JCSE Journal of Computing Science and Engineering

Special Issue

Journal of Computing Science and Engineering,
Vol. 11, No. 4, December 2017, pp. 130-141

Feature Selection Based on Bi-objective Differential Evolution

Sunanda Das
Neotia Institute of Technology, Management and Science, Diamond Harbour, West Bengal, India
Indian Institute of Engineering Science and Technology, Howrah, West Bengal, India
das.sunanda2012@gmail.com

Chi-Chang Chang
School of Medical Informatics, Chung-Shan Medical University, Taichung, Taiwan;
Biomedical Industry Research Centre and IT Office, Chung-Shan Medical University Hospital, Taichung, Taiwan
changintw@gmail.com

Asit Kumar Das* and Arka Ghosh
Indian Institute of Engineering Science and Technology, Howrah, West Bengal, India
akdas@cs.iiests.ac.in, 9007900477a@gmail.com

Abstract

Feature selection is one of the most challenging problems of pattern recognition and data mining. In this paper, a feature selection algorithm based on an improved version of binary differential evolution is proposed. The method simultaneously optimizes two feature selection criteria, namely, set approximation accuracy of rough set theory and relational algebra based derived score, in order to select the most relevant feature subset from an entire feature set. Superiority of the proposed method over other state-of-the-art methods is confirmed by experimental results, which is conducted over seven publicly available benchmark datasets of different characteristics such as a low number of objects with a high number of features, and a high number of objects with a low number of features.

Category: Databases / Data Mining

Keywords: Feature selection; Rough set theory; Differential evolution; Classification

I. INTRODUCTION

Data mining is the kind of knowledge discovery process for searching meaningful patterns or interesting information from the real life voluminous dataset with a huge number of redundant, noisy and inconsistent data. Tremendous growth in technology in the past few decades results in a mammoth amount of data which is to be analyzed intelligently to extract fruitful knowledge from it. Feature selection is an important part of dimension reduction which is mainly done in two approaches: filter approach [1], which is independent of underlying classifier strategy, and wrapper approach [2], which entirely depends upon the underlying classifier. However, some hybrid feature selection [2] and embedded feature selection [3] techniques are noticeable in the literature. To overcome the curse of dimensionality [4], feature selection is the most important part of data mining. Objective of feature

Open Access http://dx.doi.org/10.5626/JCSE.2017.11.4.130 http://jcse.kiise.org

Received 25 May 2017; **Accepted** 25 November 2017

*Corresponding Author

pISSN: 1976-4677 eISSN: 2093-8020

selection process is to select the most informative and compact subset from an entire feature set without losing any important information required for subsequent data mining process. Small sample size and external noise are often treated as obstacles for achieving good mining result from the data. The stability which indicates the insensitivity of feature selection over external perturbation in the data set or induced noise in the data set plays an important role in the knowledge discovery process.

Inherently, feature selection is a combinatorial optimization problem [5] which searches an optimal feature subset of a pool of 2^N competing candidate subset in the data set of N features. Optimum feature selection can be done by means of several unsupervised measures like feature similarity measure (FSFS) [6], unsupervised discriminative feature selection (UDFS) [7], multi-cluster feature selection [8], etc. Popular methods for feature selection include the fuzzy-rough feature selection [9-11], rough set based feature selection [12, 13] and probabilistic feature selection [14]. Independent of learning mechanism to search for an optimal feature subset is the primary objective of feature selection algorithm. As optimal feature subset selection from a pool of 2^N competing candidate solution is an NP-complete problem, several evolutionary search methods [15-20] are used for the purpose.

At present situation, differential evolution (DE) [21, 22] holds a very prominent role in evolutionary optimizer space. Various measures have been taken to improve the performance of DE for scalable optimization problems. Some DE-variants incorporate parameter adaptation based on success-history (like SaDE [23], JADE [24]), new mutation and crossover policies (like Pro-DE [25], MDE-pBX [26]), and combining diverse offspring generation policies (like CoDE [27], EPSDE [28]). For large-scale optimization problems (greater than 500 dimensions) DE has been adopted by methods including comprehensibly all the above three algorithmic philosophies. Owing to its inherent simplicity, DE was used as the base optimizer in Yang et al.'s first work [29] on random grouping based CCEAs. Zamuda et al. [30] enriched DE by log-normal self-adaptation of its control parameters and by using cooperative co-evolution as a decomposition mechanism based on each dimension. The cooperative micro-DE developed by Parsopoulos [31] for large scale global optimization. Zhao et al. [32] hybridized the self-adaptive DE with MTS for large scale optimization. Some other approaches of improving DE for high-dimensional function optimization can be found in [33-35].

In the paper, multi-objective search for optimal feature subset selection using bi-objective differential evolution is proposed. In the method, feature selector uses two objective functions, one based on set approximation accuracy of rough set theory [36] and the other based on division operation of relational algebra. For selection step in proposed modified binary differential evolution, a dominance based [37] approach is used in the paper. Each bi-objective differential evolution based feature selector (DEFS) produces a non-dominated [37] feature subset from which the final feature subset is chosen. Some well-known datasets with typical characteristics like small sample size with large feature size, and large sample size with small feature size are collected from publicly available UCI ML repository [38] to show the effectiveness of the proposed method. Performance comparison of proposed method with some state of the art feature selection methods like feature similarity measure or FSFS [4], Laplacian score for feature selection or LSFS [39], multi-cluster feature selection or MCFS [40], dense subgraph finding with feature clustering or DSFFC [41], correlation based feature selection or CFS [42], CON [42], particle swarm optimization search or PSO search [43], genetic search [44] and improved differential evolution based unsupervised feature selection or IMoDEFS [45] is provided in the paper. Classification accuracies on reduced datasets generated by CFS, CON, PSO search and genetic search are measured by various classifiers available at WEKA [46] tool while the same is collected from paper [41] for feature selection methods like FSFS, LSFS, MCFS and DSFFC. The experimental result shows the effectiveness of the proposed method.

The remaining part of the paper is organized as follows: some preliminary concepts of DE used in proposed method are described in Section II. Subsequently, detailed feature selection methodologies based on DE, experimental results and comparative performance analysis are provided in Sections II and III, respectively. Finally, Section IV holds the conclusion and final comments.

II. THE DIFFERENTIAL EVOLUTION PRELIMINARIES

Initialization, mutation, recombination and selection are four basic steps of standard differential evolution or DE algorithm. The generations are executed sequentially until some termination criteria are satisfied (proposed work used exhaustion of maximum iteration as termination criteria).

A. Initialization

In DE, initialization of *Np* real-valued vectors whose components represent *D* parameters of the optimization problem is the first step of search for global optimum in *D* directional search space. A generalized notation used to identify the i^{th} solution (real parameter vector) of the present generation *G* can be shown as:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G} \dots x_{D,i,G}].$$

Given the decision parameter bounds,

$\vec{X}_{max} = [x_{1,max}, x_{2,max} \dots x_{D,max}]$, and $\vec{X}_{min} = [x_{1,min}, x_{2,min} \dots x_{D,min}]$, the j^{th} dimension of i^{th} individual is initialized by Eq. (1).

$$x_{i,j} = x_{j,min} + rand_{i,j} \times (x_{j,max} - x_{j,min}) \qquad (1)$$

where $rand_{i,j}$ is a uniformly distributed random number lying in the range [0, 1] and it is newly generated for each ordered pair (i, j).

B. Mutation

Invited target vector is the term for any population member (say, i) of the current population, mutated with scaled difference vectors $(\vec{\boldsymbol{X}}_{r1,G} - \vec{\boldsymbol{X}}_{r2,G})$ to produce a new mutant or *donor vector*. The indices r_1 and r_2, sampled from $\{1,2,\dots,Np\}$, are different from the current index i. A scaling factor F, usually lying in the range [0.4, 2], scales the difference vector(s). Two well-known DE mutation strategies are listed in Eq. (2a) and (2b).

$$DE / rand / 1: : \; \vec{V}_{i,G} = \vec{X}_{r_1,G} + F(\vec{X}_{r_2,G} - \vec{X}_{r_3,G}) \qquad (2a)$$

$$DE / best / 1: : \; \vec{V}_{i,G} = \vec{X}_{best,G} + F(\vec{X}_{r_2,G} - \vec{X}_{r_3,G}) \qquad (2b)$$

where r_1, r_2, and r_3 are mutually exclusive indices that are randomly selected from $\{1,2,\dots,Np\}$.

C. Recombination

In DE, the crossover step aims to combine the individual components of the parent and the mutant vector into a single offspring commonly known as *trial vector* $\vec{\boldsymbol{U}}_{i,G} = [\boldsymbol{u}_{1,i,G}, \boldsymbol{u}_{2,i,G}, \dots \boldsymbol{u}_{D,i,G}]$. DE primarily employs either *exponential* (two-point modulo) or *binomial* (uniform) crossover strategies. Generally, binomial crossover is preferred since it does away with the inherent representational bias in n-point crossover by simulating D random trials. Moreover, a recent work [35] attributing to the sensitivity of crossover to population size has reported the exponential variant to be more prone as compared to its binomial counterpart. Owing to the aforesaid observations, here we employ binomial crossover to form the trial vector.

In order to implement the binomial crossover, the control parameter *Crossover rate* (*Cr*) is set to a fixed value lying in the range [0, 1] and then D independent random numbers (between 0 and 1) are sampled uniformly and compared with Cr to decide which component is to be included in the trial vector. The components of the trial vector are computed using Eq. (3).

$$\boldsymbol{u}_{j,i,G} = \begin{cases} \boldsymbol{v}_{j,i,G}, \; if \; rand_{i,j} < CR \; and \; j = j_r, \\ \quad x_{j,i,G} \; otherwise \end{cases} \qquad (3)$$

where, j_r is a randomly chosen index from $\{1, 2\dots D\}$ and it ensures that at least one component from the mutant vector is present in the offspring produced.

D. Selection

Finally, a selection process is performed through a one-to-one competition between the parent and the offspring to maintain a constant population size. The selection process can be defined using Eq. (4).

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_i, \; if \; f(\vec{U}_i) \leq f(\vec{X}_i), \\ X_i, \; otherwise \end{cases} \qquad (4)$$

where, $f(.)$ is the objective function to be minimized.

III. DE BASED FEATURE SELECTION

In the paper, the concepts of DE are used to select the most important features of the dataset that can fully characterize the overall system.

A. Population Generation

As DE is a population based stochastic search algorithm, the initial population for proposed method is created at random. The dimension of population set is $P \times N$ where P is the defined population size and N is the number of features in the dataset.

B. Encoded Solution Representation

Binary string representation of potential solution is opted in this work. The length of the string is the same as the total number of features present in the dataset. Each encoded population member consists only of two values, where '1' means that index feature is present in current subset and '0' means that index feature is absent in current subset. So, a chromosome C_l of length K is represented as {1 0 1 1 . . . 1 0 }.

C. Mutation Strategy

While traditional 'vector difference based' mutation strategies are perfect for real values, this scheme is not suitable for the proposed work where population of binary strings is considered. To overcome this problem the proposed method uses hamming distance [47] and binary bitwise operation based mutation scheme. In this scheme, for the difference vector two distinct population members are selected from current population at unbiased manner. Then hamming distance is calculated among these two population members and a scaling parameter $F \in [0.5, 2]$ is multiplied with it and its ceiling value is considered as the difference factor for proposed mutation scheme. Now from base vector a random string is generated, which has the same hamming distance of difference factor with the base vector.

D. Recombination Strategy

Here, binomial recombination [21] strategy is used where each dimension of newly generated trial vector comes from the corresponding dimension of either base vector or mutant vector based on the control parameter *Cr*. The value of *Cr* is chosen either 0 or 1 at random and unique for each dimension of trial vector and finally the standard binomial recombination is opted.

E. Set Dominance Relationship, Non-dominated Set and Pareto Front

Dominance Relationship: A solution X_1 dominates other solution X_2 and it is written as $X_1 \geq X_2$ if both of the following conditions are satisfied:

(1) The solution X_1 is no worse than X_2 with respect to all objective functions;
(2) The solution X_1 is strictly better than X_2 with respect to at least one objective function.

Mathematically, $X_1 \geq X_2$ iff $f_i(X_1) \geq f_i(X_2)$ for all *M* objective functions $f_1, f_2, \ldots, f_M$ and there exist at least one objective function f_j (for $j = 1, 2, \ldots, M$) for which $f_j(X_1) > f_j(X_2)$. Dominance relation is not reflexive and symmetric but transitive. Non-dominated set is the set among a set of solutions *R* that are not dominated by any member of the set *R*.

Strong dominance solution is the solution X_1 which strongly dominates another solution X_2, i.e., $X_1 > X_2$ if solution X_1 is strictly better than solution X_2 in all *M* objective functions.

Algorithm 1 of proposed DE based feature selection method or DEFS is given below.

Algorithm 1 DEFS (U,N_P,F_V,CR_V,D,I_M)

```
/* U is the dataset used for feature selection, N_P is population size, F_V is the vector of [0.5 2] for scaling factor F, CR_V is the vector of [1 0] for recombination factor CR, D is the dimension of the dataset used and I_M is maximum permitted iteration */
Begin
for $i = 1.. N_P$ do
    for $j = 1.. D$ do
        $x_i^j \leftarrow rand[0,1]$
    end for
end for
for $g=1..I_M$ do
    for $i = 1.. N_P$ do
        Randomly generate two integers $r_1, r_2 (\mp i) \in [1, N_P]$
        $F \leftarrow F_V(rand_i(length(F_V)))$
        $v_{i,g} \leftarrow x_{i,g} \otimes \lceil F.hamming\_dist(x_{r_1,g}, x_{r_2,g}) \rceil$
    end for
    for $i = 1.. N_P$ do
        $\boldsymbol{jrand} \leftarrow \lfloor \boldsymbol{D.rand(0,1)} \rfloor$
        $\boldsymbol{CR} \leftarrow \boldsymbol{CR_V}(rand_i(length(CR_V)))$
        for $j = 1.. D$ do
            if $\boldsymbol{rand(0,1) \leq CR \text{ or } j = jrand}$ then
                $u_{i,g}^j \leftarrow v_{i,g}^j$
            else $u_{i,g}^j \leftarrow x_{i,g}^j$
            end if
        end for
    end for
    for $i = 1.. NP$ do
        if $\boldsymbol{f(U, u_{i,g})} > \boldsymbol{f(U, x_{i,g})}$ then
            $x_{i,g+1} \leftarrow u_{i,g}$
        else $x_{i,g+1} \leftarrow x_{i,g}$
        end if
    end for
end for
return (x)
End procedure
Output: Reduct Set
```

F. Objective Functions

The proposed bi-objective feature selection algorithm uses approximation accuracy of rough set theory and relational algebra based score as searching criteria for selecting the most relevant and noise-insensitive feature subset from a large feature set. To calculate lower bound approximation for any target set *X* with respect to an attribute subset *P*, universe of discourse *U* is partitioned into equivalence classes $[x]_p$ using indiscernible relation *IND*(*P*), defined in Eq. (5).

$$\mathrm{IND(P)} = \{(x,y) \in (|U| \times |U|) | \forall a \in P, f_a(x) = f_a(y) \quad (5)$$

where, *f* is the function that represents attribute value of an object. Similarly, equivalence classes $[x]_D$ are formed using Eq. (5) for the subset *D* consisting of decision attributes. Thus, two different partitions *U*/*P* and *U*/*D* of equivalence classes $[x]_P$ and $[x]_D$ are obtained. Now, each class $[x]_D$ in *U*/*D* is considered to be the target set *X*, i.e., $X \in U/D$. The lower approximation $\underline{P}X$ of *X* under *P* is computed using Eq. (6), for all $X \in U/D$ which contains the set of objects that positively belong to the target set *X*. The positive region $POS_P(D)$ is obtained by taking the union of the lower approximations $\underline{P}X$ under *P* for all *X* in *U*/*D*, using Eq. (7), dependency value of *D* on *P* (i.e., $\gamma_P(D)$) is obtained using Equation 8 which signifies how much the decision attribute depends on the condition attribute set *P*. If *D* is fully dependent on *P*, i.e., if $\gamma_P(D) = 1$ then *P* is the sufficient to represent the decision system and remaining attributes are irrelevant in the sense that they do not contribute any significant information about the system.

$$\underline{P}X = \{x|[x]_P \subseteq X\} \tag{6}$$

$$POS_P(D) = \cup_{X \in U/D} \underline{P}X \tag{7}$$

$$\gamma_P(D) = \frac{|POS_P(D)|}{|U|} \tag{8}$$

The relational algebra operation division (÷), defined in Definition 1, has been used to compute the second objective function of proposed method. For each present feature C_i in any population member, $S(C_i)$ is calculated according to Eq. (9.)

$$S(C_i) = |\Pi_{C_i \cup D}(DS) \div \Pi_D(DS)| \tag{9}$$

Final score of any given population member is calculated by taking average of all individual C_i scores, where i=1,2,…,N.

Minimum score of C_i implies that there is a maximum number of objects having feature values similar to C_i, which can uniquely take the decisions. Thus, the population member with minimum average score is of maximum importance and so lower score implies higher possibility of becoming a member of reduct.

Definition 1: Relational algebra operation division (÷) is a binary operation applied on two relations R_1 (P) and R_2 (Q) and produce another relation R (P – Q) where $Q \subset P$, where P, Q are set of attributes of R_1, R_2, respectively. So, R (i.e., R_1 ÷ R_2) contains set of all tuples t such that for any tuple t_1 and t_2 of R_1 and R_2, respectively, following conditions hold.

- $t\,[P\text{-}Q] = t_1\,[P\text{-}Q]$
- $t_1\,[P\text{-}Q] = t_2\,[Q]$

Hence, to find most compact feature set first objective namely set approximation accuracy, is to 'be maximized' and second objective function relational algebra based score is to be minimized; in implementation negative of second objective is considered so now both objective values are to be maximized.

IV. EXPERIMENTAL RESULTS

Extensive experiments are done to evaluate proposed feature selection method using some well-known machine learning datasets from UCI ML repository [38]. Details of datasets are given in Section IV-A and a performance comparison of proposed method with FSFS, LSFS and MCFS. DSFS is opted from [41] for comparison reason, results of IMoDEFS [45] on those datasets are taken from [45]. For LSFS and MCFS, neighborhood size is taken as default (=5). The details of these benchmark datasets, used classifiers and experimental results are given in this section. Parameters for proposed DE based feature selector is given underneath, population size is taken to be 100, maximum number of function evaluation is set to $D \times 10^4$, where D is the problem dimensionality; for comparative study all parameters are taken as respective algorithms in the literature.

Table 1. Characteristics of datasets

Dataset	Features	Samples	Class
Multiple features	649	2000	10
Isolet	617	6238	26
Spambase	57	4601	2
Ionosphere	34	351	2
Wdbc	30	569	2
Sonar	60	208	2
Spectf	44	80	2

A. Used Dataset

The proposed method is validated using seven publicly available datasets, namely, Ionosphere, Sonar, Spambase, WDBC, Isolet, Multiple Features and Spectf, collected from UCI ML repository. A detailed description of all these datasets is given in (Table 1). The datasets are normalized using max-min normalization and all features are sealed into [0, 1] interval. These datasets are specifically chosen for some characteristics like large sample size (isolet, multiple features, spambase,) small feature size (ionosphere, sonar, spectf), two class (spambase, ionosphere, wdbc, sonar, spectf) and multiclass (multiple features, isolet) which are ideal to demonstrate effectiveness of the feature selection algorithm.

B. Classifiers Used

Five classifiers, namely Naïve-Bayes (NB) [48], support vector machine (SVM) [49], K-nearest neighbors (k-NN) [48], AdaBoost_Naive Bayes [50], C4.5 [51], are used to compare and contrast classification accuracy of the different feature selection algorithms.

SVM is used with RBF Kernel, k value of k-NN is set to the square root of sample size of data. Ten-fold cross-validation is used for this classification performance evaluation. All these classifiers are used with WEKA tool.

C. Evaluation Criteria

Five evaluation criteria are used for all these datasets for every classifier, as follows:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Recall = \frac{TP}{P} = \frac{TP}{TP+FN}$$

$$Fall\_out = \frac{FP}{N} = \frac{FP}{FP+TN}$$

$$Specificity = \frac{TN}{N} = \frac{TN}{FP+TN} = 1 - Fall\_out$$

$$F1\_score = \frac{2 \times TP}{2 \times TP+FP+FN}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{((TP+FP)(TP-FN)(TN+FP)(TN+FN))}}$$

where TP, TN, FP, FN stand for true positives, true negatives, false positives, and false negatives, respectively. MCC stands for Mathew's correlation coefficient. For all test datasets proposed method is executed 10 times and average values of those evaluation criteria's are reported here. For multiple features and Isolet dataset only accuracy is taken as evaluation criteria as these datasets are multi class in nature.

D. Comparative Study

To show the effectiveness of the proposed method rigorous test has been done and the result of those test are listed in (Table 3) for comparison purpose. For these tests following parameter values are used: Population number = 100, Max Iteration = 500, CR = [0, 1], F = [2, 0.5].

Comparative accuracy for different degree of proposed feature selection method and other methods are given in Tables 2–6. Each entry of the table is average taken over 10 independent runs.

Table 2. Performance comparison on Ionosphere and Sonar dataset

		Ionosphere					**Sonar**				
		NB	**SVM**	**K-NN**	**Boosting**	**C4.5**	**NB**	**SVM**	**k-NN**	**Boosting**	**C4.5**
FSFS	Accuracy (%)	73.70	91.80	75.41	85.93	80.10	70.83	80.24	68.51	77.16	70.11
	Recall	0.740	0.920	0.756	0.863	0.810	0.712	0.808	0.692	0.779	0.702
	Fallout	0.260	0.080	0.246	0.142	0.200	0.289	0.202	0.317	0.231	0.298
	Specificity	0.740	0.920	0.754	0.858	0.800	0.712	0.798	0.683	0.769	0.702
	F1 score	0.740	0.920	0.756	0.860	0.800	0.712	0.804	0.689	0.775	0.702
	MCC	0.480	0.840	0.510	0.721	0.610	0.423	0.606	0.375	0.548	0.404
LSFS	Accuracy (%)	76.80	91.40	84.67	88.83	82.90	71.88	81.01	67.98	75.67	75.25
	Recall	0.770	0.920	0.851	0.891	0.830	0.721	0.817	0.683	0.760	0.760
	Fallout	0.230	0.090	0.153	0.114	0.170	0.279	0.192	0.317	0.240	0.240
	Specificity	0.770	0.910	0.847	0.886	0.830	0.721	0.808	0.683	0.760	0.760
	F1 score	0.770	0.920	0.849	0.889	0.830	0.721	0.813	0.683	0.760	0.760
	MCC	0.540	0.830	0.698	0.778	0.660	0.442	0.625	0.365	0.519	0.519
MCFS	Accuracy (%)	87.90	94.20	82.11	90.46	88.20	67.36	82.45	70.14	77.21	73.12
	Recall	0.880	0.940	0.824	0.909	0.890	0.683	0.827	0.702	0.779	0.740
	Fallout	0.120	0.060	0.177	0.097	0.120	0.327	0.173	0.298	0.231	0.269
	Specificity	0.880	0.940	0.823	0.903	0.880	0.673	0.827	0.702	0.769	0.731
	F1 score	0.880	0.940	0.824	0.906	0.880	0.679	0.827	0.702	0.775	0.737
	MCC	0.760	0.890	0.647	0.812	0.770	0.356	0.654	0.404	0.548	0.471
DSFFC	Accuracy (%)	89.10	94.10	82.54	90.85	87.50	69.42	82.21	70.83	79.09	71.01
	Recall	0.890	0.940	0.829	0.909	0.880	0.702	0.827	0.721	0.798	0.712
	Fallout	0.110	0.060	0.176	0.091	0.130	0.308	0.183	0.279	0.212	0.289
	Specificity	0.890	0.940	0.824	0.909	0.880	0.692	0.817	0.721	0.789	0.712
	F1 score	0.890	0.940	0.826	0.909	0.880	0.699	0.823	0.721	0.794	0.712
	MCC	0.780	0.890	0.652	0.818	0.760	0.394	0.644	0.442	0.587	0.423

Table 2. Continued

		Ionosphere					Sonar				
		NB	SVM	K-NN	Boosting	C4.5	NB	SVM	k-NN	Boosting	C4.5
CFS	Accuracy (%)	77.300	78.60	79.88	80.00	68.30	60.12	59.35	70.11	68.25	67.00
	Recall	0.780	0.790	0.801	0.801	0.630	0.606	0.596	0.702	0.683	0.673
	Fallout	0.230	0.220	0.200	0.200	0.370	0.394	0.404	0.298	0.317	0.327
	Specificity	0.770	0.780	0.800	0.800	0.630	0.606	0.596	0.702	0.683	0.673
	F1 score	0.780	0.790	0.801	0.801	0.630	0.606	0.596	0.702	0.683	0.673
	MCC	0.550	0.570	0.601	0.601	0.27	0.212	0.192	0.404	0.365	0.346
CON	Accuracy (%)	67.60	63.20	70.00	72.35	63.00	61.29	63.55	69.53	62.39	65.00
	Recall	0.680	0.630	0.703	0.726	0.630	0.615	0.644	0.702	0.625	0.654
	Fallout	0.320	0.370	0.301	0.278	0.370	0.385	0.365	0.308	0.375	0.346
	Specificity	0.680	0.630	0.699	0.722	0.630	0.615	0.635	0.692	0.625	0.654
	F1 score	0.680	0.630	0.701	0.724	0.630	0.615	0.641	0.699	0.625	0.654
	MCC	0.360	0.270	0.402	0.447	0.270	0.231	0.279	0.394	0.250	0.308
PSO search	Accuracy (%)	70.00	70.30	73.93	78.05	73.00	70.00	69.53	62.39	76.55	70.33
	Recall	0.700	0.710	0.743	0.783	0.730	0.702	0.702	0.625	0.769	0.712
	Fallout	0.300	0.300	0.261	0.222	0.270	0.298	0.308	0.375	0.231	0.298
	Specificity	0.700	0.700	0.739	0.778	0.730	0.702	0.692	0.625	0.769	0.702
	F1 score	0.700	0.710	0.741	0.781	0.730	0.702	0.699	0.625	0.769	0.708
	MCC	0.400	0.410	0.482	0.561	0.460	0.404	0.394	0.250	0.539	0.414
Genetic search	Accuracy (%)	75.300	79.00	78.09	72.33	72.00	65	63.11	70.29	61.00	56.09
	Recall	0.760	0.790	0.794	0.726	0.720	0.654	0.635	0.712	0.615	0.577
	Fallout	0.250	0.210	0.210	0.278	0.280	0.346	0.365	0.298	0.394	0.433
	Specificity	0.750	0.790	0.790	0.722	0.720	0.654	0.635	0.702	0.606	0.567
	F1 score	0.760	0.790	0.792	0.724	0.720	0.654	0.635	0.708	0.612	0.574
	MCC	0.510	0.58	0.584	0.447	0.440	0.308	0.269	0.414	0.221	0.144
IMoDEFS	Accuracy (%)	90.10	93.70	83.87	91.28	90.60	68.77	84.18	75.05	81.83	73.99
	Recall	0.900	0.940	0.841	0.915	0.910	0.692	0.846	0.760	0.827	0.740
	Fallout	0.100	0.060	0.160	0.086	0.090	0.308	0.154	0.250	0.183	0.260
	Specificity	0.900	0.940	0.840	0.914	0.910	0.692	0.846	0.750	0.817	0.740
	F1 score	0.900	0.940	0.841	0.915	0.910	0.692	0.846	0.756	0.823	0.740
	MCC	0.810	0.88	0.681	0.829	0.820	0.385	0.692	0.510	0.644	0.481
Proposed	Accuracy (%)	89.70	93.10	84.89	84.11	87.10	73.56	83.65	79.98	76.65	79.12
	Recall	0.900	0.930	0.851	0.846	0.870	0.74	0.837	0.808	0.769	0.798
	Fallout	0.100	0.070	0.153	0.159	0.130	0.26	0.164	0.202	0.231	0.212
	Specificity	0.900	0.930	0.847	0.841	0.870	0.74	0.837	0.798	0.769	0.789
	F1 score	0.900	0.930	0.849	0.843	0.870	0.74	0.837	0.804	0.769	0.794
	MCC	0.800	0.860	0.698	0.687	0.740	0.481	0.673	0.606	0.539	0.587

Table 3. Performance comparison on Spam and WDBC dataset

		Spam					WDBC				
		NB	**SVM**	**k-NN**	**Boosting**	**C4.5**	**NB**	**SVM**	**k-NN**	**Boosting**	**C4.5**
FSFS	Accuracy (%)	66.70	79.00	80.81	66.85	77.00	91.11	94.41	93.22	94.22	95.55
	Recall	0.670	0.790	0.808	0.669	0.770	0.912	0.947	0.933	0.944	0.958
	Fallout	0.330	0.210	0.192	0.332	0.230	0.088	0.056	0.067	0.056	0.046
	Specificity	0.670	0.790	0.808	0.668	0.770	0.912	0.944	0.933	0.944	0.954
	F1 score	0.670	0.790	0.808	0.669	0.770	0.912	0.946	0.933	0.944	0.956
	MCC	0.330	0.580	0.617	0.337	0.540	0.824	0.891	0.866	0.888	0.912
LSFS	Accuracy (%)	69.30	83.80	82.86	69.28	69.10	93.71	96.87	95.87	95.85	90.29
	Recall	0.690	0.840	0.827	0.693	0.690	0.940	0.972	0.961	0.961	0.905
	Fallout	0.310	0.160	0.173	0.307	0.310	0.063	0.032	0.042	0.042	0.098
	Specificity	0.690	0.840	0.827	0.693	0.690	0.937	0.968	0.958	0.958	0.902
	F1 score	0.690	0.840	0.827	0.693	0.690	0.939	0.970	0.960	0.960	0.903
	MCC	0.390	0.680	0.654	0.386	0.380	0.877	0.940	0.919	0.919	0.807
MCFS	Accuracy (%)	65.30	80.00	82.27	65.25	72.60	93.39	96.68	96.22	95.11	88.69
	Recall	0.650	0.800	0.823	0.653	0.730	0.937	0.968	0.965	0.954	0.888
	Fallout	0.350	0.200	0.177	0.347	0.270	0.067	0.031	0.039	0.049	0.113
	Specificity	0.650	0.800	0.823	0.653	0.730	0.933	0.968	0.961	0.951	0.887
	F1 score	0.650	0.800	0.823	0.653	0.730	0.935	0.968	0.963	0.953	0.888
	MCC	0.310	0.600	0.646	0.305	0.450	0.87	0.937	0.926	0.905	0.775
DSFFC	Accuracy (%)	75.60	86.70	84.31	75.71	69.90	94.34	96.82	95.73	96.22	91.23
	Recall	0.760	0.870	0.844	0.757	0.700	0.944	0.968	0.958	0.965	0.916
	Fallout	0.240	0.130	0.157	0.243	0.300	0.056	0.032	0.042	0.039	0.088
	Specificity	0.760	0.870	0.843	0.757	0.700	0.944	0.968	0.958	0.961	0.912
	F1 score	0.760	0.870	0.843	0.757	0.700	0.944	0.968	0.958	0.963	0.914
	MCC	0.510	0.730	0.687	0.515	0.400	0.888	0.937	0.916	0.926	0.828
CFS	Accuracy (%)	76.30	79.10	78.59	70.00	70.10	90.00	91.22	93.31	92.56	92.32
	Recall	0.760	0.790	0.786	0.700	0.700	0.902	0.916	0.933	0.926	0.926
	Fallout	0.240	0.210	0.214	0.300	0.300	0.099	0.088	0.067	0.074	0.077
	Specificity	0.760	0.790	0.786	0.700	0.700	0.901	0.912	0.933	0.926	0.923
	F1 score	0.760	0.790	0.786	0.700	0.700	0.902	0.914	0.933	0.926	0.924
	MCC	0.530	0.580	0.572	0.400	0.400	0.803	0.828	0.866	0.852	0.849
CON	Accuracy (%)	70.00	70.00	69.03	68.95	65.10	88.59	89.56	90.23	91.69	89.99
	Recall	0.700	0.700	0.691	0.690	0.650	0.888	0.898	0.905	0.919	0.902
	Fallout	0.300	0.300	0.31	0.310	0.350	0.113	0.105	0.098	0.084	0.099
	Specificity	0.700	0.700	0.69	0.690	0.650	0.887	0.895	0.902	0.916	0.901
	F1 score	0.700	0.700	0.691	0.690	0.650	0.888	0.896	0.903	0.917	0.902
	MCC	0.400	0.400	0.381	0.379	0.300	0.775	0.793	0.807	0.835	0.803
PSO search	Accuracy (%)	73.50	79.10	81.00	72.35	76.00	91.29	92.35	91.36	90.25	89.89
	Recall	0.740	0.790	0.810	0.724	0.760	0.916	0.926	0.916	0.905	0.901
	Fallout	0.270	0.210	0.190	0.277	0.240	0.088	0.077	0.088	0.098	0.102
	Specificity	0.740	0.790	0.810	0.277	0.760	0.912	0.923	0.912	0.902	0.898
	F1 score	0.740	0.790	0.810	0.724	0.760	0.914	0.924	0.914	0.903	0.900
	MCC	0.470	0.580	0.620	0.447	0.520	0.828	0.849	0.828	0.807	0.800

Table 3. Continued

		Spam					WDBC				
		NB	**SVM**	**k-NN**	**Boosting**	**C4.5**	**NB**	**SVM**	**k-NN**	**Boosting**	**C4.5**
Genetic search	Accuracy (%)	70.20	62.10	63.39	69.99	70.10	90.29	90.35	89.39	88.36	85.78
	Recall	0.700	0.620	0.634	0.700	0.700	0.905	0.905	0.895	0.884	0.860
	Fallout	0.300	0.380	0.366	0.300	0.300	0.098	0.095	0.106	0.116	0.141
	Specificity	0.700	0.620	0.634	0.700	0.700	0.902	0.905	0.894	0.884	0.859
	F1 score	0.700	0.620	0.634	0.700	0.700	0.903	0.905	0.895	0.884	0.860
	MCC	0.410	0.240	0.268	0.400	0.400	0.807	0.81	0.789	0.768	0.719
IMoDEFS	Accuracy (%)	76.10	87.80	85.47	75.99	91.80	93.18	96.38	95.73	96.27	95.20
	Recall	0.760	0.880	0.855	0.760	0.920	0.933	0.965	0.958	0.965	0.954
	Fallout	0.240	0.120	0.145	0.240	0.080	0.067	0.035	0.042	0.039	0.049
	Specificity	0.760	0.880	0.855	0.760	0.920	0.933	0.965	0.958	0.961	0.951
	F1 score	0.760	0.880	0.855	0.760	0.920	0.933	0.965	0.958	0.963	0.953
	MCC	0.520	0.760	0.710	0.520	0.840	0.866	0.93	0.916	0.926	0.905
Proposed	Accuracy (%)	78.90	92.50	91.22	88.56	90.70	95.37	85.66	96.89	92.23	95.72
	Recall	0.790	0.080	0.913	0.886	0.910	0.954	0.859	0.972	0.923	0.958
	Fallout	0.210	0.930	0.088	0.114	0.090	0.046	0.144	0.032	0.078	0.042
	Specificity	0.790	0.930	0.912	0.886	0.910	0.954	0.856	0.968	0.923	0.958
	F1 score	0.790	0.930	0.912	0.886	0.910	0.954	0.858	0.97	0.923	0.958
	MCC	0.580	0.850	0.825	0.771	0.810	0.909	0.715	0.94	0.845	0.916

Table 4. Performance comparison (accuracy, %) on Spectf and Multiple features dataset

	Spectf					Multiple features				
	NB	**SVM**	**k-NN**	**Boosting**	**C4.5**	**NB**	**SVM**	**K-NN**	**Boosting**	**C4.5**
FSFS	73.63	73.38	66.00	65.50	65.50	96.00	97.90	94.49	96.54	91.00
LSFS	72.79	74.00	69.63	69.00	62.33	94.00	97.70	93.02	96.15	92.00
MCFS	72.13	71.88	66.38	72.75	63.21	96.00	98.10	95.58	97.06	94.00
DSFFC	79.75	76.88	68.13	76.88	76.12	94.00	98.40	95.61	96.22	91.00
CFS	70.23	71.11	70.21	69.22	60.29	92.00	94.60	93.99	96.55	92.00
CON	70.11	63.33	61.39	65.55	60.00	93.00	94.60	93.99	96.55	92.00
PSO search	71.11	72.19	70.11	61.15	65.59	92.00	90.20	93.55	90.00	90.00
Genetic search	61.00	62.00	63.35	59.99	60.10	90.00	91.50	88.92	89.99	90.00
IMoDEFS	85.75	73.88	66.8	74.63	66.25	95.00	98.20	95.99	96.21	95.00
Proposed	79.20	76.69	72.50	72.35	79.00	99.00	97.20	96.44	97.84	97.00

Table 5. Performance comparison (accuracy, %) on Isolet dataset

	Isolet				
	NB	**SVM**	**k-NN**	**Boosting**	**C4.5**
FSFS	66.00	88.20	71.42	65.78	61.70
LSFS	76.00	93.00	82.60	75.53	76.60
MCFS	82.00	95.80	87.99	81.99	78.30
DSFFC	84.00	95.30	86.19	84.82	79.60
CFS	80.00	82.30	83.69	84.81	81.20
CON	80.00	78.20	70.11	76.25	71.20
PSO search	80.00	81.30	82.55	83.92	69.00
Genetic search	70.00	68.20	71.55	68.99	75.30
IMoDEFS	86.00	95.50	87.79	86.03	80.70
Proposed	86.00	96.00	89.46	85.45	85.00

From Table 6, it is clear that proposed method works better than FSFS, LSFS, MCFS, DSFFC, CFS, CON, PSO search, Genetic search, and is a close contender of IMoDEFS method and in some cases outperforms them.

The experiments are carried out in MATLAB using workstation having 5 GHz octa-core CPU and 32 GB of RAM.

V. CONCLUSIONS

In this work, an algorithm for feature selection using modified binary differential evolution is proposed. For this purpose two objective functions, namely division based

Table 6. Number of times best results are obtained by each of the above mentioned feature selection algorithm

	Ionosphere	Sonar	Spam	Wdbc	Isolet	Multiple features	Spectf
FSFS	0	0	0	0	0	0	0
LSFS	0	0	0	1	0	0	0
MCFS	1	0	0	0	0	0	0
DSFFC	0	0	0	0	0	0	3
CFS	0	0	0	0	0	0	0
CON	0	0	0	0	0	0	0
PSO search	0	0	0	0	0	0	0
Genetic search	0	0	0	0	0	0	0
IMoDEFS	3	2	1	1	1	1	0
Proposed	1	3	4	3	4	4	2

feature score and set lower boundary approximation, are introduced to be used along with proposed global version of binary differential evolution algorithm. Effectiveness of proposed algorithm is shown by comparing it with other several state of the art feature selection methods by means of different statistical accuracy measures. Computational complexity of proposed method is relatively higher than other compared algorithms but on average proposed method gives significantly better result. The proposed method used Pareto-based approach to tackle two objective optimization, thus there are several other approaches to tackle this kind of problem in future applicability of those methods along with proposed differential evolution algorithm. Finally, a detailed sensitivity analysis of the feature selection technique with various algorithmic parameter need to be done in future.

REFERENCES

1. M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131-156, 1997.
2. S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Proceedings of the 18th International Conference on Machine Learning* (*ICML)*, Williamstown, MA, 2001, pp. 74-81.
3. Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507-2517, 2007.
4. R. Bellman, "Dynamic programming and Lagrange multipliers," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 42, no. 10, 767, 1956.
5. R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1, pp. 273-324, 1997.
6. P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301-312, 2002.
7. Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "$l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *Proceedings of 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain, 2011, pp. 1589-1594.
8. D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, 2010, pp. 333-342.
9. R. Jensen and Q. Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. Hoboken, NJ: John Wiley & Sons, 2008.
10. R. Jensen and Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Transactions Fuzzy Systems*, vol. 17, no. 4, pp. 824-838, 2009.
11. N. Parthalain, Q. Shen, and R. Jensen, "A distance measure approach to exploring the rough set boundary region for attribute reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 3, pp. 305-317, 2010.
12. A. K. Das, S. K. Pati, and S. Chakrabarty, "Reduct generation of microarray dataset using rough set and graph theory for unsupervised learning," in *Proceedings of the 2nd International Conference on Computational Science, Engineering and Information Technology*, Coimbatore, India, 2012, pp. 555-561.
13. S. Sengupta and A. K. Das, "Dimension reduction using clustering algorithm and rough set theory," in *Swarm, Evolutionary, and Memetic Computing.* Heidelberg: Springer, 2012, pp. 705-712.
14. M. Dash and H. Liu, "Consistency-based search in feature selection," *Artificial Intelligence*, vol. 151, no. 1, pp. 155-176, 2003.
15. S. K. Pati, A. K. Das, and A. Ghosh, "Gene selection using multi-objective genetic algorithm integrating cellular automata and rough set theory," in *Swarm, Evolutionary, and Memetic Computing.* Heidelberg: Springer, 2013, pp. 144-155.
16. J. Wroblewski and M. Szczuka, "Neural network architecture for synthesis of the probabilistic rule based classifiers," *Electronic Notes in Theoretical Computer Science*, vol. 82,

no. 4, pp. 251-262, 2003.

17. G. L. Pappa, A. A. Freitas, and C. A. Kaestner, "Attribute selection with a multi-objective genetic algorithm," in *Advances in Artificial Intelligence.* Heidelberg: Springer, 2002, pp. 280-290.
18. B. Liu, F. Liu, and X. Cheng, "An adaptive genetic algorithm based on rough set attribute reduction," in *Proceedings of 2010 3rd International Conference on Biomedical Engineering and Informatics*, Yantai, China, 2010, pp. 2880-2883.
19. D. P. Muni, N. R. Pal, and J. Das, "Genetic programming for simultaneous feature selection and classifier design," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 1, pp. 106-117, 2006.
20. X. Wang, J.Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459-471, 2007.
21. R. Storn and K. Price, "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.
22. S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4-31, 2011.
23. A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398-417, 2009.
24. J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945-958, 2009.
25. M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 99-119, 2011.
26. S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482-500, 2012.
27. Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55-66, 2011.
28. R. Mallipeddi and P. N. Suganthan, "Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies," in *Swarm, Evolutionary, and Memetic Computing*. Heidelberg: Springer, 2010, pp. 71-78.
29. Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985-2999, 2008.
30. A. Zamuda, J. Brest, B. Boskovic, and V. Zumer, "Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC2008)*, Hong Kong, 2008, pp. 3718-3725.
31. K. E. Parsopoulos, "Cooperative micro-differential evolution for high-dimensional problems," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, Montreal, Canada, 2009, pp. 531-538.
32. S. Z. Zhao, P. N. Suganthan, and S. Das, "Self-adaptive differential evolution with multi-trajectory search for large-scale optimization," *Soft Computing*, vol. 15, no. 11, pp. 2175-2185, 2011.
33. J. Brest and M. S. Maucec, "Self-adaptive differential evolution algorithm using population size reduction and three strategies," *Soft Computing*, vol. 15, no. 11, pp. 2157-2174, 2011.
34. H. Wang, Z. Wu, and S. Rahnamayan, "Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems," *Soft Computing*, vol. 15, no. 11, pp. 2127-2140, 2011.
35. M. Weber, F. Neri, and V. Tirronen, "Shuffle or update parallel differential evolution for large-scale optimization," *Soft Computing*, vol. 15, no. 11, pp. 2089-2107, 2011.
36. Z. Pawlak, "Rough set approach to knowledge-based decision support," *European Journal of Operational Research*, vol. 99, no. 1, pp. 48-57, 1997.
37. K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Hoboken, NJ: John Wiley & Sons, 2001.
38. K. Bache and M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml.
39. X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," *Advances in Neural Information Processing Systems*, vol. 18, pp. 507-514, 2005.
40. D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, 2010, pp. 333-342.
41. S. Bandyopadhyay, T. Bhadra, P. Mitra, and U. Maulik, "Integration of dense subgraph finding with feature clustering for unsupervised feature selection," *Pattern Recognition Letters*, vol. 40, pp. 104-112, 2014.
42. M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, Hamilton, New Zealand, 1999.
43. J. Garcia-Nieto, E. Alba, L. Jourdan, and E. Talbi, "Sensitivity and specificity based multiobjective approach for feature selection: application to cancer diagnosis," *Information Processing Letters*, vol. 109, no. 16, pp. 887-896, 2009.
44. M. A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1437-1447, 2003.
45. T. Bhadra and S. Bandyopadhyay, "Unsupervised feature selection using an improved version of differential evolution," *Expert Systems with Applications*, vol. 42, no. 8, pp. 4042-4053, 2014.
46. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and H. I. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10-18, 2009.
47. G. Forney, "Generalized minimum distance decoding," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125-131, 1966.
48. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*,

2nd ed. New York, NY: John Wiley & Sons, 2001.

49. V. Vapnik, *The Nature of Statistical Learning Theory*, 1st ed. New York, NY: Springer, 1995.

50. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119-139, 1997.

51. J. R. Quinlan, *C4. 5: Programs for Machine Learning*, 1st ed. San Mateo, CA: Morgan Kaufmann, 1993.

Sunanda Das

Sunanda Das is an Assistant Professor of Computer Science and Engineering at Neotia Institute of Technology, Management and Science, West Bengal, India. She has received M.Tech. degree in Computer Science and Technology from Haldia Institute of Technology, West Bengal, India. She is registered for Ph.D. (Engg.) degree at IIEST, Shibpur, Howrah. Her research interests include data mining and pattern recognition, bio-informatics, rough set theory, etc.

Chi-Chang Chang

Chi-Chang Chang is currently an associate professor of medical informatics in the Chung Shan Medical University and also a consultant on Cancer Center, Biomedical Industry Research Center & Medical Information Center, Chung-Shan Medical University Hospital. Dr. Chang received the Ph.D. degree in Industry Engineering Management from the Yuan-Ze University. He is academic member of the Association for Information Systems and Society for Medical Decision Making. His primary research interests are in the areas of medical decision analysis, shared medical decision making, and clinical operational research. He has edited several special issues for journals such as *The Journal of Universal Computer Science (SCI), International Journal of Management, Economics and Social Sciences (Ei Inspect), Journal of Quality (Ei Compendex), Journal of Computing Science and Engineering (Ei Compendex)* and so on. Dr. Chang has rich experience in keynote address such as MESS2016, ICMLC2016, ICBP2016, PSYBEHAV2016, ICRE2016, MES2016, MEDLIFE2016, ICMLC2017, and PSYBEHAV2017.

Asit Kumar Das

Asit Kumar Das is Associate Professor of Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India. He has received B.Sc. Honours in Mathematics, B. Tech. and M.Tech degree in Computer Science and Engineering from Calcutta University. He obtained Ph.D. (Engg.) degree from Bengal Engineering and Science University, Shibpur, Howrah. He teaches discrete structures, database management systems, design and analysis of algorithm, introduction to data mining, etc. His research area of interest is data mining and pattern recognition, machine learning, bioinformatics, social networks, text and image mining, rice disease prediction, crime data analysis, audio and video data analysis, etc.

Arka Ghosh

Arka Ghosh was born in Burdwan, West Bengal, India in 1990. He is currently working towards Ph.D. degree in the Department of Computer Science & Technology, Indian Institute of Engineering Science & Technology, Shibpur, India. He has published research articles in peer reviewed journals and international conference proceedings. His research interest includes evolutionary algorithms, and data mining.