

# Hybridized Metaheuristic Algorithm for Feature Selection Guided with Naïve Bayes Classifier

Pallavi<sup>1</sup>, Smt. T Jayakumari<sup>2</sup>

<sup>1</sup>M.Tech Dept of Computer Science and Engineering, BTL Institute of Technology, Bangalore, India

<sup>2</sup>Assistant professor Dept of Computer Science and Engineering, BTL Institute of Technology, Bangalore, India

**Abstract**-Feature selection is a problem of discovering efficient features among all the features in which the final feature set can improve the accuracy and reduce complexity. This is very essential due to rapid escalation in amount of data and information for every 20 months. In some cases, too many redundant or irrelevant features may defeat main features for classification. Feature selection can remedy this problem and therefore improve the prediction accuracy and reduce the computational overhead of classification algorithms. This feature selection methods can be used in many fields like pattern recognition, machine learning, signal processing. Irrelevant features do not contribute to the predictive accuracy, and redundant features do not contribute to getting a better predictor for that they provide most information which is already present in other feature(s). Many feature selection methods have been proposed and studied for machine learning applications. The proposed hybridised metaheuristic algorithm for feature selection guided with Naïve Bayes classifier will select minimum number of relevant features in order to maintain the classification accuracy. This feature selection method is compared against other two algorithms such as Exhaustive Search and Genetic Search. This work focuses on three perspectives: Number of features, classification accuracy and generalization with multiclass dataset. Results shows that BANB outperforms against other two feature selection algorithms in selecting lower number of features by removing irrelevant, redundant, or noisy features to maintain highest classification accuracy.

**Keywords:** No. of Bats, Fitness function, attribute evaluator, local search, feature selection.

**GA**-Genetic Algorithm, **BANB**- Bat Algorithm with Naïve Bayes

## I INTRODUCTION

The importance of data mining technique is to mine the information from massive data set and make over it into a reasonable form for supplementary purpose. Classification is a significant task in data analysis and data mining applications. The enthusiasms to perform feature selection in classification experiments are two-fold. First is to enrich the classifier performance by selecting only useful features and removing irrelevant, redundant, or noisy features. Second is to reduce the number of features when classification algorithm could not scale up to the size of feature set either in space or time. In general, there are two essential steps in feature selection: (i) searching for desired subset using some search strategies. (ii) evaluating the subset produced.

Search strategy may be exhaustive or approximate. Exhaustive search strategy evaluates all probabilities of the feature subset, while approximate search strategy generates only high quality solutions with no guarantee of finding a global optimal solution [1].

Branch and bound method [2] is one of the most prominent algorithm in exhaustive search. Exhaustive search guarantees optimal solution but this method is not practical for even a medium-sized dataset in finding the optimal subset of features, which is an NP-hard problem [3]. For N number of features, the number of possible solutions will be exponential to  $2^n$ . Hence research and efforts have been shifted to metaheuristic algorithms, which are considered as a subclass of approximate methods. The fiction has shown that the metaheuristic algorithms are proficient in handling large size problem instances with acceptable solutions within a judicious time [4–7]. Hybridized metaheuristic is used for feature selection and Naïve Bayes classifier is used for classification accuracy with multi class in this work.

The major challenge in feature selection is to select the minimal set of features with modicum or no loss of classification accuracy. While literature has shown abundant developments in achieving this [8 - 10]. The basis of comparison is rather limited in terms of number of features, classification accuracy, and generalization in isolation. The importance of generalization is to investigate their effect on the performance of different classifiers.

The Objectives of this work is as follows: first is to design a Hybridized Metaheuristic Algorithm (BA) for feature selection which exploits Naïve Bayes classifier to escort algorithm with highest classification accuracy, second to evaluate the performance of the proposed hybrid algorithm against other well-known feature selection algorithms in terms of feature selection and third in terms of generalization with multi class dataset, compare the accuracy of all three algorithms with Naïve Bayes classifiers using Weka Tool.

## II RELATED WORK

The applications of metaheuristic algorithms has shown high effectiveness as well as efficiency to solve complex and large problems in feature selection.

Generally, there are two categories of metaheuristic algorithms: (i) single solution-based metaheuristic

algorithm (SBM) which manipulate and transform single solution during the search.(ii) population-based metaheuristic algorithm (PBM) in which whole population of solution is evolved.

Hill Climbing (HC) algorithm [1,11] is one of the simplest and oldest SBM method used in feature selection. In order to improve the quality of solution This algorithm starts with a random initial solution and swaps the current solution with a neighbouring solution in the following iteration. When all the neighbouring candidate subsets are poorer than the current solution then searching will stop, which means that the algorithm will be most probably trapped in local optimum. Simulated Annealing (SA) is proposed [10]In order to overcome this problem. SA accepts the worse moves that commensurate to the parameter determined at the initial stage, called the temperature, which is inversely proportional to the change of the fitness function. Modified SA algorithm called the Great Deluge Algorithm (GDA) is proposed [12] in more recent work to provide a deterministic acceptance function of the neighbouring solutions. Tabu Search (TS) also accepts non improving solutions to escape from local optima. TS stores in formation related to the search process, which is a list of all previous solutions or moves in what is termed as Tabu list [13, 14]. Nonetheless, Hill Climbing and Simulated Annealing suffer from two major disadvantages. First, they often converge towards local optima and second they can be very sensitive to the initial solution [1].

Different from SBM, PBM signifies an iterative improvement in a population of solutions that works as follows. Firstly, the population is initialized then a new population of solution is generated. Next, the new population is integrated into the existing one by using some selection procedures. When certain criterion is satisfied the search process will terminate. Genetic Algorithm (GA) [5, 15, 16] is one of the most prominent and oldest population-based solution used in features selection. Crossover and mutation operations are the major roles in GA which used to couple solutions and to arbitrarily adjust the individual content, to boost diversity aiming to decrease the risk of sticking in local optima.

Ant Colony Optimization (ACO) is another PBM algorithm, which takes form as a multi agent system, building unit of this system represents virtual ants as inspired by the behaviour of real ants. In nature, a chemical trace called pheromone is left on the ground and is used to guide a group of ants heading for the target point since ants are distinguished to see very well [6, 17, 18]. Particle Swarm Optimization (PSO) is another nature-inspired algorithm which simulates the social behaviour of natural creatures such as bird flocking and fish schooling to discover a place with adequate food [7,19]. Scatter Search (SS) is another PBM method that recombines solution selected from a reference set to generate other solutions by building an initial population satisfactory to the criteria of quality and diversity [20].

Once feature subset have been searched, each candidate from the resulting subset generated needs to be evaluated based on some predetermined assessment criteria.

Embedded, Wrapper, Filter, and Hybrid methods are four methods of feature subset evaluations depending on how the searching strategy is being associated with the classification model. Embedded methods incorporate feature selection as a part of the training process and are usually specific to given learning algorithms, and therefore may be more efficient than the other three categories. Wrapper methods use a predictive model to score feature subsets. Each new subset is used to train a model, which is tested on a hold-out set. Filter methods use a proxy measure instead of the error rate to score a feature subset. This measure is chosen to be fast to compute, whilst still capturing the usefulness of the feature set. Filters are usually less computationally intensive than wrappers, but they produce a feature set which is not tuned to a specific type of predictive model. Hybrid methods are a combination of filter and wrapper methods by using a filter method to reduce search space that will be considered by the subsequent wrapper. Thus, we will focus on the Hybrid method in this paper.

### III PROPOSED HYBRIDIZEDMETAHEURISTIC ALGORITHM WITH NAÏVE BAYES CLASSIFIER (BANB):

**Naïve Bayes Classifier (NB):** Naïve Bayes classifier is an attribute evaluator used to classify the features based on the conditional probability. NB is considered as simple classifier based on “Bayes Theorem”. The Bayesian algorithm is trademarked as “Naive” because it is founded on Bayes Rule, which has a strong hypothesis that the features are conditionally independent from each other with respect to the class [21]. This classifier belongs to wrapper method. NB has proven its effectiveness in various domains such as text classification, improving search engine quality, image processing, fault prediction, and medical diagnoses.

Naive Bayes classifier works as follows: let U be a vector of random variables denoting the observed attribute values in the training set  $U = [u_1, u_2, u_3 \dots, u_n]$  to certain class label C in the training set. The probability of each class given the vector of observed values for the predictive attributes can be computed using the following formula 1:

$$P(V_j|U) = \frac{P(U_j) P(U|V_j)}{\sum_{i=1}^C P(V_i) P(U|V_i)}, j=1, \dots, C \tag{1}$$

Where  $P(U_j)$  is the prior probability of class  $U_j$ ,  $P(U|V_j)$  is the posterior probability of U conditioned on  $V_j$ ,  $P(V_i)$  is the prior probability of class  $V_i$ ,  $P(U|V_i)$  is the posterior probability of U conditioned on  $V_i$  and  $P(V_j|U)$  is the class conditional probability density function. Basically put, the conditional independence assumption assumes that each variable in the dataset is conditionally independent of the other. This is simple to compute for test cases and to estimate from training data as follows in formula (2):

$$P(U/V_j) = \prod_{i=1}^n P(U_i / V_j) \quad j=1,2,3,\dots,C \tag{2}$$

where  $U_i$  is the value of the  $i^{th}$  attribute in  $U$  and  $n$  is the number of attributes.  $V_j$  is the value of  $j^{th}$  attribute in  $V$ . The probability distribution over the set of features is calculated using the following formula (3):

$$P(u) = \prod_{i=1}^k P(C_i) P(U/C_i) \tag{3}$$

where  $k$  be the number of classes, and  $C_i$  is the  $i^{th}$  class. Following are the several characteristics of Naïve Bayes classifier: (i) Computational efficiency is high compared to other wrapper methods because it is inexpensive since it is considered linear time complexity classifier. (ii) Noise, missing values can be handled in the dataset with high capability. (iii) Low variance with less searching.

**Bat Algorithm (BA):** Bat algorithm is based on the echolocation behavior of microbats with varying pulse rates of emission and loudness. BA was first presented in [22] and it was found to outperform Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) in evaluation using benchmark functions. BA has also been successfully applied to tough optimization problems such as motor wheel optimization problems, clustering problems, global engineering optimizations, and constrained optimization tasks.

The idealization behind echolocation of microbats can be summarized as follows: Each virtual bat flies randomly with a velocity  $V_i$  at position (solution)  $x_i$  with a varying frequency  $[f_{min}, f_{max}]$  or wavelength  $[\lambda_{min}, \lambda_{max}]$  and loudness  $A_i$ . Loudness can be different in many ways, the loudness differs from a large (positive)  $A_0$  to a minimum constant value  $A_{min}$ . As it searches and finds its prey. Once prey is found it changes frequency, loudness and pulse emission rate where  $r_i \in [0, 1]$ . Search is intensified by a local random walk. Selection of the best bat continues until certain stopping conditions are met. This essentially uses a frequency-tuning technique to control the dynamic behavior of a swarm of bats, and the balance between exploration and exploitation can be controlled by tuning algorithm-dependent parameters in bat algorithm.

**A. Parameter Description:**

In this work, the balance between exploration and exploitation can be controlled by tuning algorithm-dependent parameters. Following are the parameters used in Hybrid Bat Algorithm:

**Frequency (f):**

Frequency is represented as real number, defined in formula (4). The maximum and minimum frequency can be chosen based on the application domain,  $\beta \in [0, 1]$  is a random number.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{4}$$

**Velocity (V):**

Velocity suggests the number of bat attributes that should change at a certain amount of time defined in formula (5) and represented by positive integer number. The bats communicate with each other through the global best solution and move towards the global best solution.

$$V_i^t = V^{t-1}_i + (x^* - x_i^t)f_i \tag{5}$$

where  $V_i^t$  is the velocity of  $i^{th}$  bat,  $V^{t-1}_i$  is the previous velocity of  $i^{th}$  bat,  $(x^* - x_i^t)$  is the difference between the length of global best bat and length of the  $i^{th}$  bat where length is the number of selected features,  $f_i$  is the frequency. In  $(x^* - x_i^t)$ , if the difference is positive the global best bat has more features than those of  $i^{th}$  bat, with reference to formula (5) when the result is summed with previous velocity, it will accelerate the  $i^{th}$  bat towards the global best bat. If the difference is negative, then the  $i^{th}$  bat has more features than that of the global best bat. Therefore, with reference to formula (5) when the result is summed with previous velocity, it will decrease the velocity of  $i^{th}$  bat and help to attract towards the global best bat. In proposed algorithm maximum velocity ( $V_{max}$ ) will set to  $(1/3) * N$ , where  $N$  is the number of features.

**Bat Position (x):**

Each bat position is encoded as binary string with length  $N$ , where  $N$  is the total number of features. Each bit encodes a single feature. As shown in Fig (I), bit “1” implies the corresponding feature is selected and “0” implies corresponding feature is not selected.

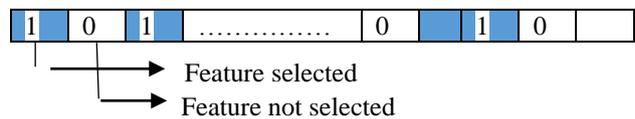


Fig (I): Bat position encoding as a binary bit string

The positions are categorized into two groups according to bit difference between  $i^{th}$  bat and global best bat in order to align exploration and exploitation during searching. From formula (6), the bat’s position is adjusted depending on one of the following condition:

- i) Velocity of  $i^{th}$  bat is lower or equal to the number of different bits, then  $i^{th}$  bat will copy some of the features from global best bat, thus moving towards global best bat, while still exploring new search space.
- ii) Velocity of the  $i^{th}$  bat is higher than the velocity of global best bat, then  $i^{th}$  bat will import all the features from global best bat to be same as the global best bat with a few different bits to facilitate further exploitation.

$$x_i^t = x^{t-1}_i + v^t_i \tag{6}$$

where  $x_i^t$  is the  $i^{th}$  bat position,  $x^{t-1}_i$  is the previous  $i^{th}$  bat position,  $v^t_i$  is the velocity of the  $i^{th}$  bat,  $t$  is the time.

**Loudness (A<sub>i</sub>):**

Loudness is the change in number of features at certain time during local search around the global best bat, as well as local search around the  $i^{th}$  bat. The formula for the loudness is shown in formula (7).

$$x_{new} = x_{old} + \epsilon A^t \tag{7}$$

where  $x_{new}$  is the position of new bat,  $x_{old}$  is the position of old bat (previous local search position of same bat),  $A^t$  is the average loudness of all the bats at certain iteration,  $\epsilon \in [-1, 1]$  is a random number. The value for sound loudness ( $A$ ) ranges between the maximum and minimum loudness.

When bat starts approaching the best solution the loudness value will decrease. The following formula (8) shows that the amount of decrease is determined by  $\alpha$ :

$$A^{t+1}_i = \alpha A^t_i \tag{8}$$

where  $A^{t+1}$  is the loudness of new bat solution,  $\alpha$  is constant value,  $A^t$  is the loudness of old bat solution. The maximum and minimum loudness value is determined based on domain of application and size of the dataset. In proposed algorithm, the maximum loudness has been determined empirically as  $(1/5) * N$ , where  $N$  is the number of features in certain dataset.

**Pulse Emission ( $r_i$ ):**

The role of  $r_i$  is to decide whether local search around the global best bat solution should be skipped or otherwise. If pulse rate is high then the amount of conducting local search will be reduced. Therefore when the bat approaches best solution, pulse rate value will increase and subsequently reduce the chances to conduct a local search around the global best. The following formula (9) shows that the amount of increase is determined by  $\gamma$ .

$$r^{t+1}_i = r^0_i [1 - \exp(-\gamma t)] \tag{9}$$

where  $r^{t+1}_i$  is the pulse rate of new bat solution,  $r^0_i$  is the pulse rate of previous bat solution,  $t$  is the time,  $\gamma$  is the constants value.

**Fitness Function ( $F_i$ ):**

Fitness function as shown in formula (10) can be used by each candidate solution, where  $P (V_j / U)$  is the classification accuracy from Naïve Bayes classifier (1),  $TF$  is the total number of all features and  $SF$  is the selected features from the dataset.  $\delta \in [0, 1]$  and  $\phi = 1 - \delta$ . Classification accuracy is having more importance because it gives more weight than size of the subset.

$$F_i = \delta \cdot P (V_j / U) + \phi \cdot \frac{TF - SF}{TF} \tag{10}$$

**IV BANB ALGORITHM FLOW DIAGRAM**

The irrelevant features, along with redundant features, severely affect the accuracy of the learning machines. Thus, feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. Keeping these in mind, in future this work develops a Hybrid Bat algorithm with Naïve Bayes classifier to deal with both irrelevant and redundant features in the dataset, and obtain a good feature subset with more classification accuracy compared to other feature selection algorithms as well as with other classifiers.

**A. Flow Diagram:**

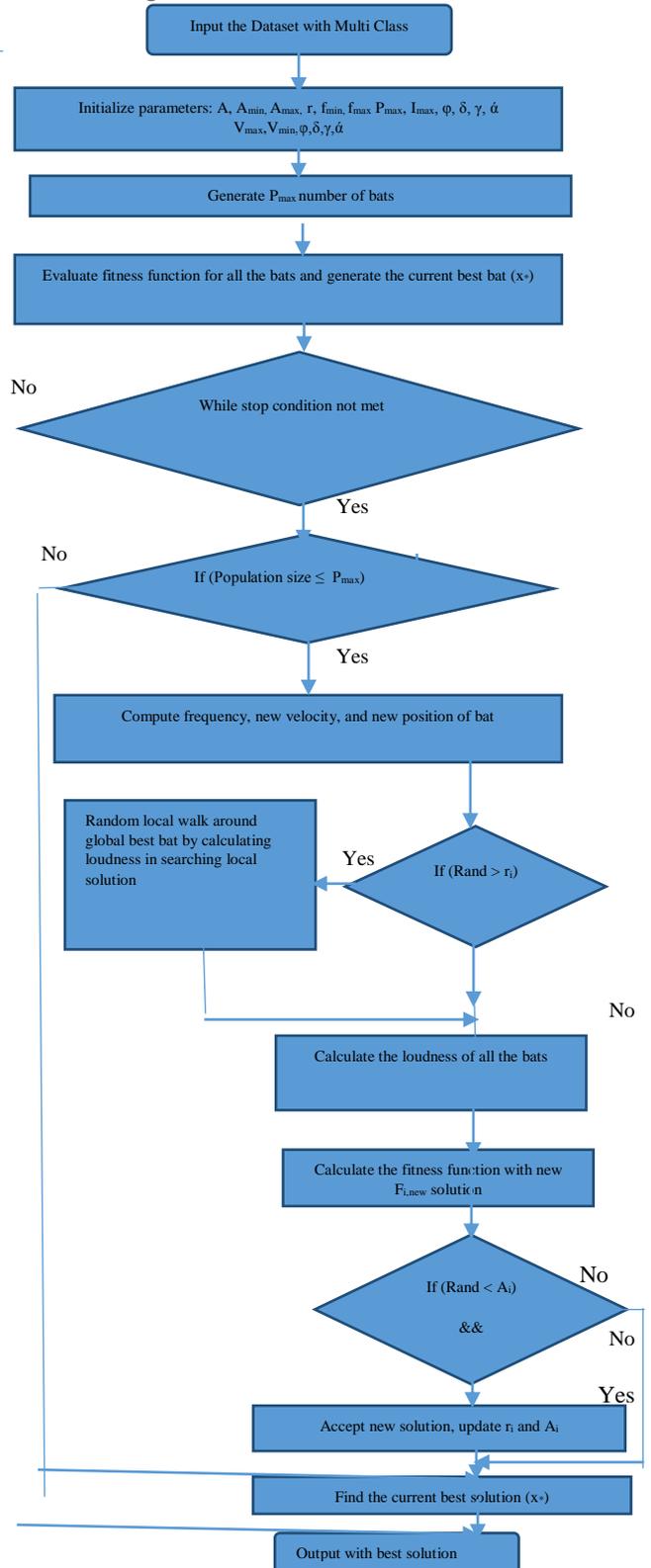


Fig (II): Hybrid Metaheuristic algorithm with Naïve Bayes

**V RESULTS AND DISCUSSIONS**

The results of proposed BANB is compared against other two feature selection algorithms such as Exhaustive

Search and Genetic Search in terms of number of relevant features selected from the original dataset and classification accuracy obtained from Naïve Bayes classifier using Weka Tool.

#### A. Dataset Description:

This work explores an experimental study on Process Glass dataset with Multi Class (11 attributes, 214 instances) [23], taken from UCI repository for feature selection.

#### B. Performance Evaluation:

The proposed Hybridised Metaheuristic algorithm (BA) is implemented in Net beans IDE. With reference to the base paper [24] and Flow Diagram Fig (II), the experiment was employed with the population size ( $P_{max}$ ) of 5, where decrease loudness and increase pulse rate were set to 0.6. The initial value of pulse rate set to 0.2. The value of  $\delta$  set to 0.9,  $\phi$  value set to 0.1. The  $f_{max}$  and  $f_{min}$  were set to 3 kHz, 10 kHz respectively. The  $A_{max}$  and  $A_{min}$  value were set to 3, 1 respectively.

Following Table I provides the comparison results.

**Table I**

Feature Selection Techniques	Number of Attributes Selected	Naïve Bayes Classification Accuracy
BANB	7	99.53%
Exhaustive Search	8	98.12%
Genetic Search	10	99.06%

From Table I, BANB selects minimal number of relevant features compared to other two algorithms with highest classification accuracy.

## VI CONCLUSION

In this paper, a Hybridized Metaheuristic algorithm for Feature selection was presented with multi class dataset. The Hybrid Bat Algorithm employed with Naïve Bayes Classifier to intelligently select most efficient and relevant features that could maximize the classification accuracy while ignoring the irrelevant features and noisy features. From the experiments, this work concluded that the proposed Metaheuristic Bat Algorithm with Naive Bayes Classifier outperformed other two algorithms with selection of minimal number of relevant features and highest classification accuracy. For the future work, we plan to explore with very high-dimensional datasets and study some formal properties of feature space.

## REFERENCES

- [1] E.G.Talbi, Metaheuristics: From Design to Implementation, John Wiley and Sons, 2009.
- [2] B. Yu and B. Yuan, "A more efficient branch and bound algorithm for feature selection," *Pattern Recognition*, vol. 26, no. 6, pp. 883–889, 1993.
- [3] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 1-2, pp. 237–260, 1998.
- [4] R. Jensen and Q. Shen, "Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches," *IEEE*

- Transactions on Knowledge and Data Engineering, vol. 16, no. 12, pp. 1457–1471, 2004.
- [5] J. Yang and V. Honavar, "Featuresubsetselectionusinggeneticalgorithm," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 2, pp. 44–48, 1998.
- [6] L. Ke, Z. Feng, and Z. Ren, "Anefficientantcolonyoptimization approach to attribute reduction in rough set theory," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1351–1357, 2008.
- [7] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Featureselection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.
- [8] M. Anbarasi, E. Anupriya, and N. Iyengar, "Enhanced prediction of heart disease with featuresubsetselectionusinggenetic algorithm," *International Journal of Engineering Science and Technology*, vol. 2, pp. 5370–5376, 2010.
- [9] H. Vafaie and I. F. Imam, "Feature selection methods: genetic algorithms vs greedy-like search," in *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems Louisville, 1994*.
- [10] J. C. W. Debusse and V. J. Rayward Smith, "Featuresubsetselectionwithina simulated annealing data mining algorithm," *Journal of Intelligent Information Systems*, vol. 9, no. 1, pp. 57–81, 1997.
- [11] R. Caruana and D. Freitag, "Greedy attributes selection," in *Proceedings of the 11th International Conference on Machine Learning*, pp. 28–36.
- [12] N. S. Jaddani and S. Abdullah, "Nonlinear great deluge algorithm for rough set attribute reduction," *Journal of Information Science and Engineering*, vol. 29, pp. 49–62, 2013.
- [13] M. A. Tahir, A. Bouridane, and F. Kurugollu, "Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 438–446, 2007.
- [14] A. R. Hedar, J. Wang, and M. Fukushima, "Tabu search for attribute reduction in rough set theory," *Soft Computing*, vol. 12, no. 9, pp. 909–918, 2008.
- [15] A. Jain and D. Zongker, "Featureselection: evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.
- [16] R. Leardi, "Application of a genetic algorithm to feature selection under full validation conditions and to outlier detection," *Journal of Chemometrics*, vol. 8, pp. 65–79, 1994.
- [17] R. Jensen and Q. Shen, "Finding rough set reducts with Ant colony optimization," in *Proceedings of the UK Workshop on Computational Intelligence*, pp. 15–22, 2003.
- [18] R. K. Sivagaminathan and S. Ramakrishnan, "A hybrid approach for feature subset selection using neural networks and ant colony optimization," *Expert Systems with Applications*, vol. 33, no. 1, pp. 49–60, 2007.
- [19] E. G. Talbi, L. Jourdan, J. Garcia Nieto, and E. Alba, "Comparison of population based metaheuristics for feature selection: application to microarray data classification," in *Proceedings of the 6th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA'08)*, pp. 4552, April 2008.
- [20] J. Wang, A. R. Hedar, G. Zheng, and S. Wang, "Scatter search for rough set attribute reduction," in *Proceedings of the International Joint Conference on Computational Sciences and Optimization (CSO'09)*, pp. 531–535, chn, April 2009.
- [21] G. I. Webb, "Naïve bayes," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., pp. 713–714, Springer, New York, NY, USA, 2010.
- [22] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies For Optimization (NICSO'10)*, J. Gonzalez, D. Pelta, C. Cruz, G. Terrazas, and N. Rasnigor, Eds., pp. 65–74, Springer, Berlin, Germany, 2010.
- [23] UCI Repository <https://archive.ics.uci.edu/ml/>.
- [24] X. S. Yang, "Naïve Bayes Guided Bat algorithm for feature selection." *The Scientific World Journal* Volume 2013, Article ID 325973, <http://dx.doi.org/10.1155/2013/325973>.