

Modified Binary Bat Algorithm for Feature Selection in Unsupervised Learning

Rajalaxmi Ramasamy and Sylvia Rani

Department of Computer Science and Engineering, Kongu Engineering College, India

Abstract: Feature selection is the process of selecting a subset of optimal features by removing redundant and irrelevant features. In supervised learning, feature selection process uses class label. But feature selection is difficult in unsupervised learning since class labels are not present. In this paper, we present a wrapper based unsupervised feature selection method with the modified binary bat approach with *k*-means clustering algorithm. To ensure diversification in the search space, mutation operator is introduced in the proposed algorithm. To validate the selected features by our method, classification algorithms like decision tree induction, Support Vector Machine and Naïve Bayesian classifier are used. The results show that the proposed method identifies a minimal number of features with improved accuracy when compared with the other methods.

Keywords: Feature selection, unsupervised learning, binary bat algorithm, mutation.

Received March 10, 2015; accepted December 21, 2015

1. Introduction

Data mining is the process of knowledge discovery by analyzing the large amount of data [10]. It has various applications and these applications have enriched the various fields of human life, including business, education, medical, scientific, etc., The data collection process never looks into the usefulness of different dimensions of data. Hence, it is essential to identify the useful dimensions/features related to the domain to get relevant knowledge. Feature selection is the process of selecting a subset of relevant features for use in model construction [4]. Finding a solution for the feature selection problem is trivial. Hence, approximation algorithms have been proposed to deal with this [36].

These algorithms are mainly categorized as filter, wrapper, embedded and hybrid approaches. Filter approach [8] is independent of the learning algorithm that is adopted to tune and/or build the system that exploits the selected variables as inputs. The wrapper approach considers machine learning as a black box in order to select subsets of variables on the basis of their predictive power. Unlike previous methods, embedded approach performs the variable selection in the learning machine. Table 1 shows a comparison [26] among the approaches.

All of the feature selection approaches can be applied for supervised as well as unsupervised learning [12]. In supervised learning, feature selection process is guided by the class labels of the data. On the other hand, unsupervised learning method uses data without class labels [22, 29]. Traditional clustering methods group the objects based on the similarity between them. It is determined using different distance measures.

These measures utilize the feature information to identify the membership of an object to a cluster. When the feature space is high, it leads to misclassify the objects. Hence, it becomes necessary to identify the relevant features.

Table 1. Comparison of feature selection methods.

Method	Advantages	Disadvantages	Examples
Filter	Univariate		
	Independent of Learning algorithm, Scalable and Fast	Ignores learning algorithm, Ignores relationship among features	Chi-square, Information gain, Gain ratio, Fishers score, Gini index etc.,
	Multivariate		
	Independent of Learning algorithm, Utilize feature relationships, fast	Ignores learning algorithm, Poor Scalability, Slow	Correlation based feature selection, Minimal-Redundancy-maximal Relevance etc.,
Wrapper	Deterministic		
	Use Learning algorithm, Utilize feature relationships, fast	Learning algorithm dependent, Leads to overfitting, Easy to get trapped in local optima	Sequential Forward Selection, Sequential Backward Elimination, etc.,
	Randomized		
	Use Learning algorithm, Utilize feature relationships, Avoids trapping in local optimum	Learning algorithm dependent, computationally intensive, More prone to overfitting	Genetic Algorithm, Particle Swarm Optimization, ant Colony Optimization, Firefly algorithm etc.,
Embedded	Use Learning algorithm, Utilize feature relationships, Fast	Learning algorithm dependent	Decision trees, weighted Naïve bayes etc.,

Computational intelligence methods help to provide an approximate solution for the feature selection

problem [6, 21, 28, 30, 36]. They are widely accepted due to their ability to provide near-optimal solution. Many approaches have been proposed for feature selection using swarm intelligence [3]. Recently, bat algorithm has gained popularity in solving optimization problems. This paper proposes a Modified Binary Bat Algorithm (MBBA) for feature selection in unsupervised learning. An enhanced version of the binary bat algorithm with mutation operator is designed to cope with premature convergence of BBA algorithm.

The MBBA is a wrapper method with every feature subset being evaluated using k-means clustering technique. The experimental results indicate that the proposed method show improved performance in comparison to other methods.

The rest of the paper is organized as follows: section 2 provides an overview of unsupervised feature selection methods. In section 3, we discuss UFS-MBBA approach in detail. The experimental procedure and results are discussed in section 4 and 5. Finally, conclusion and future work are presented in section 6.

2. Related Work

In the unsupervised feature selection based on filter methods, they analyze the dependency among the features without using clustering algorithms [2, 12, 14, 22]. These methods are broadly categorized into univariate [12] and multivariate methods [18, 31].

Wrapper approaches identify a feature subset with optimal value of the predefined criterion, such as the sum of squared error [4] or the normalized likelihood [4] or the DB-index [23]. The author [11] suggested a method to select a feature subset that can produce clusters most similar to the one obtained by the ensemble learning algorithm. Random Forests (RF) has been also extended to unlabeled data leading to unsupervised learning [29]. A localized feature selection algorithm was proposed in [19] by computing normalized scatter separability values for each individual cluster for low-dimensional data.

In embedded approach, the search for an optimal feature subset is built into the clustering algorithm [5, 7, 13]. Entropy based k-means for the subspace clustering algorithm [15] and the feature group weighting k-means for the subspace clustering algorithm [9] deal with high dimensional datasets. Also, co-clustering methods have been developed to perform unsupervised feature selection [17, 31].

Various feature selection algorithms for classification and clustering were surveyed in [16, 20, 33, 35]. Genetic Algorithm (GA) based wrapper method for classification of hyper spectral data using Support Vector Machine (SVM), a state-of-art classifier has found success in a variety of areas [36]. A wrapper approach to feature selection was developed using GA.

Saxena *et al.* [27] the authors proposed four methods to select a set of features using a task independent criterion. The authors [17] proposed a new UnSupervised Quick Reduct (USQR) algorithm using rough set theory. A new and robust unsupervised feature selection approach that evaluated terms in groups was developed in [28]. The authors in [30] presented an Unsupervised Feature Selection method based on Ant Colony Optimization, called UFSACO with low computational complexity. The authors [25] presented a wrapper feature selection approach based on BA and Optimum-Path Forest (OPF) for supervised learning with improved classification effectiveness.

The results are compared with other meta-heuristic algorithms like Firefly Algorithm (FA), Gravitational Search Algorithm (GSA), Harmony Search (HS) and Particle Swarm Optimization (PSO) algorithms. These algorithms on the whole have poor exploitation capabilities in comparison to one another. In paper [14], the authors proposed a method using UnSupervised Particle Swarm Optimization based Relative Reduct (US-PSO-RR).

Many feature selection methods, including filter, wrapper and embedded approaches have been developed. Even though, BA with OPF has been used for feature selection, it is used in supervised learning.

As UFSACO method produces good results, the selected features can be further reduced. Both these methods have poor exploitation capabilities which need to be improved. The proposed method tries to improve the local search ability of BA and uses k-means algorithm as a wrapper to select features in unsupervised learning.

3. Modified Binary Bat Algorithm

3.1. Bat Algorithm

Bat Algorithm (BA) has been developed based on the echolocation behavior of bats [25]. In BA, an artificial bat has a position, velocity and frequency vectors which are updated during the course of iterations. The artificial bats move around the search space utilizing the position and velocity vectors within the continuous real domain. For each bat (b_i), it has a position (x_i), frequency (f_i) and velocity (v_i). At each step t , the bats move to the next position with new velocities as follows:

$$v_i(t+1) = v_i(t) + (x_i(t) - gbest)f_i \quad (1)$$

Where $gbest$ is the best solution obtained so far. Now, the position of the bat is updated as follows:

$$x_i(t+1) = x_i(t) + v_i(t) \quad (2)$$

The frequency of i^{th} bat is computed at each iteration as follows:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (3)$$

Where β is a random number uniformly distributed in the range [0, 1]. The exploitation capability of BA is improved with a random walk method as given below:

$$x_{new} = x_{old} + \varepsilon A^t \quad (4)$$

Where ε is a random number between [-1, 1] and A is the loudness. For each iteration, the loudness and pulse emission rate are adjusted as follows:

$$A_i(t+1) = \alpha A_i(t) \quad (5)$$

$$r_i(t+1) = r_i(0) + [1 - \exp(-\gamma t)] \quad (6)$$

Where α and γ are constants.

BA is simple and efficient. Other swarm intelligence algorithms use a constant value for the standard parameters. From Equations (5) and (6) it is noticed that, loudness and pulse rate are adjusted at every iteration.

Due to this, it is possible for the algorithm to switch between exploration and exploitation [34].

3.2. Binary Bat algorithm

In feature selection problem, each feature subset is coded as a binary string of 1s and 0s where 1 denotes the presence and 0 represents the absence of a feature.

To map the problem in BA, each bat's position is considered as binary values leading to form a binary search space. The bats can move to a new position by flipping various numbers of bits. Thus, the binary version of BA must employ a different strategy to update its velocity and position. Therefore, a transfer function [21] is used for changing bat's positions from '0' to '1' or vice versa.

$$V(v_i^k) = \left\lfloor \frac{2}{\pi} \arctan\left(\frac{\pi}{2}\right) v_i^k \right\rfloor \quad (7)$$

Where v_i^k is the velocity of the bat i in k th dimension.

Now the position of the bat is modified as

$$x_i^k = \begin{cases} (x_i^k)^{-1} & \text{if } V(v_i^k) > \delta \\ x_i^k & \text{if } V(v_i^k) \leq \delta \end{cases} \quad (8)$$

Where δ is uniformly distributed between [0, 1]. After updating the position, the fitness of each bat is evaluated. Since this work deals with unsupervised feature selection, the fitness is evaluated based on the cluster quality measured using Sum of Squared Error [10] as follows:

$$fitness(b_i) = \sum_{j=1}^k \sum_{x_j \in c_j} \|x_i - m_j\|^2 \quad (9)$$

3.3. Mutation in Binary Bat Algorithm

In BBA the bat's positions are encoded as binary strings. Hence, it is difficult for them to deviate from the solution space. But, there is a risk for premature

convergence. This leads to getting the bats struck at sub-optimal solutions. In order to move them to untouched areas of the search space we introduced mutation operator as follows:

$$x_{ij} = \begin{cases} \square x_{ij} & \text{if } \text{rand}() \leq r_m \\ x_{ij} & \text{otherwise} \end{cases} \quad (10)$$

Where r_m is the probability of random mutation. After updating the bats position as per Equations (7) and (8), the bits are mutated with probability r_m . To ensure that at least one bit in each bat gets flipped, the mutation probability r_m is set as $1/d$ where d represents the number of dimensions of the dataset. The pseudocode of UFS-MBBA is presented in Algorithm 1. The initial population contains bats with random number of 0's and 1's. To generate bats with good exploration, random mutation is performed on the bats. On each iteration the position is updated using the transfer function. The fitness of new bats is calculated as per Equation (9). If the fitness of initial bat is less than the fitness of new bat and the generated random number is greater than the loudness, then the initial bat is updated. Next the loudness is reduced and the pulse rate is increased. Again, mutation is applied over the bats to improve exploration. Now, the fitness is evaluated. If the new fitness is less than the minimum fitness of the initial population, then the initial Gbest is updated. The number of selected features in final Gbest will be the final subset of selected features.

Algorithm 1: Pseudocode of UFS-MBBA

Input:

$r=0.9, A=0.5, \text{Number of bats}=10, \text{max}=50, r_m$

begin

initialize the bat population

Apply mutation on the bats as per equation(10)

fitness1=fitness of initial bats are calculated using equation (9)

min_fit=bat with minimum fitness value

gbest=bat with minimum fitness

while (t<max)

adjust frequency and velocity

calculate transfer function

if (T<rand) then

generate new bats

end

if (rand>r) then

update newbats with gbest

end

improve new bat using mutation operator as per equation (10)

fitness2=fitness of newbats are calculated using equation (9)

if (fitness1<fitness2 && rand>A) then

update initial bat & reduce loudness, increase pulse rate

end

if (fitness2<min_fit) then

update gbest

end

end

3.4. Classification Algorithms

The selected features obtained are independent of the classifiers. Hence, experiments are conducted to validate the selected features relevance to the dataset with the classification algorithms. C4.5 is a well-known algorithm [10] used to generate a decision tree. It follows a top-down approach, starting with a training set of tuples and their associated class labels. SVM is a popular classifier used for classifying both linear and non-linear data. To separate the classes, data can be mapped to a high dimensional space to find the hyperplane. In order to perform mapping, some popular kernel functions like linear kernel, polynomial kernel, radial basis kernel and sigmoid kernel are used. In this work, SVM [10] with polynomial kernel is used.

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class [10]. In this work the above algorithms are implemented using WEKA tool [32].

4. Experimental Procedure

The proposed algorithm is implemented in *Java* using Intel core i5 processor with 4 GB RAM and 500 GB hard disk running on Windows 7 OS.

4.1. Dataset Description

The proposed method is tested on eight different datasets which are publicly available in UCI machine learning repository. The datasets and their description are given in the Table 1. The first four datasets are selected for evaluating the BA algorithm with Optimum path classifier [24]. The remaining datasets are taken from [26] which evaluated the UFSACO algorithm.

The class label of all these datasets is removed in the proposed method as it uses an unsupervised learning approach. The class labels of the dataset are used while evaluating the selected features using supervised learning algorithms.

Table 2. Dataset description.

S.No.	Data Set	No. of Samples	No. of features	Classes
1	SVM Guide	3089	4	2
2	Splice	1000	60	2
3	Ionosphere	351	34	2
4	Wisconsin Breast Cancer(WBC)	683	10	2
5	Dermatology	366	34	6
6	Hepatitis	155	19	2
7	Wine	178	13	3
8	WDBC	569	30	2

4.2. Evaluation Parameters

The performance of the proposed method is validated using accuracy and number of selected features. A confusion matrix [10] is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.

Confusion matrix is also known as a contingency table or an error matrix. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The confusion matrix consists of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The confusion matrix is shown in Table 3.

Table 3. Confusion matrix.

		PREDICTION	
		0	1
AC	TU	0	TN
	AI	1	FN
		FP	TP

The test samples can be classified into one of the following groups:

- *True Positive.* the positive tuples that were correctly labeled
- *False Positive.* the negative tuples that were incorrectly labeled
- *True Negative.* the negative tuples that were correctly labeled
- *False Negative.* the positive tuples that were incorrectly labeled

Accuracy of the classifier is the ratio of the number of samples correctly classified to the total number of samples which is given as follows:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100 \tag{11}$$

The features that give highest classification accuracy is the best features. The accuracy of the selected features is tested with the classification algorithms like decision tree induction, naïve Bayes and SVM.

5. Results and Discussion

In the experiments, the classification accuracy and the number of features selected is used as the performance measure. The classification model is developed by 10-fold cross validation with the selected features of the proposed method. The number of agents and generations for other meta-heuristic methods are 30 and 100 respectively. In UFS-MBBA, the number of bats and generations is 10 and 50 respectively.

Initially, the performance of UFS-MBBA is evaluated over three classification algorithms DT, NB and SVM. Then, the classifier that gives overall best classification accuracy with minimal features is selected. The comparison of UFS-MBBA is done with the previous methods in the literature. Since different datasets were used by the researchers in the previous

studies, we have divided the experiments into two sets. In the first set, we compare the results of the first four datasets of Table 2. The second set of experiments used the remaining datasets in Table 2.

Tables 4 presents the features selected using the proposed method. The method is compared with other swarm intelligence methods, namely Firefly Algorithm (FA) [30], Gravitational Search Algorithm (GSA) [24], Harmony Search (HS) and BA with OPF. Any swarm intelligence based algorithm is claimed as best based on its exploration and exploitation capability. Although, these methods are simple and efficient, they have the possibility to get stuck in a local optimum. The main advantages [33] of bat algorithm are quick convergence rate and variability in the control parameters (pulse rate and loudness). Due to these factors, BA seems to be better than other swarm intelligence algorithms. To further enhance the performance, we have extended BA with the mutation operator to improve exploitation.

From the results, it is found that, UFS-MBBA identified minimal features in comparison to previous methods.

Table 4. Comparison of features selected using UFS-MBBA for first four datasets.

Algorithm	Dataset			
	SVMGuide1	WBC	Ionosphere	Splice
UFS-MBBA	2	3	12	19
BA with OPF	2	5	16	30
FA	3	4	18	35
GSA	3	4	14	28
HS	2	6	20	34
PSO	3	4	14	28

Table 5 summarizes the average classification accuracy over 10 independent runs of the three classifiers with the selected features. It is noticed that, SVMGuide1 and Ionosphere datasets achieved high accuracy using DT classification. However, WBC and Splice datasets achieved improved accuracy, using NB classifier.

Table 5. Accuracy with different classifiers using UFS-MBBA for first four datasets.

Classifier	Dataset			
	SVMGuide1	WBC	Ionosphere	Splice
Decision Tree	96.60	93.56	90.88	82.60
Naïve Bayes	91.03	96.34	79.77	84.10
SVM	94.30	96.05	84.90	83.61

Table 6 provides the overall comparison of the classification accuracy against the previous methods. The improvement percentage of UFS-MBBA's classification accuracy with BA over the four datasets are 12.34%, 18.9%, 14.88% and 19.1% respectively. Also, the proposed method obtains higher accuracy than the other methods.

Table 6. Comparison of accuracy with other algorithms for first four datasets.

Algorithm	Dataset			
	SVMGuide1	WBC	Ionosphere	Splice
UFS-MBBA	96.60	96.34	90.88	84.1
BA with OPF	84	77	76	65
FA	94	92	76	66
GSA	94	92	76	66
HS	83	64	74	61
PSO	94	92	76	67

The second set of experiments is performed on four different datasets [16]. Also, the proposed method is compared against the other methods of the paper. From Table 7 it is found that a considerable amount of feature reduction is obtained by the proposed method when compared with other methods. However, same number of features are obtained for Hepatitis and Wine datasets.

Table 7. Comparison of features selected using UFS-MBBA for last four datasets.

Algorithm	Dataset			
	Dermatology	Hepatitis	Wine	WDBC
UFS-MBBA	16	5	5	10
UFSACO	25	5	5	5
RSM	25	5	5	5
MC	25	5	5	5
RRFS	25	5	5	5
TV	25	5	5	5
LS	25	5	5	5

Table 8 provides a summary of the classification accuracy obtained with the different classifiers and other feature selection methods. It is noticed that, NB classifier produced high accuracy for Dermatology and Hepatitis datasets. However, for Wine and WDBC datasets SVM classifier obtained high accuracy. Even though, same number of features are selected for Hepatitis and Wine datasets, high classification accuracy is achieved due to the different subset of features selected by the proposed method.

Table 8. Accuracy with different classifiers for last four datasets.

Algorithm	Classifiers	Dataset			
		Dermatology	Hepatitis	Wine	WDBC
UFS-MBBA	Decision Tree	87.43	78.75	94.38	92.97
	Naïve Bayes	96.37	87.5	96.63	93.32
	SVM	94.13	83.75	97.19	94.9
UFSACO	Decision Tree	91.84	78.87	95.08	91.91
	Naïve Bayes	93.92	79.06	90.17	92.42
	SVM	95.28	83.15	95.08	90.72
RSM	Decision Tree	91.60	77.55	86.34	86.34
	Naïve Bayes	94.72	82.64	80.33	86.65
	SVM	94.88	80.94	81.97	83.82
MC	Decision Tree	91.12	83.59	92.35	91.08
	Naïve Bayes	93.92	82.45	92.89	90.93
	SVM	94.56	82.73	89.62	88.97
RRFS	Decision Tree	91.82	76.04	93.99	90.98
	Naïve Bayes	89.44	77.17	95.74	90.05
	SVM	93.44	79.44	93.99	90.36
TV	Decision Tree	89.92	77.17	86.34	92.06
	Naïve Bayes	92.00	80.00	93.44	90.04
	SVM	91.36	79.06	89.07	89.90
LS	Decision Tree	88.24	82.84	90.16	91.86
	Naïve Bayes	91.20	79.06	93.77	90.52
	SVM	91.20	77.74	91.26	89.80

Table 9 provides a summarized view of the best classification accuracy achieved among the different classifiers. It is clearly noticed that UFS-MBBA performs better than the other methods. Also, the improvement percentage of UFS-MBBA with UFSACO in accuracy of Dermatology, Hepatitis, Wine and WDBC found to be 2.45%, 4.35%, 2.11% and 2.48% respectively.

Table 9. Comparison of accuracy with other algorithms for last four datasets.

Algorithm	Dataset			
	Dermatology	Hepatitis	Wine	WDBC
UFS-MBBA	96.37	87.50	97.19	94.90
UFSACO	93.92	83.15	95.08	92.42
RSM	94.88	82.64	86.34	86.65
MC	94.56	83.59	92.89	90.93
RRFS	93.44	79.44	95.74	92.06
TV	92.00	80.00	93.44	92.06
LS	91.20	82.84	93.77	91.86

Figure 1 shows the best accuracy and features obtained with UFS-MBBA for the datasets of Table 2.

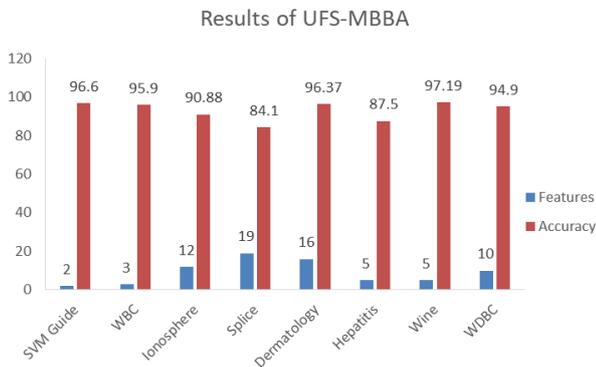


Figure 1. Summarized performance results of UFS-MBBA.

Figure 2 provides the convergence rate of the proposed algorithm. The number of iterations fixed in the experiments is 50 whereas other meta-heuristic algorithms [27] used 100 iterations. On the whole, it is noticed that convergence occurred in 2nd iteration for SVMGuide1 and in 19th iteration for Splice. As the proposed algorithm improves the local search with the mutation operator, it is possible to attain convergence with less number of iterations. Since the number of features of a dataset defines the solution space of the feature selection problem, the convergence rate for the datasets differ from each other.

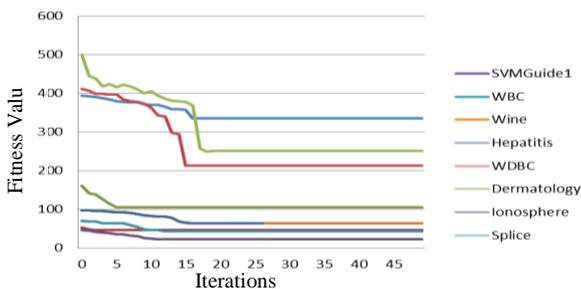


Figure 2. Convergence of UFS-MBBA.

Even though other swarm intelligence algorithms perform a heuristic search to find the solutions (feature subset), our proposed algorithm provides better results, because UFS-MBBA determines a new solution without getting trapped in a local optimum. In summary, UFS-MBBA algorithm produce better results than other algorithms in terms of accuracy and number of features.

6. Conclusions

This work determined the best feature subset using unsupervised method. To accomplish this, we proposed UFS-MBBA algorithm using the clustering procedure as a wrapper. The effectiveness of the proposed method is assessed using DT, NB and SVM classifiers. Moreover, the proposed algorithm is compared with supervised BBA, UFSACO and other unsupervised methods. The experimental results show that it can produce high classification accuracy with minimal number of features. Also, the proposed method is independent of the classifier. In the future work, crossover operator can be combined along with the algorithm to reduce the number of features. Also, other unsupervised learning algorithms can be used to further enhance the performance. Other swarm intelligence algorithms like cuckoo search, shuffled frog leap algorithms can be combined with a bat algorithm for solving the problem.

References

- [1] Bennet J., Ganaprakasam C., and Kumar N., “A hybrid Approach for Gene Selection and Classification using Support Vector Machine,” *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 695-700, 2015.
- [2] Dash M., Choi K., Scheuermann P., and Liu H., “Feature Selection for Clustering-a Filter Solution,” in *Proceedings of IEEE International Conference on Data Mining*, Maebashi, pp. 115-122, 2002.
- [3] Diao R. and Shen Q., “Nature Inspired Feature Selection Metaheuristics,” *Artificial Intelligence Review*, vol. 44, no. 3, pp. 311-340, 2015.
- [4] Dy J. and Brodley C., “Feature Selection for Unsupervised Learning,” *Journal of Machine Learning Research*, vol. 5, pp. 845-889, 2004.
- [5] Frigui H. and Nasraoui O., “Unsupervised Learning of Prototypes and Attribute Weights,” *Pattern Recognition*, vol. 37, no. 3, pp. 567-581, 2004.
- [6] Geem Z., *Music-inspired Harmony Search Algorithm: Theory and Applications*, Springer Publishing Company, 2009.
- [7] Grozavu N., Bennani Y., and Lebbah M., “From Variable Weighting to Cluster Characterization in Topographic Unsupervised Learning,” in

- Proceedings of IEEE International Joint Conference on Neural Network*, Atlanta, pp. 1005-1010, 2009.
- [8] Gu Q., Li Z., and Han J., "Generalized Fisher Score for Feature Selection," in *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, Barcelona, pp. 266-273, 2011.
- [9] Gullo F., Talukder A., Luke S., Domeniconi C., and Tagarelli A., "Multiobjective Optimization of Co-clustering Ensembles," in *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, Philadelphia, pp. 1495-1496, 2012.
- [10] Han J. and Kamber M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2006.
- [11] Hong Y., Kwong S., Chang Y., and Ren Q., "Consensus Unsupervised Feature Ranking from Multiple Views," *Pattern Recognition Letters*, vol. 29, no. 5, pp. 595-602, 2008.
- [12] Hong Y., Kwong S., Chang Y., and Qingsheng R., "Unsupervised Feature Selection using Clustering Ensembles and Population based Incremental Learning Algorithm," *Pattern Recognition*, vol. 41, no. 9, pp. 2742-2756, 2008.
- [13] Huang J., Ng M., Rong H., and Li Z., "Automated Variable Weighting in K-means Type Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 657-668, 2005.
- [14] Inbarani H., Banu P., and Azar A., "Feature Selection using Swarm-based Relative Reduct Technique for Fetal Heart Rate," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 793-806, 2014.
- [15] Jing L., Ng M., and Huang J., "An Entropy Weighting k-means Algorithm for Subspace Clustering of High-dimensional Sparse Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1026-1041, 2007.
- [16] Kaur P. and Kaur T., "A comparative Study of Various Metaheuristics Algorithm," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 5, pp. 6701-6704, 2014.
- [17] Kluger Y., Basri R., Chang J., and Gerstein M., "Spectral Biclustering of Microarray Cancer Data: Co-clustering Genes and Conditions," *Genome Research*, vol. 13, no. 4, pp. 703-716, 2003.
- [18] Lai C., Reinders M., and Wessels L., "Random Subspace Method for Multivariate Feature Selection," *Pattern Recognition Letters*, vol. 27, no. 10, pp. 1067-1076, 2006.
- [19] Li Y., Dong M., and Hua J., "Localized Feature Selection for Clustering," *Pattern Recognition Letters*, vol. 29, no. 1, pp. 10-18, 2008.
- [20] Liu H. and Yu L., "Toward Integrating Feature Selection Algorithms for Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491-502, 2005.
- [21] Mirjalili S., Mirjalili S., and Yang X., "Binary Bat Algorithm," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 663-681, 2014.
- [22] Mitra P., Murthy C., and Pal S., "Unsupervised Feature Selection using Feature Similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301-312, 2002.
- [23] Morita M., Sabourin R., Bortolozzi F., and Suen C., "Unsupervised Feature Selection using Multi-objective Genetic Algorithms for Handwritten Word Recognition," in *Proceedings of 7th International Conference on Document Analysis and Recognition*, Edinburgh, pp. 666-670, 2003.
- [24] Rashedi E., Nezamabadi-pour H., and Saryazdi S., "BGSA: Binary Gravitational Search Algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727-745, 2009.
- [25] Rodrigues D., Pereira L., Nakamura R., Costa K., Yang X., Souza A., and Papa J., "A wrapper Approach for Feature Selection based on Bat Algorithm and Optimum-Path Forest," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2250-2258, 2014.
- [26] Saeys Y., Inza I., and Larranaga P., "A Review of Feature Selection Techniques in Bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507-2517, 2007.
- [27] Saxena A., Pal N., and Vora M., "Evolutionary Methods for Unsupervised Feature Selection Using Sammon's Stress Function," *Fuzzy Information and Engineering*, vol. 2, no. 3, pp. 229-247, 2010.
- [28] Shamsinejadbabki P. and Saraee M., "A new Unsupervised Feature Selection Method for Text Clustering based on Genetic Algorithms," *Intelligent Information Systems*, vol. 38, no. 3, pp. 669-684, 2012.
- [29] Shi T. and Horvath S., "Unsupervised Learning with Random Forest Predictors," *Journal of Computational and Graphical Statistics*, vol. 15, no. 1, pp. 118-138, 2006.
- [30] Tabakhi S., Moradi P., and Akhlaghian F., "An Unsupervised Feature Selection Algorithm based on Ant Colony Optimization," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 112-12, 2014.
- [31] Velayutham C. and Thangavel K., "Unsupervised Quick Reduct Algorithm Using Rough Set Theory," *Journal of Electronic Science and Technology*, vol. 9, no. 3, pp. 193-201, 2011.
- [32] Weka, available at: <http://www.cs.waikato.ac.nz/ml/weka/>, Last Visited, 2015.

- [33] Yang X., "Swarm Intelligence based Algorithms: a Critical Analysis," *Evolutionary Intelligence*, vol. 7, no. 1, pp. 17-28, 2014.
- [34] Yang X., "Bat algorithm; Literature Review and Applications," *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, pp. 141-149, 2013.
- [35] Yang X., "Firefly algorithm Stochastic Test Functions and Design Optimization," *International Journal of Bio-inspired Computing*, vol. 2, no. 2, pp. 78-84, 2011.
- [36] Zhuo L., Zheng J., Wang F., Li X., Ai B., and Qian J., "A Genetic Algorithm based Wrapper Feature Selection Method for Classification of Hyperspectral Images using Support Vector Machine," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 397-402, 2008.



Rajalaxmi Ramasamy received BE [CSE] and ME [CSE] from Bharathiar University during 1990 and 2001 respectively. She completed her Ph.D in Anna university, India during 2011. She is working as Professor in CSE, Kongu Engineering College, India. Her research interests include Data Mining, Nature Inspired Computing and Big Data Analytics.



Sylvia Rani completed B.E [CSE] during 2009. She obtained M.E [CSE] from Kongu Engineering College, Tamilnadu, India in 2015. She has published two papers in national and international conference. Her research interests include Nature inspired computing and Parallel Processing.