, SMOTE + SVM, FRSE + PSO + RF, FRSE + PSO + SVM and SMOTE + RF outperform FRSE + ACO + FRKNN. However, SMOTE + FRNN is not statistically significant than FRSE + PSO + FRNN.
  - (ii) SMOTE + RF is statistically superior to FRSE + PSO + RF, FRSE + ACO + FRKNN and FRSE + PSO + FRNN.
  - (iii) No sufficient evidence is obtained in order to make a decision whether SMOTE + RF, SMOTE + FRNN, SMOTE + SVM and FRSE + PSO + SVM are having more statistical significance than the other.

From all the above statistical test, it is apparent that even though the average classification accuracy of SMOTE + RF is more than that of SMOTE + FRNN, SMOTE + SVM and FRSE + PSO + SVM, still it is not found to be statistically significant in comparison with the other. Further, it is worth noting here that average classification accuracy difference of 4.92 % (between SMOTE + RF and FRSE + PSO + RF) may be statistically significant, whereas the difference of 7.2 % (between SMOTE + RF and SMOTE + FRNN) is not statistically significant.

This may be understood that while average classification accuracy provides knowledge about the average performance of the algorithm over many used dataset under this investigation, the statistical test, as discussed above, provides an idea about the consistency of an algorithm over another on each test dataset. Hence, even though the accuracy difference is small among the algorithms, one may be considered significantly better than the other.

Finally, we perform statistically significant test for our best algorithm SMOTE + RF with highest average accuracy of 90.04 % with that of EIS + RFS [68] with average accuracy of 80.54 %, the following observations are made:

- P value with z test = 0.174; with t test = 0.204
- P value with sign test = 0.219; with Wilcoxon's signed-rank test = 0.063
- From both types of statistical test, it is concluded that even though accuracy difference is around 10 % between them, still SMOTE + RF is not statistically significant than EIS + RFS.

7 Conclusion

Feature selection plays an important role in classification data mining. Feature evaluation functions used to compute the quality of features are a key issue in feature selection. In the rough set theory dependency and fuzzy dependency has been successfully used to evaluate features. However, we find these function are not robust. In practice, data are usually corrupted by noise. So design of robust models of rough sets using bio-inspired algorithms and SMOTE are discussed. Further, we used SVM and RF to build efficient classifiers using PSO and fuzzy rough set feature reduction techniques. Based on the analysis, many hybrid algorithms based on FRSE are proposed to combine the strengths of FR-KNN, SVM, RF and the PSO classifier. Instance-based supervised feature selection method SMOTE is also used along with SVM, FRKNN and RF in order to understand the efficacy of the classification accuracy of the proposed system. Extensive experiments conducted on these methodologies to validate the classification data mining in the real-life datasets. Even though FRSE + PSO is considered to provide faster, stable, effective average classification data mining, still the SMOTE-based SVM classifier steals the show with better accuracy and fitness value.

We conducted z test, t test, sign test and Wilcoxon's signed-rank test to evaluate the performance comparison various classification data mining methods. We observed that the Wilcoxon signed-rank test is better than all other kinds of statistical test. We also found that there is no sufficient evidence to support the SMOTE + RF statistically significant in comparison with other hybrid evolutionary algorithms discussed even though it has the highest average accuracy among all. This may be envisaged from here that different classification algorithms may be able to suits for a dataset, whereas the other does not.

References

1. Mitra S, Pal SK, Mitra P (2002) Data mining in soft computing framework: a survey. IEEE Trans Neural Networks 13:3–14
2. Zhong N et al (2001) Using rough sets with heuristics for feature selection. J Intell Inf Syst 16:199–214
3. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97:273–324
4. Whitney A (1971) A direct method of nonparametric measurement selection. IEEE Trans Comput 9(C-20):1100–1103
5. Marill T, Green D (1963) On the effectiveness of receptors in recognition systems. IEEE Trans Inf Theory 9(1):11–17
6. Mohemmed A, Zhang M, Johnston M (2009) Particle swarm optimization based adaboost for face detection. In: IEEE congress on evolutionary computation (CEC'09), pp 2494–2501
7. Neshatian K, Zhang M (2009) Dimensionality reduction in face detection: a genetic programming approach. In: 24th international conference image and vision computing New Zealand (IVCNZ'09), pp 391–396
8. Unler A, Murat A (2010) A discrete particle swarm optimization method for feature selection in binary classification problems. Eur J Oper Res 206(3):528–539
9. Yang CS, Chuang LY, Ke CH, Yang CH (2008) Boolean binary particle swarm optimization for feature selection. In: IEEE congress on evolutionary computation (CEC'08), pp 2093–2098
10. Yuan H, Tseng SS, Gangshan W (1999) A two-phase feature selection method using both filter and wrapper. In: IEEE international conference on systems, man, and cybernetics (SMC'99), vol 2, pp 132–136
11. Kennedy J, Spears W (1998) Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In: IEEE congress on evolutionary computation (CEC'98), pp 78–83
12. Qablan T, Al-Radaidehl QA, Abu Shuqeir S (2012) A reduct computation approach based on ant colony optimization. Basic Sci Eng 21(1):29–40
13. Chen Y, Miao D, Wang R (2010) A rough set approach to feature selection based on ant colony optimization. Pattern Recogn Lett 31:226–233
14. Wang J, Xu M, Wang H, Zhang J (2007) Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. In: International conference on signal processing proceedings, 4129201
15. Chandana S, Leung H, Trpkov K (2009) Staging of prostate cancer using automatic feature selection, sampling and Dempster–Shafer fusion. Cancer Inform 7:57–73
16. Pawlak Z (1982) Rough sets. Int J Comput Inform Sci 11(5):341–356
17. Mi JS, Wu WZ, Zhang WX (2004) Approaches to knowledge reduction based on variable precision rough set model. Inform Sci 159(3–4):255–272
18. Saha M, Sil J, Sengupta N (2013) Genetic algorithm and fuzzy-rough based dimensionality reduction applied on real valued dataset. Int J Comput Inf Syst Ind Manag Appl 5:462–471
19. Lingras P, Jensen R (2007) Survey of rough and fuzzy hybridization. In: Proceedings of the 16th international conference fuzzy systems, pp 125–130
20. Jensen R, Shen Q (2009) New approaches to fuzzy-rough feature selection. IEEE Trans Fuzzy Syst 17(4):824–838
21. Pedrycz W, Skowron A (2001) Rough sets and fuzzy sets in data mining. In: Zytkow W, Klosgen W (eds) Handbook of knowledge discovery & data mining. Oxford University Press
22. Keller JM, Gray MR, Givens JA (1985) A fuzzy K-nearest neighbor algorithm. IEEE Trans Syst Man Cybernet 15(4):580–585
23. Sarkar M (2007) Fuzzy-rough nearest neighbors algorithm. Fuzzy Sets Syst 158:2123–2152
24. Jones DT (1999) Protein secondary structure prediction based on position specific scoring matrices. J Mol Biol 292:195–202
25. Panda M, Patra MR (2009) Mining knowledge from network intrusion data using data mining techniques. In: Dehuri SN et al (eds) Knowledge mining using intelligent agents. World Scientific, Singapore
26. Panda M, Patra MR (2009) Ensemble voting system for anomaly based network intrusion detection. Int J Recent Trends Eng 2(5):8–13
27. Dehuri SN, Nanda BK, Cho S-B (2009) A hybrid APSO-aided learnable Bayesian classifier. In: Proceedings of Indian international conference on artificial intelligence (IICAI), pp 695–706
28. Xue B, Zhang M, Browne WN (2012) Multi-objective particle swarm optimisation (PSO) for feature selection, GECCO'12, July 7–11, 2012. ACM Press, Philadelphia, Pennsylvania, USA, pp 81–88
29. Grosan C, Abraham A, Chis M (2006) Swarm intelligence in data mining. In: Abraham A et al (eds) Studies in computational intelligence series, vol 34. Springer, Berlin
30. Abraham A, Guo H, Liu H (2006) Swarm intelligence: foundations, perspectives and applications. In: Abraham A et al (eds) Swarm intelligence: foundations, perspectives and applications, studies in computational intelligence (SCI), vol 26. Springer, Germany, pp 3–25
31. Suguna N, Thanushkodi K (2010) A novel rough set reduct algorithm for medical domain based on bee colony optimization. J Comput 2(6):49–54
32. Ding S, Chen J, Xu X, Li J (2011) Rough neural networks: a review. J Comput Inf Syst 7(7):2338–2346
33. Fazayeli F, Wang L, Mandziuk J (2008) Feature selection based on the rough set theory and EM clustering algorithm. In: Proceedings of the 6th international conference on rough sets and current trends in computing, Springer, pp 272–282
34. Wang KJ, Adrian AM (2013) Breast cancer classification using hybrid synthetic minority over-sampling technique and artificial immune recognition system algorithm. Int J Comput Sci Electron Eng (IJCSEE) 1(3):408–412
35. Wanga X, Yanga J, Jensenb R, Liua X (2006) Rough set feature selection and rule induction for prediction of malignancy degree in brain glioma. Comput Methods Programs Biomed 83:147–156
36. Derrac J, Cornelis C, Garcıa S, Herrera1 F (2011) A preliminary study on the use of fuzzy rough set based feature selection for improving evolutionary instance selection algorithms. In: Cabestany J, Rojas I, Joya G (eds) IWANN 2011, part I, LNCS 6691, pp 174–182
37. Ganivada A, Raya SS, Pal SK (2013) Fuzzy rough sets, and a granular neural network for unsupervised feature selection. Neural Netw 48:91–108
38. Sabzevari R, Montazer GA (2008) An intelligent data mining approach using neuro-rough hybridization to discover hidden knowledge from information systems. J Inf Sci Eng 24:1111–1126
39. Sangeetha R, Kalpana B (2013) Enhanced fuzzy roughset based feature selection strategy using differential evolution. Int J Comput Sci Appl (TIJCSA) 2(06):13–20
40. Hu X, Shi Y, Eberhart RC (2004) Recent advances in particle swarm. In: Proceedings of congress on evolutionary computation (CEC), Portland, Oregon, pp 90–97
41. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, vol 4. Perth, Australia, IEEE Service Center, Piscataway, NJ, pp 1942–1948
42. Kennedy J (1997) Minds and cultures: particle swarm implications. Socially intelligent agents. Papers from the 1997 AAAI fall

symposium. Technical report FS-97-02. AAAI Press, Menlo Park, CA, pp 67–72
43. Kennedy J (1998) The behavior of particles. In: Proceedings of 7th annual conference on evolutionary programming. San Diego, USA
44. Kennedy J (1997) The particle swarm: social adaptation of knowledge. In: Proceedings of IEEE international conference on evolutionary computation. Indianapolis, Indiana, IEEE Service Center, Piscataway, NJ, pp 303–308
45. Kennedy J (1997) Thinking is social: experiments with the adaptive culture model. J Confl Resolut 42:56–76
46. Pomeroy P (2003) An introduction to particle swarm optimization. http://www.adaptiveview.com/articles/ipsop1.html
47. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. Theoret Comput Sci 344(2–3):243–278
48. Dorigo M, Di Caro G, Gambardella LM (1999) Ant algorithms for discrete optimization. Artif Life 5(2):137–172
49. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66
50. Dorigo M, Bonaneau E, Theraulaz G (2000) Ant algorithms and stigmergy. Future Gener Comput Syst 16:851–871
51. Toksari MD (2006) Ant colony optimization for finding the global minimum. Appl Math Comput 176(1):308–316
52. Chowla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
53. Chen D, Zhang L, Zhao S, Hu Q, Zhu P (2012) A novel algorithm for finding reducts with fuzzy rough sets. IEEE Trans Fuzzy Syst 20(2):385–389
54. Bhatt RB, Gopal M (2005) On fuzzy-rough sets approach to feature selection. Pattern Recogn Lett 26(7):965–975
55. Thangavel K, Pethalakshmi A, Jaganathan P (2006) A comparative analysis of feature selection algorithms based on rough set theory. Int J Soft Comput 1(4):288–294
56. Wang X, Han D, Han C (2012) Fuzzy-rough set based attribute reduction with a simple fuzzification method. In: IEEE control and decision conference (CCDC), pp 3793–3797
57. Keller JM, Gray MR, Givens JA (1985) A fuzzy K-nearest neighbor algorithm. IEEE Trans Syst Man Cybernet 15(4):580–585
58. Sarkar M (2007) Fuzzy-rough nearest neighbors algorithm. Fuzzy Sets Syst 158:2123–2152
59. Wang X, Yang J, Teng X, Peng N (2005) Fuzzy-rough set based nearest neighbor clustering classification algorithm. Lect Notes Comput Sci 3613:370–373
60. Platt J (1999) SVM by sequential minimal optimization (SMO). ACM Press, USA
61. Chen C, Liaw A, Breiman L (2004) Using random forest to learn imbalanced data, July 2004
62. Liang G, Zhang C (2011) Empirical study of bagging predictors on medical data. In: Proceedings of the 9-th Australasian data mining conference (AusDM'11), vol 121, data mining and analytics. Ballarat, Australia, CRPIT, pp 31–40
63. Trawiński B, Smętek M, Telec Z, Lasota T (2012) Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. Int J Appl Math Comput Sci 22(4):867–881
64. Howell DC (2013) Statistical methods for psychology, 8th edn. Cengage Wadsworth, Belmont, CA
65. Smucker MD, Allan J, Carterette B (2007) A comparison of statistical significance tests for information retrieval evaluation, CIKM'07, November 6–8, 2007, ACM Press, Lisboa, Portugal, pp 623–632
66. Blake CL, Merz CJ (1998) UCI repository of machine learning databases. http://www.ics.uci.edu/mlearn/MLRepository.html
67. Witten IH, Frank E (2005) Data mining-practical machine learning tools and techniques, 2nd edn. Morgan Kauffman Publishers, Elsevier, Amsterdam
68. Derrac J, Cornelis C, Garcia S, Herrera F (2011) A preliminary study on the use of fuzzy rough set based feature selection for improving evolutionary IS algorithms. In: Cabestany J, Rojas I, Jaya G (eds) IWANN 2011, part-1, LNCS 6691, pp 174–182
69. Wang KJ, Adrian AM (2013) Breast cancer classification using hybrid synthetic minority oversampling technique and artificial immune recognition system algorithm. Int J Comput Sci Electron Eng 1(3):408–412
70. Hu Q, Yu D, Xie Z (2005) A hybrid attribute reduction for classification based on a fuzzy roughest technique. Fifth SIAM International conference on data mining, pp 195–204
71. Wang X, Yang J, Tang X, Xia W, Jensen R (2007) Feature selection based on roughest and particle swarm optimization. Pattern Recogn Lett 28:459–471
72. Tan KC, Teoh EJ, Yu Q, Goh KC (2009) A hybrid evolutionary algorithm for attribute selection in data mining. Exp Syst Appl 36:8616–8630
73. Homlich M, Ramdani M (2012) Data classification by fuzzy ant-miner. Int J Comput Stud 19(3–3):201–206