Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 325973, 9 pages
http://dx.doi.org/10.1155/2013/325973

Research Article

Naive Bayes-Guided Bat Algorithm for Feature Selection

Ahmed Majid Taha,[1] Aida Mustapha,[2] and Soong-Der Chen[3]

[1] *College of Graduate Studies, Universiti Tenaga Nasional, 43000 Kajang, Selangor, Malaysia*
[2] *Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*
[3] *College of Information Technology, Universiti Tenaga Nasional, 43000 Kajang, Selangor, Malaysia*

Correspondence should be addressed to Ahmed Majid Taha; ahmed.majid.taha@gmail.com

Received 30 September 2013; Accepted 14 November 2013

Academic Editors: Z. Cui and X. Yang

When the amount of data and information is said to double in every 20 months or so, feature selection has become highly important and beneficial. Further improvements in feature selection will positively affect a wide array of applications in fields such as pattern recognition, machine learning, or signal processing. Bio-inspired method called Bat Algorithm hybridized with a Naive Bayes classifier has been presented in this work. The performance of the proposed feature selection algorithm was investigated using twelve benchmark datasets from different domains and was compared to three other well-known feature selection algorithms. Discussion focused on four perspectives: number of features, classification accuracy, stability, and feature generalization. The results showed that BANB significantly outperformed other algorithms in selecting lower number of features, hence removing irrelevant, redundant, or noisy features while maintaining the classification accuracy. BANB is also proven to be more stable than other methods and is capable of producing more general feature subsets.

1. Introduction

The motivations to perform feature selection in a classification experiment are two-fold. The first is to enhance the classifier performance by selecting only useful features and removing irrelevant, redundant, or noisy features. The second is to reduce the number of features when classification algorithms could not scale up to the size of feature set either in time or space. In general, feature selection consists of two essential steps, which are searching for desired subset using some search strategies and evaluating the subset produced. A search strategy in searching the feature subset could be exhaustive or approximate. While exhaustive search strategy evaluates all probabilities of the feature subset, approximate search strategy only generates high quality solutions with no guarantee of finding a global optimal solution [1].

One of the most prominent algorithms in exhaustive search is branch and bound method [2]. Exhaustive search guarantees optimal solution but this method is not practical for even a medium-sized dataset as finding the optimal subset of features is an NP-hard problem [3]. For N number of features, the number of possible solutions will be exponential to 2^N. Since exhaustive search is not practical, research effort and focus on search strategies have since shifted to metaheuristic algorithms, which are considered as a subclass of approximate methods. The literature has shown that the metaheuristic algorithms are capable of handling large-size problem instances with satisfactory solutions within a reasonable time [4–7].

After searching for feature subset, each candidate from the resulting subset generated needs to be evaluated based on some predetermined assessment criteria. There are three categories of feature subset evaluations depending on how the searching strategy is being associated with the classification model, whether as filter, wrapper, or embedded methods. These three categories will be explained in more detail in the next section.

Nonetheless, the main challenge in feature selection is to select the minimal subset of features with modicum or no loss of classification accuracy. While the literature has shown numerous developments in achieving this [8–10], the basis of comparison is rather limited in terms of number of features, classification accuracy, stability, or feature generalization in

isolation. Generalization of the produced features is important to investigate their effect on the performance of different classifiers.

In view of this, the objectives of this paper are as follows: first to design a new hybrid algorithm that exploits a Naive Bayes algorithm to guide a Bat Algorithm, second to evaluate the performance of the proposed hybrid algorithm against other well-known feature selection algorithms, and third to test the effect of the resulting features in terms of generalization using three different classifiers. The remainder of this paper is organized as follows. Section 2 reviews the related works on searching and evaluating algorithms in feature selection. Section 3 details out the principles of Naive Bayes algorithms, Section 4 presents mechanics of the Bat Algorithm, and Section 5 introduces the proposed Naïve Bayes-guided Bat Algorithm for feature selection. Next, Section 6 describes the experimental settings, Section 7 discusses implications of the results, and, finally, Section 8 concludes with some recommendations for future work.

2. Related Work

The application of metaheuristics algorithms in searching the feature subset has shown high effectiveness as well as efficiency to solve complex and large problems in feature selection. In general, there are two categories of metaheuristic search algorithms: single solution-based metaheuristics (SBM) that manipulate and transform a single solution during the search and population-based metaheuristics (PBM) where a whole population of solutions is evolved. The simplest and oldest SBM method used in feature selection is Hill Climbing (HC) algorithm [1, 11]. This algorithm starts with a random initial solution and swaps the current solution with a neighboring solution in the following iteration in order to improve the quality of solution. Searching will stop only when all the neighboring candidate subsets are poorer than the current solution, which means that the algorithm will be most probably trapped in local optimum [4].

In order to overcome this problem, Simulated Annealing (SA) is proposed [10]. 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. In more recent work, a modified SA algorithm called the Great Deluge Algorithm (GDA) is proposed [12] to provide a deterministic acceptance function of the neighboring solutions. Tabu Search (TS) also accepts nonimproving solutions to escape from local optima. TS stores information 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, SBM algorithms such as 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].

The PBM methods have been equally explored in feature selection. Different from SBM, PBM represents an iterative improvement in a population of solutions that works as follows. Firstly, the population is initialized. Then, a new population of solutions is generated. Next, the new population is integrated into the existing one by using some selection procedures. The search process is terminated when a certain criterion is satisfied. The most prominent and oldest population-based solution used in feature selection is Genetic Algorithm (GA) [5, 15, 16]. The major roles in GA are the crossover and mutation operations used to couple solutions and to arbitrarily adjust the individual content, to boost diversity aiming to decrease the risk of sticking in local optima.

Another PBM algorithm is the Ant Colony Optimization (ACO), which takes form as a multiagent system, whereby the building unit of this system represents virtual ants as inspired by the behavior 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 not able to see very well [6, 17, 18]. Another nature-inspired algorithm is the Particle Swarm Optimization (PSO) algorithm that simulates the social behavior 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 solutions elected from a reference set to generate other solutions by building an initial population satisfactory to the criteria of quality and diversity [20].

The next step in feature selection is evaluating the feature subset produced. The evaluation methods can be broadly classified into three categories. First, the filter approach or independent approach evaluates candidate solutions by depending on intrinsic characteristics of the features themselves, without considering any mining algorithm. Filter approach includes several types such as distance [21], information [22], dependency [23], or consistency [24]. Second, the wrapper approach or dependent approach requires one predetermined learning model and selects features with the purpose of improving the generalization performance of that particular learning model [13]. Although the wrapper approach is known to outperform the filter approach with regard to prediction accuracy [25], the method is time-consuming. Third, the embedded approach in feature evaluation attempts to capitalize on advantages of both approaches by implementing the diverse evaluation criteria in different search phases [26]. By integrating the two approaches at different phases, the embedded approach is capable to achieve accuracy of a wrapper approach at the speed of a filter approach. Choosing an evaluation method for particular search method is a critical mission because the interaction between the evaluation method and the search strategy will affect the overall quality of solution.

3. Naive Bayes Algorithm

Naive Bayes (NB) algorithm is one of the most effective and efficient inductive learning algorithms for data mining along with machine learning. This algorithm belongs to the wrapper approach. NB is considered a simple classifier based on the classical statistical theory "Bayes theorem." The Bayesian algorithm is branded "naïve" because it is founded on Bayes Rule, which has a strong supposition that the features are

conditionally independent from each other with regard to the class [27]. In the literature, the NB algorithm has proven its effectiveness in various domains such as text classification [28], improving search engine quality [29], image processing [30, 31], fault prediction [32], and medical diagnoses [8].

Naive Bayes classifier works as follows: let X be a vector of random variables denoting the observed attribute values in the training set $X = [x_1, x_2, \ldots, x_n]$ 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:

$$P\left(Y_j \mid X\right) = \frac{P\left(Y_j\right) P\left(X \mid Y_j\right)}{\sum_{i=1}^{c} P\left(Y_i\right) P\left(X \mid Y_i\right)}, \quad j = 1, \ldots, c, \tag{1}$$

where $P(Y_i)$ is the prior probability of class Y_i and $P(Y_j \mid X)$ is the class conditional probability density functions. 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:

$$P\left(X \mid Y_j\right) = \prod_{i=1}^{n} P\left(X_i \mid Y_j\right), \quad j = 1, \ldots, c, \tag{2}$$

where X_i is the value of the ith attribute in X and n is the number of attributes. Let k be the number of classes, and c_i is the ith class; the probability distribution over the set of features is calculated using the following equation:

$$P\left(x\right) = \prod_{i=1}^{k} p\left(c_i\right) p\left(X \mid c_i\right). \tag{3}$$

Effectiveness of Naive Bayes algorithm in classification and learning is attributed to several characteristics such as the following [27].

(i) High computational efficiency as compared to other wrapper methods because it is inexpensive since it is considered linear time complexity classifier.

(ii) Low variance due to less searching.

(iii) Incremental learning because NB functions work from approximation of low-order probabilities that are deduced from the training data. Hence, these can be quickly updated as new training data are obtained.

(iv) High capability to handle noise in the dataset.

(v) High capability to handle missing values in the dataset.

Furthermore, NB implementation has no required adjusting parameters or domain knowledge. The major drawback of NB only lies in the assumption of features independence [33]. Despite this, NB often delivers competitive classification accuracy and is widely applied in practice especially as benchmark results. Good survey on the variety of adaptations to NB in the literature can be found in [33].

4. Bat Algorithm

The idea of the Bat Algorithm (BA) is to mimic the behaviors of bats when catching their prey. BA was first presented in [34] and it was found to outperform Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) in evaluation using benchmark functions. BA has also been successfully applied to tough optimization problems such as motor wheel optimization problem [35], clustering problem [36], global engineering optimization, and constrained optimization tasks [37–40]. Very recently, two versions of bat-inspired algorithms have been proposed for feature selection [41, 42]. The implementation of BA is more complicated than many other meta-heuristic algorithms [43] because each agent (bat) is assigned a set of interacting parameters such as position, velocity, pulse rate, loudness, and frequencies. This interaction affects the quality of a solution and the time needed to obtain such solution.

The principle of bat algorithm is as follows. A swarm of bats is assumed to fly randomly with velocity V_i at position X_i with a fixed frequency f, varying wavelength λ, and loudness A_0 to search for a prey. They have the capability to adjust the wavelength of their emitted pulses and regulate the rate of pulse emission $r \in [0, 1]$, which is important to determine their closeness of the target. Although the loudness can be different in many ways, the loudness differs from a large (positive) A_0 to a minimum constant value $A_{\min}$. The frequency f is in the range $[f_{\min}, f_{\max}]$ that corresponds to a range of wavelengths $[\lambda_{\min}, \lambda_{\max}]$. For example, a frequency range of [20 kHz, 500 kHz] corresponds to a range of wavelengths from 0.7 mm to 17 mm.

5. Proposed Naive Bayes-Guided Bat Algorithm

5.1. Frequency. Frequency in the proposed algorithm is represented as a real number as defined in (4). The choice of minimum and maximum frequency depends on the application domain, where β is a random number range between 0 and 1. Frequency also affects the velocity as shown in (5). Consider the following:

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

5.2. Velocity. The velocity of each bat is represented as a positive integer number. Velocity suggests the number of bat attributes that should change at a certain moment of time. The bats communicate with each other through the global best solution and move towards the global best position (solution). The following equation shows the formula for velocity:

$$v_i^t = v_i^{t-1} + \left(x_* - x_i^t\right) f_i, \tag{5}$$

where $(x_* - x_i^t)$ refers to the difference between the length of global best bat and the length of the ith bat. When the difference is positive, this means that the global best bat has more features than those of the ith bat. When the result is summed with the previous velocity, it will accelerate the ith

bat towards the global best bat. If the difference is negative, this means that the ith bat has more features than those of the global best bat. Therefore, when the output is summed with the previous velocity, it will decrease the velocity of ith bat and help to attract it closer to global best bat. In the proposed Bat Algorithm-Naive Bayes (BANB) algorithm, the maximum velocity was setting ($V_{\max}$) equal to $(1/3) * N$, where N is the number of features. In the proposed BANB, (2) is used to adjust the velocity during each iteration; therefore, the proposed algorithm is adaptive for feature selection problem in order to mimic the original algorithm behavior. Velocity representation is also one major difference between BANB and the Binary Bat Algorithm (BBA) [42]. In BBA, the velocity is calculated for each single feature; hence, the algorithm is more time-consuming and departing from the original algorithm attitude. On the contrary, the velocity in the proposed BANB is calculated once for the entire solution; hence, the velocity amount determines the piece of change.

5.3. Position Adjustment. In the proposed algorithm, each bat position is formulated as a binary string of length N, where N is the total number of features. Each feature is represented by bit, where "1" means that the corresponding feature is selected and the "0" means that it is not selected. The positions are categorized into two groups according to the bit difference between the ith bat and the global best bat in order to align exploitation and exploration during searching.

The bat's position is adjusted depending on one of the following conditions. In the case where the velocity of ith bat is lower or equal to the number of different bits, ith bat will copy some features from global best bat, thus moving towards global best bat, while still exploring new search space. In the case where the velocities of ith bat are higher than the velocity of global best bat, then the ith bat will import all features from the global best bat to be the same as the global best bat with a few different bits to facilitate further exploitation. The following equation shows the position adjustment, where x is bat position, and v is the velocity of the ith bat at time t:

$$x_i^t = x_i^{t-1} + v_i^t. \tag{6}$$

5.4. Loudness. Loudness A_i in the proposed algorithm is represented as the change in number of features at certain time during local search around the global best bat, as well as local search around the ith bat. The formula for loudness is shown in (7), where A_i^t is the average loudness of all the bats at certain iteration and $\varepsilon \in [-1, 1]$. The value for sound loudness (A) ranges between the maximum loudness and minimum loudness. Consider the following:

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^t. \tag{7}$$

Generally, the loudness value will decrease when the bat starts approaching the best solution. The following equation shows that the amount of decrease is determined by α:

$$A_i^{t+1} = \alpha A_i^t. \tag{8}$$

The value for sound loudness also plays an important role in obtaining good quality solutions within a reasonable amount of time. The choice of the maximum and minimum loudness depends on the domain of application and also the size of the dataset. In the proposed BANB algorithm, the maximum loudness has been determined empirically as $(1/5) * N$, where N is number of features. Value for maximum loudness is dynamic depending on number of features in certain dataset. For example, when $A_{\max} = 3$ and $A_{\min} = 1$, the bat begins to reduce the number of features from 3 features to 2 features and the value then becomes a single feature when it gets closer to the target.

5.5. Pulse Rate. Pulse rate r_i has the role to decide whether a local search around the global best bat solution should be skipped or otherwise. Higher pulse rate will reduce the probability of conducting a local search around the global best and vice versa. Therefore, when the bat approaches the best solution, pulse rate value will increase and subsequently reduce the chances to conduct a local search around the global best. The amount of increase is determined by γ as defined in the following:

$$r_i^{t+1} = r_i^0 \left[1 - \exp\left(-\gamma t\right)\right]. \tag{9}$$

5.6. Fitness Function. Each candidate solution is using a fitness function defined in (10), where $P(Y_J \mid X)$ is the classification accuracy, TF is the total number of all features, and SF is the number of selected features. δ and φ are two parameters corresponding to the weight of classification accuracy and subset length, where $\delta \in [0, 1]$ and $\varphi = 1 - \delta$. From (10), we can see that the importance of classification accuracy and subset size is weighted differently. Generally, classification accuracy is given more weight than the size of the subset. In this experiment, the two parameters have been set as follows: $\delta = 0.9$, $\varphi = 0.1$. Consider the following:

$$\text{Sol}_A = \delta \cdot P\left(Y_j \mid X\right) + \varphi \cdot \frac{\text{TF} - \text{SF}}{\text{TF}}. \tag{10}$$

The complete algorithm for the proposed hybrid BA guided by Naive Bayes classifier (BANB) is shown in Algorithm 1.

6. Experiments and Results

The objective of the experiments is to evaluate the performance of the proposed Naive Bayes-guided Bat Algorithm (BANB) in terms of number of features selected and the classification accuracy achieved. To achieve this objective, we compared the number of features and classification accuracies of BANB with several well-known algorithms, which are Genetic Algorithms (GA) [44], Particle Swarm Optimization (PSO) [45], and Geometric Particle Swarm Optimization (GPSO) [46]. Similar to the proposed BANB, we also used Naive Bayes classifier for all comparative algorithms as the attribute evaluator. However, the parameters for the algorithms had the same settings as those used by the original authors. For the proposed algorithm, the parameters were set

to the following values: population size = 25 and decrease sound loudness and increase pulse rate both are set to 0.6. The initial value of pulse rate is equal to 0.2. The proposed BANB algorithm and other algorithms were run for 20 times with different initial solutions. Following [4, 17], all the algorithms were terminated after 250 iterations.

6.1. Description of Dataset. For the experiments, twelve datasets were considered to cover both cases of binary and multiclass data. Three of the datasets, namely, M-of-N, Exactly, and Exactly2, were sourced from [47]. M-of-N is an artificial binary class since the decision attribute consisting of two class labels and the dataset were generated from a uniform distribution to create the artificial domain. Exactly and Exactly2 are artificial binary classification datasets, generated based on x-of-y concepts, which are not linearly separable and are known to be difficult for many classes of learning algorithms [47]. The remaining datasets were taken from the UCI data repository [48]. The datasets are Vote, Credit, LED, Derm, Derm2, Lung, WQ, Heart, and Mushroom. Vote is widely used as a binary classification dataset in the literature. The dataset represents votes for each of the U.S. House of Representatives congressmen with the class label democrat and republican. Credit dataset is a binary classification data that is concerned with credit card applications. LED dataset in display domain is a multiclass classification data as the class label includes ten possible values in which the first seven features determine the class label of a pattern, whilst the rest of the 17 features are irrelevant.

Derm and Derm2 represent real data in dermatology concerning differential diagnosis of erythematosquamous diseases. The class labels contain six values, which refer to six different diseases. Lung dataset is the pathological types of Lung cancer that aims to demonstrate the power of the optimal discriminant plane even in ill-posed settings [49]. WQ is a multiclass label dataset that originated from the daily measures of sensors in an urban waste water treatment plant. The idea is to categorize the operational state of the plant with the purpose of predicting faults out of the state variables of the plant at each of the phases in the water treatment procedure. Heart is a binary class data that contains 76 attributes although all the published experiments reference to using only 14 of the original attributes. This data has been used to predict heart diseases, whereby the class label of zero and one refers to the absence or existence of heart disease in the patient. Finally, Mushroom is a binary class dataset that includes characterization of hypothetical samples identical to 23 types of gilled mushrooms in the *Agaricus* and *Lepiota* family. Table 1 shows the characteristics of the datasets.

(1) Initialize parameters: $A, A_{\min}, A_{\max}, r, f_{\min}, f_{\max}, P_{\max}, I_{\max}, V_{\max}, V_{\min}, \Phi, \delta, \gamma, \alpha$
(2) Generate a swarm with $P_{\max}$ bats
(3) Calculate cost function for all bats
(4) Find the current best bat (x_*)
(5) **While** stop condition not met **Do**
(6) **For** $i = 1$ **to** $P_{\max}$ **Do**
(7) Frequency $f_i = f_{\min} + (f_{\max} - f_{\min})\beta$
(8) Velocity $v_i^t = v_i^{t-1} + (x_i^t - x_i) f_i$
(9) **If** $(V_i > V_{\max})$ **Then** $(V_i = V_{\max})$ **End-If**
(10) **If** $(V_i < V_{\min})$ **Then** $(V_i = V_{\min})$ **End-If**
(11) Locations $x_i^t = x_i^{t-1} + v_i^t$
(12) **If** (Rand $> r_i$) **Then**
(13) calculate εA^t
(14) **If** $(\varepsilon A^t > A_{\max})$ **Then** $(\varepsilon A^t = A_{\max})$ **End-If**
(15) **If** $(\varepsilon A^t < A_{\min})$ **Then** $(\varepsilon A^t = A_{\min})$ **End-If**
(16) Generate a local solution around the best solution (x_*) $[x_{gb} = x_{\text{old}} + \varepsilon A^t]$
(17) **End-If**
(18) Calculate εA^t
(19) **If** $(\varepsilon A^t > A_{\max})$ **Then** $(\varepsilon A^t = A_{\max})$ **End-If**
(20) **If** $(\varepsilon A^t < A_{\min})$ **Then** $(\varepsilon A^t = A_{\min})$ **End-If**
(21) Generate a new solution around the current Solution x_i^t $[x_l = x_{\text{old}} + \varepsilon A^t]$
(22) **If** $x_l \geq x_{gb}$ $fx = x_l$ **Else** $fx = x_{gb}$ **End-If**
(23) **If** $(\text{Rand} < A_i) \,\&\, (f(fx) < f(x_*))$
(24) accept the new solution
(25) Increase r_i $r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)]$
(26) Decrease A_i $[A_i^{t+1} = \alpha A_i^t]$
(27) **End-If**
(28) **End-For**
(29) Find the current best solution (x_*)
(30) **End-While**

Algorithm 1: BANB Algorithm.

6.2. Results for Feature Selection Experiment. In this experiment, we compared the proposed BANB against GA [44], PSO [45], and GPSO [46] in terms of the number of features selected from the original dataset. Table 2 provides the comparison results. The number of features obtained from the comparative algorithms in Table 2, and the best results are highlighted in bold. Then the results are statistically tested using two tests, Kolmogorov-Smirnov and Levene test [50]. However, the Kolmogorov-Smirnov and Levene test did not meet the assumptions of normality distribution and equality of variance which then led us to use Wilcoxon test. Essentially, this test is an alternative to the paired t-test, when the assumption of normality or equality of variance is not met [51]. Wilcoxon test is rated to be a robust estimate tool that depended on the rank estimation [52]. Table 3 presents Wilcoxon test results for the proposed BANB algorithm

Table 1: Dataset characteristics.

Datasets	No. of features	No. of samples
Lung	56	32
WQ	38	521
Derm2	34	358
Derm	34	366
LED	24	2000
Mushroom	22	8124
Credit	20	1000
Vote	16	300
Heart	13	294
Exactly2	13	1000
Exactly	13	1000
M-of-N	13	1000

Table 2: Average of selected features.

Datasets	BANB	GA	GPSO	PSO
M-of-N	**6**	8	6.9	7.6
Exactly	**1**	7	7	6.9
Exactly2	**1**	3	3	2.9
Heart	**4**	6	5.7	6.25
Vote	**1**	3	3.1	3.55
Credit	**1**	10	11.7	11.9
Mushroom	**1**	5.75	6.15	5.9
LED	**5**	**5**	**5**	**5**
Derm	**12.3**	18.1	17.1	21.5
Derm2	**11.45**	17.75	17.3	20.75
WQ	**5.9**	20.1	20.15	21.05
Lung	**2**	8.6	7.95	8.7

Table 3: Wilcoxon test results.

	Wilcoxon test		
	GA-BANB	GPSO-BANB	PSO-BANB
M-of-N	.000 (BANB)	.013 (BANB)	.013 (BANB)
Exactly	.000 (BANB)	.000 (BANB)	.000 (BANB)
Exactly2	.000 (BANB)	.000 (BANB)	.000 (BANB)
Heart	.000 (BANB)	.000 (BANB)	.000 (BANB)
Vote	.000 (BANB)	.000 (BANB)	.000 (BANB)
Credit	.000 (BANB)	.000 (BANB)	.000 (BANB)
Mushroom	.000 (BANB)	.000 (BANB)	.000 (BANB)
LED	1	1	1
Derm	.000 (BANB)	.000 (BANB)	.000 (BANB)
Derm2	.000 (BANB)	.000 (BANB)	.000 (BANB)
WQ	.000 (BANB)	.000 (BANB)	.000 (BANB)
Lung	.000 (BANB)	.000 (BANB)	.000 (BANB)

against other feature selection algorithms. From Table 3, between the brackets refer to the algorithm that performs better than another algorithm. The results of Wilcoxon test are considered to be statistically significant at P less than 0.05 and are highly significant at P less than 0.01.

6.3. Results for Classification Accuracy Experiment. The second part of the experiment was to evaluate and compare the average classification accuracies achieved by BANB and other comparative algorithms over 10 runs, using 10-fold cross-validation method. Three well-known classifiers were employed for the purpose of evaluating the resulting subsets among different classifiers, which were JRip PART and J48 [53]. Tables 4, 5, and 6 show the average classification accuracy and standard deviation values from the experiment.

7. Discussions

In selecting the feature subset, Table 2 shows that the proposed BANB algorithm obtained the smallest number of features across all datasets except for LED. Table 3 confirmed that the difference between BANB and the remaining comparative algorithms is highly significant except for LED and M-of-N datasets. More significantly, BANB is able to reduce the number of features up to a single feature in five datasets as shown in Table 2. In evaluating the feature subset, if we take into consideration the interaction between classification accuracy and number of features selected by the proposed BANB algorithm as compared to other algorithms, we can categorize the results into three cases. In the first case, a reduced number of features deliver the same classification accuracy. This is shown in the Exactly dataset that produced similar classification accuracy in both JRip and J48 classifiers and even higher accuracy in PART classifier. On the contrary, features selected by other algorithms included more features, which indicate that some of the features selected are redundant. This can be seen clearly in the Exactly2 dataset when all solutions achieved exactly the same accuracy in spite of variance in the number of selected features.

In the second case, the proposed algorithm reduced the number of features while at the same time increased the classification accuracy. For example, BANB selected only two features from the Lung dataset as opposed to additional eight features among other algorithms. The difference between the numbers of features selected is attributed to noisy features, which cause a decrease in classification accuracy such as in the Vote dataset. In the third case, smaller feature subset that is selected delivers a slightly lower classification accuracy, such as in Heart and Mushroom dataset with the exception of LED dataset. All algorithms could deliver the same accuracy with the same number of features because the LED dataset contains very protruding features.

Finally, it can be noted from Tables 4, 5, and 6 that the classification accuracies achieved by the proposed BANB algorithm are in less disagreement or very close across three different classifiers. This can be noted obviously from the experimental results using Exactly, Credit, Lung, and Derm datasets. To support this finding, we calculated standard deviation for each dataset over the three different classifiers and we averaged the values for each algorithm. The results were as follows: BANB equals 0.36, GPSO equals 0.99, PSO equals 1.04, and, finally, GA equals 1.11. This implies that the proposed feature selection algorithm BANB has better generalization as compared to other feature selection algorithms. Results from Table 2 also show that BANB is capable of selecting the same number of features for 9 out of 12 datasets over 20 iterations. This is followed by GA, GPSO,

Table 4: Average classification accuracy with standard deviation for JRip.

Datasets	JRip			
	BANB	GA	GPSO	PSO
M-of-N	98.9 ± 0	99.1 ± 0	97.62 ± 4.32	98.92 ± 0.23
Exactly	68.8 ± 0	68 ± 0	68 ± 0	68.01 ± 0.22
Exactly2	75.8 ± 0	75.8 ± 0	75.8 ± 0	75.8 ± 0
Heart	80.61 ± 0	81.97 ± 0	81.66 ± 0.49	82.47 ± 5.33
Vote	95 ± 0	93.66 ± 0	93.92 ± 0.43	94.42 ± 0.35
Credit	71.6 ± 0	70.7 ± 0	70.75 ± 0.75	70.48 ± 0.79
Mushroom	98.52 ± 0	100 ± 0	99.46 ± 0.48	99.33 ± 0.45
LED	100 ± 0	100 ± 0	100 ± 0	100 ± 0
Derm	93.30 ± 1.69	89.39 ± 0.90	90.18 ± 2.41	91.08 ± 0.78
Derm2	90.77 ± 1.77	89.64 ± 1.64	90.47 ± 1.31	89.35 ± 1
WQ	70.99 ± 1.57	69.49 ± 1.35	68.59 ± 1.61	68.34 ± 1.39
Lung	87.5 ± 0	84.68 ± 1.77	83.74 ± 4.37	84.05 ± 2.73

Table 5: Average classification accuracy with standard deviation for PART.

Datasets	PART			
	BANB	GA	GPSO	PSO
M-of-N	100 ± 0	100 ± 0	98.53 ± 4.62	100 ± 0
Exactly	68.8 ± 0	65.6 ± 0	65.6 ± 0	65.93 ± 0.82
Exactly2	75.8 ± 0	75.8 ± 0	75.8 ± 0	75.8 ± 0
Heart	79.59 ± 0	80.27 ± 0	80.57 ± 0.49	79.79 ± 1.44
Vote	94.33 ± 0	94.33 ± 0	94.42 ± 0.15	94.49 ± 0.36
Credit	71.7 ± 0	72.9 ± 0	72.24 ± 1.14	71.81 ± 1.05
Mushroom	98.52 ± 0	100 ± 0	99.33 ± 0.46	99.39 ± 0.41
LED	100 ± 0	100 ± 0	100 ± 0	100 ± 0
Derm	94.39 ± 1.76	94.73 ± 0.86	95.65 ± 0.99	95.62 ± 0.71
Derm2	93.15 ± 2.80	95.19 ± 2.82	95.27 ± 0.76	95.61 ± 0.76
WQ	68 ± 0.7	67.76 ± 1.78	68.70 ± 1.28	67.07 ± 2.92
Lung	78.12 ± 0	80.93 ± 4.52	81.21 ± 5.84	80.27 ± 4.36

Table 6: Average classification accuracy and standard deviation for J48.

Datasets	J48			
	BANB	GA	GPSO	PSO
M-of-N	100 ± 0	100 ± 0	98.53 ± 4.64	100 ± 0
Exactly	68.8 ± 0	68.8 ± 0	68.8 ± 0	68.8 ± 0
Exactly2	75.8 ± 0	75.8 ± 0	75.8 ± 0	75.8 ± 0
Heart	79.93 ± 0	79.25 ± 0	79.25 ± 0	79.08 ± 0.28
Vote	95 ± 0	94 ± 0	94.13 ± 0.23	94.39 ± 0.30
Credit	71.7 ± 0	72.2 ± 0	72.21 ± 0.46	72.08 ± 1.08
Mushroom	98.52 ± 0	100 ± 0	99.49 ± 0.45	99.39 ± 0.41
LED	100 ± 0	100 ± 0	100 ± 0	100 ± 0
Derm	93.92 ± 1.04	95.02 ± 0.52	94.64 ± 0.77	94.36 ± 0.68
Derm2	91.25 ± 2.44	94.38 ± 1.67	93.73 ± 0.51	94.32 ± 0.89
WQ	65.08 ± 0.46	69.11 ± 1.35	68.30 ± 2.01	68.32 ± 2.82
Lung	87.5 ± 0	79.37 ± 4.70	80.62 ± 6.21	82.18 ± 4.67

and, finally, PSO. It can be noted that the standard deviation values in Tables 4, 5, and 6 are zeros for 9 datasets. This means that our proposed BANB could obtain certain number of features with exactly the same features for each iteration. As a consequence, BANB showed the highest stability among all comparative algorithms.

8. Conclusion

In this paper, a new hybrid feature selection algorithm has been presented. The Bat Algorithm employed Naïve Bayes Algorithm to intelligently select the most convenient feature that could maximize the classification accuracy while ignoring redundant and noisy features. We compared our proposed algorithm with three other algorithms using twelve well-known UCI datasets. The performance was evaluated from four perspectives, which are the number of features, classification accuracy, stability, and generalization. From the experiments, we can conclude that the proposed Naïve Bayes-guided Bat Algorithm (BANB) outperformed other meta-heuristic algorithms with a selection of feature subsets that are significantly smaller with a less number of features. In terms of classification accuracy, BANB has proven to achieve equal, if not better results as compared to other algorithms. For stability, the proposed algorithm is more stable than other algorithms. Finally, from the perspective of generalization of results, the resulting features produced by BANB are also more general than other algorithms in practice. For future work, further investigations are required to observe the behavior of the proposed algorithm in gene expression and very high-dimensional datasets.

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, "Feature subset selection using genetic algorithm," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 2, pp. 44–48, 1998.

[6] L. Ke, Z. Feng, and Z. Ren, "An efficient ant colony optimization 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, "Feature selection 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 feature subset selection using genetic 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. Debuse and V. J. Rayward-Smith, "Feature subset selection within a 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 attribute selection," in *Proceedings of the 11th International Conference on Machine Learning*, pp. 28–36.

[12] N. S. Jaddi 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, "Feature selection: 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. García-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. 45–52, 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] L. Bobrowski, "Feature selection based on some homogeneity coefficient," in *Proceedings of the 9th International Conference on Pattern Recognition*, pp. 544–546, 1988.

[22] D. Koller and M. Sahami, *Toward Optimal Feature Selection*, 1996.

[23] M. Modrzejewski, "Feature selection using rough sets theory," in *Machine Learning: ECML-93*, pp. 213–226, Springer, 1993.

[24] H. Liu and R. Setiono, "A probabilistic approach to feature selection: a filter solution," in *Machine Learning*, pp. 319–327, 1996.

[25] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[26] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in *Machine Learning*, pp. 601–608, 2001.

[27] 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.

[28] G. Feng, J. Guo, B.-Y. Jing, and L. Hao, "A Bayesian feature selection paradigm for text classification," *Information Processing and Management*, vol. 48, no. 2, pp. 283–302, 2012.

[29] H. T. T. Nguyen and D. K. Le, "An approach to improving quality of crawlers using naïve bayes for classifier and hyperlink filter," in *Computational Collective Intelligence. Technologies and Applications*, N. T. Nguyen, K. Hoang, and P. Jędrzejowicz, Eds., pp. 525–535, Springer, Berlin, Germany, 2012.

[30] Z. Zhang, Q. Zhu, and Y. Xie, "A novel image matting approach based on naive bayes classifier," in *Intelligent Computing Technology*, D. S. Huang, C. Jiang, V. Bevilacqua, and J. Figueroa, Eds., pp. 433–441, Springer, Berlin, Germany, 2012.

[31] M. C. Yang, C. S. Huang, J. H. Chen, and R. F. Chang, "Whole breast lesion detection using naive bayes classifier for portable ultrasound," *Ultrasound in Medicine and Biology*, vol. 38, pp. 1870–1880, 2012.

[32] C. Catal and B. Diri, "Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem," *Information Sciences*, vol. 179, no. 8, pp. 1040–1058, 2009.

[33] Z. Hoare, "Landscapes of Naïve Bayes classifiers," *Pattern Analysis and Applications*, vol. 11, no. 1, pp. 59–72, 2008.

[34] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies For Optimization (NICSO' 10)*, J. González, D. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., pp. 65–74, Springer, Berlin, Germany, 2010.

[35] T. C. Bora, L. D. S. Coelho, and L. Lebensztajn, "Bat-inspired optimization approach for the brushless DC wheel motor problem," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 947–950, 2012.

[36] K. Khan, A. Nikov, and A. Sahai, "A fuzzy bat clustering method for ergonomic screening of office workplaces," in *Proceedings of the 3rd International Conference on Software, Services and Semantic Technologies (S3T '11)*, D. Dicheva, Z. Markov, and E. Stefanova, Eds., pp. 59–66, Springer, Berlin, Germany, 2011.

[37] X. S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, 2012.

[38] A. H. Gandomi, X. S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications*, vol. 22, pp. 1239–1255, 2012.

[39] I. Fister Jr., D. Fister, and X. S. Yang, "A hybrid bat algorithm," *Elektrotehniski Vestnik*, vol. 80, pp. 1–7, 2013.

[40] X. S. Yang, "Bat algorithm for multi-objective optimisation," *International Journal of Bio-Inspired Computation*, vol. 3, pp. 267–274, 2011.

[41] A. M. Taha and A. Y. C. Tang, "Bat algorithm for rough set attribute reduction," *Journal of Theoretical and Applied Information Technology*, vol. 51, no. 1, 2013.

[42] R. Nakamura, L. Pereira, K. Costa, D. Rodrigues, J. Papa, and X. S. Yang, "BBA: a binary bat algorithm for feature selection," in *Proceedings of the 25th Conference on Graphics, Patterns and Images (SIBGRAPI '12)*, pp. 291–297, 2012.

[43] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: a survey," *International Journal of Bio-Inspired Computation*, vol. 3, pp. 1–16, 2011.

[44] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1989.

[45] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, 1998.

[46] A. Moraglio, C. Di Chio, and R. Poli, "Geometric particle swarm optimisation," in *Proceedings of the 10th European Conference on Genetic Programming (EuroGP '07)*, pp. 125–136, April 2007.

[47] B. Raman and T. R. Ioerger, *Instance Based Filter for Feature Selection*, 2002.

[48] C. L. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases*, University of California, Irvine, Calif, USA, 1998.

[49] Z.-Q. Hong and J.-Y. Yang, "Optimal discriminant plane for a small number of samples and design method of classifier on the plane," *Pattern Recognition*, vol. 24, no. 4, pp. 317–324, 1991.

[50] H. W. Lilliefors, "On the Kolmogorov-Smirnov test for normality with mean and variance unknown," *Journal of the American Statistical Association*, vol. 62, pp. 399–402, 1967.

[51] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, pp. 80–83, 1945.

[52] F. Monti, R. Dell'Anna, A. Sanson, M. Fasoli, M. Pezzotti, and S. Zenoni, "A multivariate statistical analysis approach to highlight molecular processes in plant cell walls through ATR FT-IR microspectroscopy: the role of the a-expansin PhEXPA1 in Petunia hybrida," *Vibrational Spectroscopy*, vol. 65, pp. 36–43, 2013.

[53] Q. Duan, D. Miao, H. Zhang, and J. Zheng, "Personalized web retrieval based on rough-fuzzy method," *Journal of Computational Information Systems*, vol. 3, no. 3, pp. 1067–1074, 2007.