Improved Binary Bat Algorithm for Feature Selection

Md Wali Ul Alam

42008

Master of Science (Technology) Thesis

Supervisor: Adnan Ashraf

Department of Information Technologies

Åbo Akademi University

2019

Abstract

Metaheuristic algorithms are basically used to explore search space to find optimal solutions. It is a form of generalized local search which gives a good solution for optimization problems. There are many advantages that make these algorithms more useful to solve a wide range of problems today. One of the major advantages of metaheuristic algorithms is giving quick and efficient solutions for large and complicated problems. There are many metaheuristic algorithms that are used for classification problems, for example, Particle Swarm Optimization (PSO), Cuckoo Search (CS), Genetic Algorithm (GA), and Firefly Algorithm (FA). Among the existing metaheuristics, the Bat Algorithm (BA) is well known for its unique features and is powerful enough to solve classification problems such as the feature selection problem. There are different versions of BA that solve a wide range of problems. The original BA is used for continuous problems, while the Binary Bat Algorithm (BBA) is used for solving discrete problems. In this thesis, a wrapper-based unsupervised feature selection problem is being solved by using a modified version of BBA with the sum of squared error as the fitness function. An experiment conducted with four public datasets and the k-means clustering algorithm has been used to validate the proposed modified version of BBA. The results show that the proposed method outperforms several alternative methods by selecting a minimal set of features with improved accuracy.

Keywords: Bat Algorithm, Binary Bat Algorithm, Modified Binary Bat Algorithm, Feature Selection, Unsupervised Learning, K-means Clustering.

Acknowledgements

I would like to take the opportunity to express my gratitude to the people without whom this thesis would not have been possible. Firstly, I would like to show my heartiest gratefulness to my supervisor Adnan Ashraf, Research Associate, Department of Information Technologies, Åbo Akademi University for his continuous support and advice. His endless patience, careful attention, scholarly guidance, and constructive criticism has helped me to produce this thesis. Furthermore, I want to thank my friends for their advice and motivations both in and off the work. Finally, I owe my deepest appreciation to my parents throughout writing this thesis and life in general; without their support nothing would have been possible.

Md Wali Ul Alam

Turku, 2019

Contents

Abstract **i**

Acknowledgements **ii**

Contents **iii**

List of Figures **vi**

List of Tables **vii**

1 Introduction **1**

1.1 Optimization 2

1.1.1 Optimization Algorithm 2

1.2 Population Methods 4

1.3 Thesis Structure 4

2 Metaheuristic Algorithms **5**

2.1 Genetic Algorithm 6

2.2 Particle Swarm Optimization 6

2.3 Ant Colony Optimization 8

2.4 Artificial Immune System Algorithm 8

3 Original Bat Algorithm **11**

3.1 Definition 11

3.1.1 The Echolocation Characteristics of Micro-Bats 12

3.1.2 Movement of Virtual Bats 12

3.2 Pseudo Code of Bat Algorithm 13

3.3 Exploration versus Exploitation 14
3.4 Efficiency 14
3.4.1 Frequency Tuning 15
3.4.2 Automatic Zooming 15
3.4.3 Parameter Control 15
3.5 Variants of Bat Algorithm 15
3.5.1 Fuzzy Logic Bat Algorithm (FLBA) 16
3.5.2 Multiobjective Bat Algorithm (MOBA) 16
3.5.3 Chaotic Bat Algorithm (CBA) 16
3.5.4 Binary Bat Algorithm (BBA) 17
3.5.5 Improved Bat Algorithm (IBA) 17
3.6 Applications of Bat Algorithm 18
3.6.1 Classifications, Clustering and Data Mining 18
3.6.2 Continuous Optimization 19
3.6.3 Image Processing 19
3.6.4 Fuzzy Logic Application 19
3.7 Recent Researches 19
3.7.1 Binary Bat Algorithm for Feature Selection 19
3.7.2 A new improved Bat Algorithm on micro-grid operations . . . 20
3.7.3 A bat-inspired algorithm for structural optimization 20
3.7.4 Bat Algorithm for the fuel arrangement optimization of reactor core 20

4 Problem Description **21**
4.1 Feature Selection 21
4.2 Classification of Feature Selection Methods 23
4.2.1 Filter Methods 24
4.2.2 Wrapper Methods 24
4.2.3 Embedded and Hybrid Methods 24
4.3 Application of Feature Selection 25
4.3.1 Text Mining 25
4.3.2 Image Processing 25

4.3.3 Industrial Applications 26
4.3.4 Bioinformatics 26

5 Proposed Method **28**
5.1 Binary Bat Algorithm 28
5.2 Modified Binary Bat Algorithm 31
5.3 k-means Clustering 33
5.4 k-Nearest Neighbor Classifier 34

6 Experimental Evaluation **36**
6.1 Experiment Design 36
6.1.1 Dataset Description 36
6.1.2 Evaluation Metrics 37
6.2 Result and Analysis 39

7 Conclusion and Future Work **45**
7.1 Conclusions 45
7.2 Future Work 46

Bibliography **47**

List of Figures

4.1 Feature selection process 23

5.1 Sigmoid transfer function 29

5.2 V - shaped transfer function 30

5.3 Uniform distribution between 0 and 1. 32

6.1 Results of Wisconsin Breast Cancer Dataset. 40

6.2 Results of Splice Dataset. 41

6.3 Results of Ionosphere Dataset. 42

6.4 Results of SVMGuide1 Dataset. 43

List of Tables

6.1 Dataset description 37
6.2 Confusion matrix 37
6.3 Result of confusion matrices from experiment 39
6.4 Various evaluation metrics deduced from confusion matrix 39

Chapter 1

Introduction

In the field of machine learning, feature selection [1] is one of the most fundamental problems which has the ability to decrease the number of features to increase the performance of classification. Most of the features in a dataset do not contribute to better results. Therefore, these irrelevant and unimportant features not only increase the complexity of the system but also the processing time. Thus, feature selection is important for better performance in classification. Feature selection can be used in medical diagnosis tasks [2] where the task is to select a subset of clinical tests. Large-scale data mining application is another example where the feature selection problem is used. Power system control [3], text classification [4], selecting subset of sensors in autonomous robot design [5] are extensively used applications for feature selection problems. To solve these problems efficiently, metaheuristic algorithms can be used which are powerful enough to deal with such problems. Each of these algorithms is discovered through the behavior of biological nature and has its own advantages and disadvantages. However, the traditional optimization algorithm is used to solve the problem in a smaller dimension. Therefore, many researchers started to focus on nature which provided nature-inspired models to solve many difficult problems, such as optimization problems. Different researchers have devised Swarm intelligence algorithm by simulating natural biological systems. The metaheuristic algorithms that are widely used are Firefly Algorithm (FA) [6], Genetic Algorithm (GA) [7], Artificial Bee Colony (ABC) [8] and Particle Swarm Optimization (PSO) [9]. Among them Bat

Algorithm, which is a swarm intelligence-based algorithm proposed by Yang [10], is one of the most powerful and widely used algorithms because of echolocation characteristics of bats that give good balance to both exploration and exploitation [10] components. Thus, BA is considered as one of the possible solutions to solve data mining problems such as feature selection. Feature selection [1] is a binary problem, whereas the original version of Bat Algorithm is used to solve continuous problems. In this thesis, an improved version of the Binary Bat Algorithm is proposed. The proposed algorithm is evaluated by solving feature selection problems on unsupervised learning and comparative results are discussed.

1.1 Optimization

Optimization [11] is a process which solves different types of problems more efficiently by using less resources and features. In all aspects of human progression, optimization is applicable. In computing, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. The basic optimization problem consists of objective function, variables and constraints. Over the years, different approaches have been developed to solve discrete and continuous optimization problems.

1.1.1 Optimization Algorithm

To solve optimization problems the optimization algorithms [12] or optimization problems can be divided into the following groups [12]:

- Classification based on existence of constraints
  - Constraint optimization problem
  - Unconstrained optimization problem
- Classification based on nature of the design variables
  - A set of design parameters that make a prescribed function

    - A set of design parameters, which are all a continuous function of some other parameter

- Classification based on the physical structure of the problem
    - Optimal Control
    - Non-optimal Control
- Classification based on the nature of the equations involved
    - Linear programming problem
    - Non-linear programming problem
    - Geometric programming problem
    - Quadratic programming problem
- Classification based on the permissible values of the decision variables
    - Integer programming problem
    - Real-valued programming problem
- Classification based on the deterministic nature of the variables
    - Deterministic programming problem
    - Stochastic programming problem
- Classification based on separability of the functions
    - Separable programming problem
    - Non-separable programming problem
- Classification based on number of objective functions
    - Single-objective programming problem
    - Multi-objective programming problem

1.2 Population Methods

Optimization values are modified to solve optimization problems. Population-based methods are different than optimization problems. In population-based methods, each solution that exists is called individual and the set of current individuals is a population. Each individual is involved in tweaking and quality assessment and generating new population.

1.3 Thesis Structure

The thesis is organized as follows: Chapter 2 describes the Metaheuristic algorithms. Chapter 3 introduces the basic behavior, applications, and variants of the original Bat Algorithm. Chapter 4 provides descriptive information of the problem which is feature selection problem. Chapter 5 starts with discussing the binary version of the Bat Algorithm and the proposed modification over the Binary Bat Algorithm. Later, The k-nearest neighbor classifier and k-means are also discussed in Chapter 5. Chapter 6 presents the experimental procedure, dataset description, evaluation metrics, and results analysis of the experiment, and Chapter 7 provides conclusions and direction for future research.

Chapter 2

Metaheuristic Algorithms

Optimization is one of the most important parts of the engineering models. In the architecture of the software, an optimal design is accomplished by comparing various planned solutions which are designed by the use of earlier knowledge of the problem [13]. In these types of activities, the feasibilities of the particular planned solution are investigated in the initial stage. After the feasibility study is done the cost of the particular planned solution is calculated and the best planned at the low-cost solution is selected. Optimization is the way to find the appropriate solution for the problem with the calculation of the quality of the solution. In many organizations, the optimization is approached by comparing a few options of the solution to the problem and choosing the best one. The metaheuristic is a developed optimization technique to create, select or locate the heuristic that might give the appropriate solution to the problem, basically for the incomplete or partial data provided [14]. The heuristic is an essential way of solving optimization problems. It applies a practical step that is not guaranteed to be the perfect or optimal solution for the problem but is sufficient for the current goal. Some of the algorithms that have the potential in optimization and search for the complex problems. The important metaheuristic algorithms are discussed in the following sections.

2.1 Genetic Algorithm

The Genetic Algorithm is created by Holland [13] in the 1970s and was implemented for practical use in the 1980s. The Genetic Algorithm is mainly used in power system optimization problems. It also consists of the reactive power design for solving the global minimum uncertainty problems [15]. The parallel operation ability has been inherited by the Genetic Algorithm and utilizes the mail functions data that are straightly searched for the directions and not for the alternative or any auxiliary information. The steps that are to be followed in the Genetic Algorithm are listed below [16]:

Step 1: Generate the initial population.

Step 2: Evaluate the fitness of each individual in the population.

Step 3: Select best ranking individuals to reproduce.

Step 4: Perform Simulated Binary Crossover for generating two children products from the parents.

Step 5: Perform Mutation on the particular strings of the offspring. The parents are replaced by the newly generated who act as a parent for the coming up generation.

Step 6: Terminate algorithm after performing needed number of operations.

2.2 Particle Swarm Optimization

The Particle Swarm Optimization algorithm was developed by Eberhart and Kennedy [9] in 1995. It is a population-based optimization method that is inspired by the social behavior of the fish schooling and bird flocking [17]. The Particle Swarm Optimization can be successfully applied to any type of problem that is able to be expressed in terms of an objective function. The Particle Swarm Optimization algorithm involves the initialization of the number of particles that are inside the search space of the objective function.

The steps that are to be followed in the Particle Swarm Optimization are listed below [18]:

Step 1: Fix the number of steps and generate the initial population.

Step 2: Calculate objective function with planned variables.

Step 3: Set $Pbest[i] = initial\ solution$ where $i = 1, 2, ...N$ (N: no of particles).

Step 4: Store the objective variable.

Step 5: Find out the best of the particles and set the *Gbest* with the best particle value. Compare the objective function of step 2 with that of step 3 objective function, and then store the swapped data.

Step 6: Calculate the initial velocities for all the particles.

Step 7: Set the velocities to the equivalent particles,

$$present[i](new) = present[i](old) + v[i]$$

Step 8: Set the velocity according to,

$$V[i] = V[i](present) + Cl * (Pbest[i] - present[i]) + C2 * (Gbest[i] - present[i])$$

Step 9: Calculate the set particles to find out the new ones.

Step 10: Check if the number of steps for the optimization is less, then go to step 6.

Step 11: Compare the last solution that is obtained from and the objective function value and go back to step 7, if the two are not equal, else, terminate the loop.

Step 12: After finding out the required number of steps the algorithm needs to be terminated.

2.3 Ant Colony Optimization

The mechanical analogy of the ant, which the ant algorithm is based on is that of the ant colonies. The ants create the shortest route from their nest to the sources of the food and then return to their nest [19]. The tracking characteristics of the ants were very much appreciated and inspired the latest computation algorithm that was used for the optimization of the real-life systems which were encountered in applied science and were utilized for solving the large-scale problem optimization such as the Travelling Salesman Problem [20]. The steps that are followed for the Ant Colony Optimization are listed below [20]:

Step 1: Calculate objective function and the number of regions.

Step 2: Perform descending or ascending function according to problem statement on the areas, till the fitness value.

Step 3: Perform crossover and mutation over 75% of the global ants in the inferior region.

Step 4: Select the inferior region and send the ants for the local search.

Step 5: If the local search and the global search are done successfully in the improved areas, then rearrange (descending or ascending according to problem statement) the areas values of the objective function.

Step 6: Choose the best objective function.

Step 7: Terminate the algorithm after needed number of operations.

2.4 Artificial Immune System Algorithm

The Artificial Immune System planned a new creative decision process that was to be utilized in various problem statement optimizations. The Artificial Immune System is the design of the procedure that is impressed by the principles, theoretical medicines, mechanism, considered immune function, and the mechanisms that are utilized for solving of the problem statement [21]. It has a creative process algorithm that belongs to the computational intelligence family and is inspired by the biological immune system [21]. During the past decade, they have brought much attention from researchers aiming to develop immune-based models and methods to solve complex computational or engineering problems. The steps that are to be followed in the Artificial Immune System are listed below [22]:

Step 1: Generate N size of population randomly along with the planned variables.

Step 2: Copy the planned variables to the operating population.

Step 3: Calculate the objective function for the strings and do continuous ranking and sorting.

Step 4: Perform selection procedure based on ranking. The string which has the highest affinity will be considered as the best individual.

Step 5: Perform the equation from the cloning of the antibodies.

$$N_c = \sum(n - (i-1)) \tag{2.1}$$

Where, N_c is considered as the number of the clones that are generated for each of the following strings, n is considered to be the total number of strings that are present in the population, and i is considered as the present string that is starting from the string that has the highest affinity.

Step 6: Perform mutation over the set of clones that are the replicas of the variables with good affinity value. The formula for the two-phased process of the mutation is the following:

$$x + \frac{\alpha * range * generation}{N_c} \quad \text{and} \quad x' = x + \frac{\alpha * range}{N_c * generation} \tag{2.2}$$

Where, x' is the mutated value, α is the random number in the range of 0 and 1, range is the variable which lies between the upper limit and lower limit, and the generation is considered as the present generation cycle.

Step 7: After the operation of the first mutation operator, replace the older clone with the new one, otherwise, use the second operator.

Step 8: In the process of the second mutation step, if no improvement is found in the mutated string, then keep the original solution like before.

Step 9: Use receptor editing mechanism model, which is the part of the worst solution, and the current ones were generated in their place, after the predefined iterations.

Step 10: Calculate and sort the objective function again for the clones. Mutate it, rank it, and then do the extraction of the first ten best solutions from the list.

Chapter 3

Original Bat Algorithm

The modern method of the optimization algorithms is most of the time inspired by nature, and the ensuing algorithm can be of many various varieties. Still, all the algorithms tend to be utilized with some of the essential characteristics for the calculation of the main modernized formula. An immense number of heuristic and metaheuristic algorithms [23] have been derived based on the features of the biological and physical system in nature. For example, Particle Swarm Optimization [24] was developed based on the swarm behavior of birds and fish, while simulated annealing [25] was based on the tempering process of metal.

3.1 Definition

In 2010, Xin-She Yang developed a metaheuristic algorithm that is inspired by the nature of bats, which is called Bat Algorithm [10]. Micro-bats [26] play a crucial role in building the main characteristics of this algorithm. The main two parameters that are used in this algorithm are pulse rates and emission, and the values of these two parameters can be tweaked. The Bat Algorithm also utilizes the frequency-tuning method, to expand the variety of solutions that are present in the population, even though at the same time. It uses automatic zooming that attempts to adjust the exploration and exploitation throughout the process by imitating the variation of the heartbeat outflow and the loudness of the bats during the hunting of their preys. There is a unique ability of echolocation of micro-bats which can find their prey and separate different sorts of the insect in total darkness. These excellent characteristics make the algorithm more efficient with an excellent quick start.

3.1.1 The Echolocation Characteristics of Micro-Bats

- All micro-bats use echolocation behavior to find their desired subject (food or prey) and barriers. Also, by using this behavior, they can sense distance and acquire the ability to discriminate between food, prey, or barriers.
- Bats fly randomly with a velocity at any position with a fixed frequency, varying wavelength and loudness to search for prey. [26].
- The loudness of micro-bats can vary, and it can go from a large positive value to a minimum constant value [26].
- The frequency range corresponds to a range of wavelengths.

3.1.2 Movement of Virtual Bats

Bats fly randomly with velocity v_i at position v_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $\gamma\varepsilon[0,1]$, depending on the proximity of their target.

In simulations, virtual bats have their positions x_i and velocities v_i in a d- dimensional search space at time step t; that are given by [10]:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{3.1}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \tag{3.2}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3.3}$$

where $\beta \varepsilon [0,1]$ is a random vector drawn from a uniform distribution. Here x_* is the current global best location (solution) which is located after comparing all the solutions among all the n bats.

After selecting the current best solution among the other solutions, Equation (3.4) is used to generate a new solution for each bat [10].

$$x_{new} = x_{old} + \varepsilon A^t \tag{3.4}$$

The average loudness of all the bats is $A_t = < A_i^t >$ and $\varepsilon \varepsilon [-1,1]$ denotes a random number.

3.2 Pseudo Code of Bat Algorithm

Based on approximations and idealization of the echolocation characteristic, the basic steps of the Bat Algorithm (BA) can be summarized in pseudo code as follows [10]:

Algorithm 3.1 Bat Algorithm (BA)

```
Objective function f(x), x = (x_1, ..., x_d)^T
Initialize the bat population x_i (i = 1, 2, ..., n) and v_i
Define pulse frequency f_i at x_i
while (t < Maxnumberofiterations) do
    Generate new solutions by adjusting frequency,updating velocities and locations/solutions
    if (rand > r_i) then
        Select a solution among best solutions
        generate a local solution around the best solution
    end if
    Generate a new solution by flying around randomly
    if (rand < A_i) & f(x_i) then
        Accept new solutions
    end if
    Arrange the bats according to rank and highlight the current best x_*
end while
Post-process results and visualization
```

3.3 Exploration versus Exploitation

Exploration is related to global search and exploitation is similar to local search. In the first one, we are interested in exploring the search space looking for reasonable solutions, whereas in the second one we want to refine the solution and try to avoid big jumps on the search space.

Diversification is how difficult the individuals are within a population. In this context, it is desirable to have a good diversification in our population to explore the search space properly. A low level of diversification means a genetic convergence, where, in theory, the algorithm is closer to the final solution. Thus, it is desirable to have a high level of diversification at the beginning of the algorithm and a lower one at the end. Therefore, a low level of diversification, in the beginning, leads to a premature convergence, which is inconvenient. Also, when the diversification is high, the algorithm is intensified to exploration; otherwise, it is focused on the local search.

3.4 Efficiency

There are various purposes behind the achievement of the Bat Algorithm. In many applications, the Bat Algorithm achieves a certain objective by using a minimum amount of materials. Non-linear optimization problems are not as difficult to solve as topology optimization problems. Results suggest that the Bat Algorithm proved to solve both non-linear and topology optimization problems. From the key features and updating equations, efficiency of the Bat Algorithm can be summarized in the following points [27]:

- Frequency Tuning
- Automatic Zooming
- Parameter Control

3.4.1 Frequency Tuning

Bat Algorithms [10] use echolocation and frequency tuning to solve a different kinds of problems. In reality, however, it is not possible to use echolocation directly, so frequency variations are used to solve problems. Particle Swarm Optimization and Harmony Search possessed some essential functionality that is taken from characteristics of frequency tuning. These unique features put the Bat Algorithm on a more advanced position compared to other swarm intelligence techniques.

3.4.2 Automatic Zooming

The Bat Algorithm [10] can perform an automatic zooming feature which is useful to find promising solutions by zooming into the region. At the early stage of the iterations, it has a quick convergence rate because of automatic zooming that helps to swap from exploration to local intensive exploitation, which gives the Bat Algorithm a massive advantage over other metaheuristic algorithms.

3.4.3 Parameter Control

The Bat Algorithm [10] has more parameters to control. Some of the parameters can be initialized with random numbers, but other parameters are set up according to the experiment. Sometimes more parameters make it difficult to use the Bat Algorithm for beginners, but it is useful to increase the efficiency of the algorithm. The values of the parameters vary in each iteration. Whenever the algorithm is approaching for an optimal solution, this parameter control technique shows a way to switch from exploration to exploitation automatically.

3.5 Variants of Bat Algorithm

The Bat Algorithm [10] is a unique metaheuristic algorithm with many advantages. At the initial stage, it can switch from exploration to exploitation because of very fast convergence, which makes the Bat Algorithm more effective in classification and when a quick solution is needed. Many strategies and methods have been followed to produce different variants of the Bat Algorithm. The variants of the Bat Algorithm are described below [27]:

- Fuzzy Logic Bat Algorithm (FLBA)
- Multiobjective Bat Algorithm (MOBA)
- Chaotic Bat Algorithm (CBA)
- Binary Bat Algorithm (BBA)
- Improved Bat Algorithm (IBA)

3.5.1 Fuzzy Logic Bat Algorithm (FLBA)

Khan et al. [28] were first to present the fuzzy logic into Bat Algorithm, which was named Fuzzy Bat Algorithm. In the field of the distributed system, the Bat Algorithm was used by Reddy and Manoj [29] who used it to find optimal capacitor placement for loss reduction. It combined the fuzzy logic to find the optimal capacitor sizes to reduce waste. Furthermore, Tamiru and Hashim [30] applied the Bat Algorithm in gas tribune.

3.5.2 Multiobjective Bat Algorithm (MOBA)

Multiobjective algorithms [31] used to find both optimal and maximum optimal solutions. Engineering optimization with multiobjective and multidisciplinary are complex problems which need efficient optimization algorithms. Xin-She Yang [31] extended the original version of the Bat Algorithm to solve multiobjective optimization problems.

3.5.3 Chaotic Bat Algorithm (CBA)

Chaos is introduced into the Bat Algorithm [10] for robust global optimization. With the help of Levy flights and chaotic maps, Lin et al. [32] introduced the Chaotic Bat Algorithm to accomplish parameter estimation in the dynamic biological system. Later, four different variants of the Chaotic Bat Algorithm and thirteen different types of chaotic maps for validation of those variants are introduced by Amir H. Gandomi and Xin-She Yang [33].

3.5.4 Binary Bat Algorithm (BBA)

The Binary Bat Algorithm [34] is a discrete version of the Bat Algorithm which is developed by Nakamura et al. [10] to solve classification [35] and feature selection problems. Nakamura researched new feature selection techniques, and he proposed a new method dependent on bats behavior. The wrapper method is used in Nakamura's proposed approach, which is powerful enough to combine the exploration of all the bats and the Optimum-Path Forest classifier. This approach is used to find a specific set of features. Later, this specific set of features helps to increase the accuracy in a validating set. Five public datasets are used to conduct an experiment, and the results of the experiment showed that the BBA perform better compared to the Particle Swarm Optimization (PSO) [24] and the Genetic Algorithm (GA) [7].

However, the original version of the Bat Algorithm [10] cannot be involved in binary problems directly. In Xin-She Yangś paper, an experiment has been performed to draw a conclusion. Twenty-one benchmark functions are used over binary PSO and GA in the experiment. Furthermore, Wilcoxonś rank-sum non-parametric statistical test was carried out with five significance levels to judge the proposed algorithm. The results proved that the proposed algorithm outperforms others on the majority of the benchmark functions.

3.5.5 Improved Bat Algorithm (IBA)

The original version of the Bat Algorithm has two parameters that can be controlled. One is the loudness, and another one is the pulse rate. Jamil et al. [36] tweaked the values of different variations of loudness and pulse emission rates and combined these values with Levy flights to make an improvement of the original version of the Bat Algorithm.

Yilmaz and Kucuksille [37] show that the heuristic optimization technique is trendy and has extensive usage areas. Many researchers applied these methods in many fields.

The fundamental motivation behind these methods is to accomplish excellent performance. In this study, three modifications have been made over exploration and exploitation mechanisms to improve the Bat Algorithm. The performance of the proposed algorithm is tested against the original version of the algorithm based on ten benchmark test problems. The solution quality of the test result showed that the recommended version is better.

3.6 Applications of Bat Algorithm

The original version of the Bat Algorithm has been implemented on the different field of applications because of its vast diversity of variants. Some of the major applications are discussed below.

3.6.1 Classifications, Clustering and Data Mining

Komarasamy and Wahi [38] performed an experiment with the combination of k-means clustering and the Bat Algorithm and they stated that the combination of k-means with the Bat Algorithm can achieve higher accuracy and therefore better than various algorithms.

Faritha Banu and Chandrasekar [39] performed document duplication as an optimization using revised Bat Algorithm and their research proposed that the revised Bat Algorithm is better than genetic programming.

Khan and Sahai [40] presented the Binary Bat Algorithm in the context of the e-learning and recommended that bat set of rules are better than the other algorithms. Later, Khan and Sahai [41] proposed a new type of the Bat Algorithm based on clustering problems, which is named bi-sonar optimization.

Marichelvam, along with Prabahram [42] carried out an experiment with the Bat Algorithm to solve a hybrid flow-shop-scheduling problems and the results put the Bat Algorithm as the best choice for solving the hybrid flow-shop-scheduling problems.

3.6.2 Continuous Optimization

The Bat Algorithm deals with continuous optimization problems, and this is the frequent application of the Bat Algorithm, which is extensively studied, and it indicated that the BA is highly proficient at solving nonlinear problems and can locate the ideal solutions more precisely (Yang, 2010; Yang and Gandomi, 2012; Yang, 2012; Yang et al., 2012a).

Bora et al. [43] used optimization technique by the help of the Bat Algorithm on brushless DC motors and provided successful results.

3.6.3 Image Processing

Abdel-Rahman et al. [44] led an examination on whole human body utilizing the Bat Algorithm and reasoned that it performs superior to Annealed Particle Filter (APF), Particle Swarm Optimization (PSO), and Particle Filter (PF).

Du and Liu [45] researched on image matching and proposed a model based on bats. They marked their model more effective and feasible in image matching than other models such as Genetic Algorithm.

3.6.4 Fuzzy Logic Application

Lemma et al. [46] combined the fuzzy logic and the Bat Algorithm for energy modeling, and later Tamiru and Hashim [30] applied the Bat Algorithm in gas tribune.

3.7 Recent Researches

There have been some exciting researches on the Bat Algorithm over the years. Some important research topics are discussed in the following sections.

3.7.1 Binary Bat Algorithm for Feature Selection

Feature selection is an important technique that is used to find the important feature among a given set of features. The Bat Algorithm combined together with the wrapper-based approach and the speed of the Optimum-Path Forest classifier to find the set of features that maximizes the accuracy in a validating set [10]. The research was performed by R. Y. M., Pereira, Nakamura, L. A. M., Costa, K. A., Yang, X. S., Rodrigues, D., Papa, J. P., (2012).

3.7.2 A new improved Bat Algorithm on micro-grid operations

This study formulated micro-grid operation management under battery energy storage sizing. The proposed Bat Algorithm used several parallel velocity updating strategies, considered the effects of optimal battery sizing on micro-grid operation performance, validate the proposed framework using Rastrigin, Ackley, and Griewank functions and also tested the effectiveness of the proposed framework on a micro-grid test system. The research was submitted by Bahman Bahmani-Firouzi corresponding and Rasoul Azizipanah-Abarghooee [47].

3.7.3 A bat-inspired algorithm for structural optimization

In this research, a novel algorithm is developed with the Bat Algorithm for structural optimization. The algorithm has the efficiency of the Bat Algorithm and is verified through numerical examples. The investigation was performed by O. Hasançebi Corresponding, T. Teka and O. Pekcan [48].

3.7.4 Bat Algorithm for the fuel arrangement optimization of reactor core

S. Kashi, A. Minuchehr, N. Poursalehi Corresponding, and A. Zolfaghari has showed for the first time that the Bat Algorithm can be developed for the core pattern optimization problem [49]. The Bat Algorithm is efficient enough to solve almost all kind of optimization problems, but there is still room for improvement. By conducting more researches, the Bat Algorithm can be improved with better performance. In future, researchers should focus on the theoretical understanding of the metaheuristic algorithms and large-scale problems in real-world applications.

Chapter 4

Problem Description

Researchers are working on handling the massive amounts of data year after year and it is quite a challenging task. The raw data are noisy in nature and contain irrelevant features which are also known as high-dimensional data. The performance of the machine-learning algorithms is highly dependent on data and these raw data are the major cause to downgrade the performance of the algorithms. In the machine-learning process, the irrelevant features cannot be included and the redundant features contain the same information, which misleads the learning process. To overcome these problems, the feature selection technique is introduced. In the proposed thesis, the feature selection problem will be solved based on unsupervised learning. To solve this problem, the modified version of the Binary Bat Algorithm will be used.

4.1 Feature Selection

Feature selection [1] is a technique to extract a set of necessary features for an experiment. There can be thousands of features that describe a class of objects entirely. There can be billions of objects in that particular class. To find an object that is somehow similar in nature or to create a subclass or to discriminate between these sub-classes, we depend on features that describe these objects. However, classifying depending on thousands of features can take a long time, up to several years, and is very costly. Thus, we try to find an alternative way that is much cheaper and less time consuming, a way in which we can obtain almost the same results. We select a subset containing features which are most dissimilatory.

The particular problem in any machine-learning platform is to exactly evaluate the relationship $f(X)$ under input data X =$\{x_1,x_2,....,x_M\}$ and the Y output is memory dependent data points of $\{X_i,Y_i\}$, $i = 1,...,,N$, normally X_i and Y_i vectors are real and whole numbers. The Y output is undetermined for input characteristics total set of $\{x_1,x_2,...,x_M\}$. Moreover, the decision is taken by their subset of $\{x_{(1)},x_{(2)},...,x_{(m)}\}$, $(m < M)$. Under sufficiency of periods, the use of features of input with the irrelevant features to find the exactness of the function to the contrast of input and output is considerable [50]. However, the whole learning process has been observed to evoke two general problems, under the irrelevancy of features. The rise in computational costs is due to the irrelevant features in the input. The increase in features increases the computational cost concerning a large number of predictions. Over-fitting is induced to the irrelevancy of features in the input.

An insightful motivation of the feature selection is to find and prioritize the modes of ignorance to the less significant input features on the outcome (output), while the model size is maintained and the model of approximate is kept small [51]. In the community of statistics, "Subset selection" is the alternative name of feature selection, where a thorough analysis is provided in [1]. Also, Feature selection is known as variables or attributes selection where features denote variables or attributes. There are four basic steps for feature selection which are shown in figure 4.1 [52] and the steps are discussed below [52]:

- The Generation step will create a new subset from the original feature set.
- The Evaluation function will evaluate the subset that is created from the original feature set. It will decide the relevancy towards classification using dependency, consistency, distance and information.
- Stopping criteria will stop the whole procedure if an optimal feature subset is found.
- The Validation step will the take a decision whether the chosen feature subset is

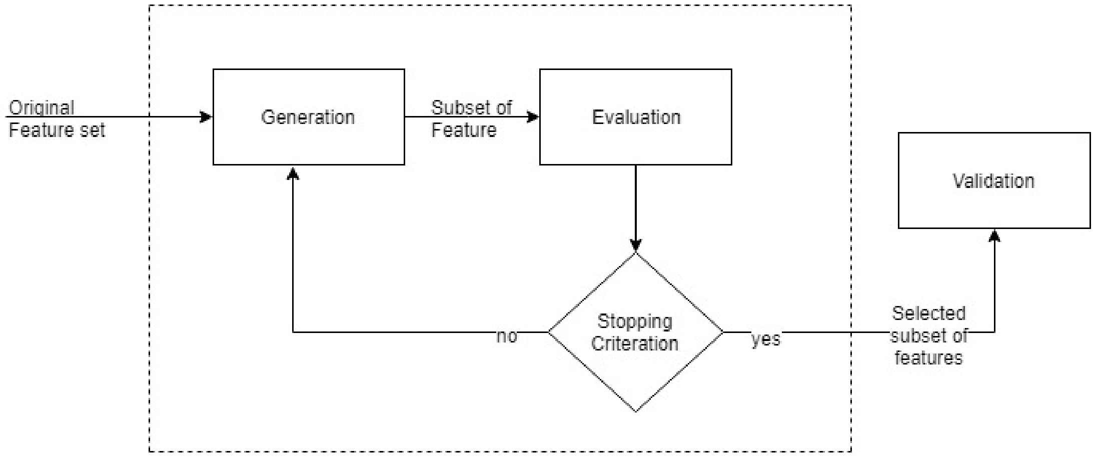

Figure 4.1: Feature selection process [52].

validated or not.

Mostly machine-learning algorithms are classified into supervised and unsupervised learning [53]. In supervised learning, classifiers are used which come from learning the labeled data and constructing a model based on that learning. Unsupervised learning is quite the opposite. In unsupervised learning, clustering is used which comes from learning unlabeled data and constructing a model based on that learning. All the features are not important. Some features affect clustering algorithms. In addition, reducing features causes problems that the unsupervised algorithm breaks down with high dimensional data.

4.2 Classification of Feature Selection Methods

Feature selection [1] methods are classified into a certain number of ways and among them the standard approaches are the filter, wrapper, embedded and hybrid methods [54, 55]. These methods can be applied in both supervised and unsupervised learning [53]. The Wrapper-based approach has been chosen for the proposed thesis, because this method gives better performance by reducing the main criteria of the learning model, which ensures fewer errors. Also, the wrapper method uses classification performance like the accuracy of a classifier to do the evaluation.

4.2.1 Filter Methods

The Filter method does not focus on building algorithms as models; instead, it focuses on picking features based on performance. Later, the modeling algorithm can use the best features after they were selected. Not all filter methods can be applied to different problems of machine learning. Different kinds of filter methods are used based on the problems, for example, classification, clustering, or regression. Univariate and multivariate feature filters are different. Univariate filters rate one feature while multivariate filters rate a whole feature subset. Univariate feature filters are independent, scalable, and fast but ignore learning algorithms and the relationship between features. Multivariate feature filters are independent and utilize feature relationship but show poor scalability and are slow.

4.2.2 Wrapper Methods

Wrapper methods are different than filter methods, because wrappers focus on the quality of the modeling algorithm and based on that wrappers select features. If the wrapper approach is applied in the classification task, then the subset of features will be evaluated based on the performance of the classifier (e.g. SVM) [56]. However, the wrapper will evaluate the subset of features based on the clustering algorithm in clustering (e.g. k-means) [57].

4.2.3 Embedded and Hybrid Methods

During the execution of the modeling algorithm, embedded methods perform feature selection. In this method, both the feature selection algorithm and learning algorithm are integrated with each other. Decision-tree algorithms and various types of regression algorithms are part of the embedded method.

Hybrid methods are the combination of both filter and wrapper approaches [58]. Filter approaches work for minimizing the search space dimension and the wrapper approach helps to find the best subset.

4.3 Application of Feature Selection

Feature selection is a brighter area in the world of machine learning and other predictable environments. It has various application areas based on different methods. Major areas such as text mining, image processing, and industrial sectors are significant areas where feature selection is mostly used.

4.3.1 Text Mining

Text mining has been classified into a wide range of applications such as text categorization, text summarization, and chosen entity extraction. Depending on the application, various methods, such as mining, classification and clustering, are used for text categorization application.

Forman [58] performed an experiment using the filter feature selection method for text classification. The performance was evaluated based on an SVM classifier. Precision, recall, F-measure, and accuracy are the evaluation metrics that are selected for analyzing the performance. Two hundred twenty-nine classification problems were used to evaluate twelve feature selection metrics.

Liu et al. [4] research feature selection problems of text clustering and the study shows that feature selection can improve its performance and efficiency. In the experiment, three document datasets are used, and five filter selection methods are tested. The authors proposed a supervised feature selection method with clustering in a bootstrap setting. The proposed method reduces entropy and increased precision.

4.3.2 Image Processing

Image classification [59, 55] is one of the problems that are solved using feature selection. Many features are used in identifying an image, but only a few of them play a vital role to increase the efficiency and effectiveness of classification.

Bins and Darper [60] researched the image classification problem by using filter-based feature selection. The combination of Relief, k-means clustering, sequential floating forward/backward feature selection algorithm is used in three steps by three different image datasets, and the results prove that the hybrid algorithm performed much better than when using a single algorithm.

Mustra et al. [61] used a mammographic image in the research and the wrapper-based feature selection method for breast destiny classification. In the following analysis, two datasets of the mammographic image have been used along with three different classifiers. The results showed a significant improvement after using feature selection.

4.3.3 Industrial Applications

The feature selection method is vital in the industrial application to maintain the excellent performance of the machines. Liu et al. [62] applied feature selection to improve the accuracy of detecting the fault in a system. The proposed approach was compared with different SVM, entropy-based feature selection, and neural network wrappers, and the best performance can be achieved by combining the proposed method with a wrapper.

4.3.4 Bioinformatics

Feature selection plays an important role in the field of bioinformatics. Genomic datasets contain individual gene features. Feature selection can be applied in these datasets to obtain the unique feature of genes and solve a particular problem based on those genes.

Dessi et al. [63] worked with DNA microarray for comparing eight selection methods on three benchmark functions. Furthermore, they broke down how comparable the yields of diverse choice strategies are and discovered that the outputs of the univariate approach appear to be increasingly similar to each other than to the multivariate procedures.

Abusamra [64] worked with two datasets of microarray gene expression data to analyze the performance of eight different filter-based feature and three classification methods. The classifier and the dataset that are used in the experiment will determine the performance of the feature selection method. Moreover, Abusamra added that the accuracy could be further improved for both datasets. All the processes that are used in the experiment selected seven features.

Chapter 5

Proposed Method

5.1 Binary Bat Algorithm

The binary version of the Bat Algorithm has a similar concept as the original version of the Bat Algorithm; the dissimilarities in the algorithm lie in the search space of the algorithm. The original Bat Algorithm performs in the continuous search space, whereas the binary version of the algorithm performs the task in the discrete search space [45]. The search space of the Binary Bat Algorithm is restricted to 0's and 1's, so the change of the velocity and the change of the position of the search space cannot be executed. In this thesis, the search space is designed as an n-cube, where n is the number of features. For a situation like this, the optimal or near optimal solutions are chosen from 2^n potential outcomes, and it compares to one of the n-cube's corners [65]. For solving continuous optimization problems, each bat moves in the search space towards the continuous-valued position. However, in case of either all or no problems this approach fails. Such situations are created by problems such as feature selection [1], binary knapsack [66], and job-shop scheduling [67], etc. To solve these types of problems the original Bat Algorithm [10] is modified to a binary version [34]. The modification is done by using the sigmoid transfer function which restricts the position of the new bats from continuous to binary values. The responsibility is to be taken that the transfer function that is selected is between the interval of [0, 1] and the returned transfer function is directly proportional to the alternative of the velocity.

The transfer function of the discussed Binary Bat Algorithm is defined in Equation (5.1) and Figure 5.1 [34].

$$S(v_i^k(t)) = \frac{1}{1+e^{-v_i^j(t)}} \tag{5.1}$$

Here, i is the particle and $v_i^k(t)$ indicates the velocity of the corresponding particle at t iteration.

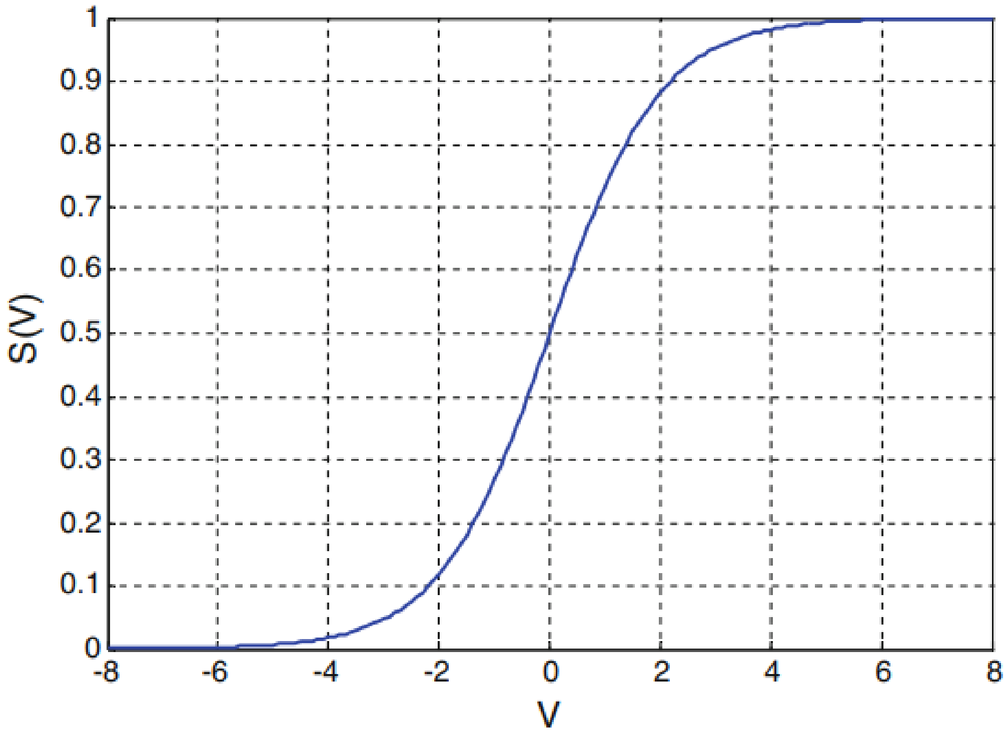

Figure 5.1: Sigmoid transfer function [34].

After the calculation of the probabilities using transfer function, Equation (5.2) is used to update the particle's new position [34].

$$x_i^k(t+1) = \begin{cases} 0 & \text{if rand} < \mathrm{S}(\mathrm{v}_i^k(t+1)) \\ 1 & \text{if rand} \geq S(v_i^k(t+1)) \end{cases} \tag{5.2}$$

Here, $x_i^k(t)$ denotes position and $v_i^k(t)$ denotes velocity corresponding to i-th particle at t-th iteration in k-th dimension. A typical V- shaped transfer function looks like Figure

5.2 [34] and the transfer function is used along with a rule that is used to update the position using Equation (5.3) and (5.4) [34].

$$V(v_i^k(t)) = |\frac{2}{\pi} arctan(\frac{\pi}{2} v_i^k(t))| \tag{5.3}$$

$$x_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1} & \text{if rand} < \mathrm{V}(\mathrm{v}_i^k(t+1)) \\ x_i^k(t) & \text{rand} \geq V(v_i^k(t+1)) \end{cases} \tag{5.4}$$

Here, $x_i^k(t)$ denotes position and $v_i^k(t)$ denotes velocity corresponding to i-th particle at t-th iteration in k-th dimension and $(x_i^k(t))^{-1}$ is the component of $x_i^k(t)$.

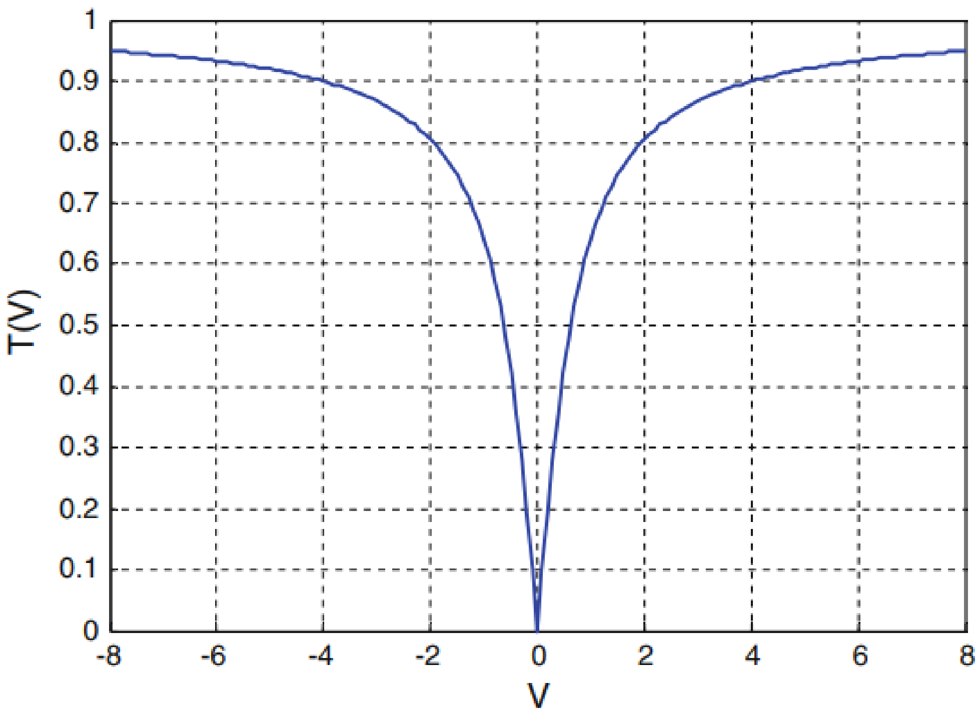

Figure 5.2: V - shaped transfer function [34].

The pseudo code of the Binary Bat Algorithm is as follows [34]:

Algorithm 5.1 Binary Bat Algorithm (BBA)

Initialize the bat population: $X_i(i = 1, 2, ..., n) = rand(0 \text{ or } 1)$ and $V_i = 0$
Define pulse frequency F_i
Initialize pulse rates r_i and the loudness A_i
while $(t < Maxnumberofiterations)$ **do**
    Adjusting frequency and updating velocities
    Calculate transfer function value using equation (5.3)
    Update positions using equation (5.4)
    if $(rand > r_i)$ **then**
        Select a solution (Gbest) among the best solutions randomly
        Change some of the dimensions of position vector with Gbest
    end if
    Generate a new solution by flying randomly
    if $(rand < A_i)$ & $f(x_i) < f(Gbest)$ **then**
        Accept the new solutions
        Increase r_i and reduce A_i
    end if
    Arrange the bats according to rank and highlight the current Gbest
end while

5.2 Modified Binary Bat Algorithm

In this thesis, we used a trial and error approach with some modification and parameter tuning with the BBA, which is previously mentioned. The following list represents the contributions to the Binary Bat Algorithm and the pseudo code of the modified version of the Binary Bat Algorithm is presented in Algorithm 5.2.

- Number of Bats: We choose the initial population to be half of the number of features in the input dataset. We assume that it should reduce overhead in computation.

- Frequency Range: We followed the paper by Rani et al. [68]. In this paper, there was no clear indication of the frequency range. So, we assumed it supports the frequency range of the original Bat Algorithm, which is for continuous numerical optimization. For our work, we determined that the frequency should range over 0 to 2 for the best and fast performance of the algorithm.

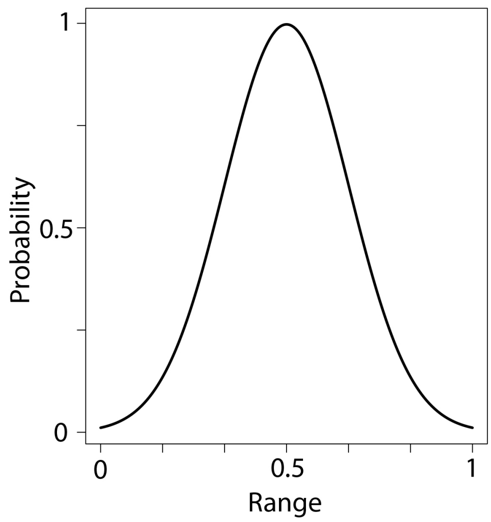

Figure 5.3: Uniform distribution between 0 and 1.

- Loudness and Pulse Rate: The method proposed in [68], initializes loudness, A_0 at 0.5 and pulse rate r_0 at 0.9 for all bats of the first generation. We changed both of these with a random function which generates values between 0 to 1 so that the population may start and change randomly. This is why our proposed method works better where there is more area to explore i.e., better exploration. We used uniform distribution for our random number generation. The distribution is shown in Figure 5.3.

Algorithm 5.2 Modified Binary Bat Algorithm (MBBA)

Initialize the bat population half of the number of features of dataset, iteration = 40
Define pulse frequency $F_i = rand(0 \text{ to } 2)$
Initialize pulse rates $r_i = rand(0 \text{ to } 1)$
Initialize loudness $A_i = rand(0 \text{ to } 1)$
fit1 = fitness of initial bats are calculated using (6.1)
minimum_fit = minimum fitness of bat
while $(t < iteration)$ **do**
    Adjust frequency and update velocity
    Calculate transfer function value using equation (5.3)
    Update positions using equation (5.4)
    if $(t < rand)$ **then**
        generate new bats
    end if
    if $(rand > r_i)$ **then**
        update newbats with gbest
    end if
    fit2 = fitness of new bats are calculated using (6.1)
    if $(fit1 < fit2)$ && $(rand > A_i)$ **then**
        Update initial bat, reduce A_i and increase r_i
    end if
    if $(fit2 < minimum\_fit)$ **then**
        update gbest
    end if
end while

5.3 k-means Clustering

Clustering follows the property of an object to change the disorganized set of objects to an organized one. There are lots of popular clustering algorithms, but among them, the most popular one is the k-means algorithm, which is suitable for the fast iterative algorithms [69]. The main task of this algorithm is to divide n objects from the dataset for k clusters that use center-based clustering methods [70]. This algorithm is helpful when the label of the data is missing, and it is also the simplest of all the algorithms to implement and to run. In this thesis, the k-means clustering algorithm is used to validate the relevance of the selected features. The k-means algorithm is described as follows [71]:

Algorithm 5.3 k-means Clustering Algorithm

1: Choose initial centroids $\{m_1, ..., , m_k\}$ of the clusters $\{C_1, ..., , C_k\}$.

2: Calculate new cluster membership. A feature vector x_j is assigned to the cluster C_i if and only if,

$$i = argmin_{k=1...k} ||x_j - m_k||^2 \tag{5.5}$$

3: Recalculate centroids for the cluster according.

$$m_i = \frac{1}{|C_i|} \sum_{x_j \varepsilon C_i} x_j \tag{5.6}$$

4: If none of the cluster centroids have changed, finish the algorithm. Otherwise go to Step 2.

5.4 k-Nearest Neighbor Classifier

Classification is a field of research in machine-learning, which classifies objects, images, sound, and text, etc. The most straightforward classification algorithm in machine-learning techniques is the k-Nearest-Neighbor (k-NN) [72], which is simple but effective. The k-NN is a lazy learning algorithm because it stores the training data and waits for the testing data and classification is conducted using the most related training data. This method classifies instances based on the relationship among variables and can be used for both classification and regression. The k-NN algorithm is described as follows [72]:

Algorithm 5.4 k-NN Classification Algorithm

```
Classify $(X,Y,x)$ where X: training data, Y: class label, x: unknown sample
for $i = 1$ to $m$, do Compare distance $d(X_i, x)$
Compute set $I$ containing indices for $k$ smallest distances $d(X_i, x)$
return majority label for $\{Y_i$ where $iI\}$
```

Chapter 6

Experimental Evaluation

The proposed binary version of modified Bat Algorithm (MBBA) is implemented in MATLAB using Intel Core i5 processor with 12 GB RAM and 1 TB hard disk running on Windows 10 operating system.

6.1 Experiment Design

In this experiment, confusion matrix, precision, recall, and f-measure are used as the evaluation metrics to evaluate the results. Classification accuracy and the number of selected features are used for performance measure and comparative analysis. The accuracy of the selected features is tested with the k-nearest neighbor classification algorithm using WEKA [73]. To get better results, the algorithm was executed forty times, and the 10-fold cross-validation is used for the selected features as a classification model in the proposed method which helps to evaluate the model by dividing the original sample into training and test set. Sum of Squared Error is selected as the fitness function for the k-means clustering algorithm.

6.1.1 Dataset Description

The proposed method is tested on four different real-world datasets: Wisconsin Breast Cancer, Splice, Ionosphere, and SVMGuide1, which are publicly available in the UCI machine-learning repository. The class level of all the datasets is removed because the proposed method is for unsupervised learning. Table 6.1 presents the key characteristics of these datasets.

Table 6.1: Dataset description

S. No.	Data Set	No. of Samples	No. of Features	Classes
1	Wisconsin Breast Cancer	683	10	2
2	Splice	1000	60	2
3	Ionosphere	351	34	2
4	SVMGuide1	3089	4	2

6.1.2 Evaluation Metrics

The performance of the proposed method is validated with six evaluation metrics. First, the confusion matrix [74], which is also known as the error matrix, gives performance visualization of an algorithm. The columns and the rows of the confusion matrix has a different meaning. The columns represent each instance in a predicted class, whereas the rows represent each instance in an actual class. For two-class classification problem, the confusion matrix is shown in Table 6.2 [74].

Table 6.2: Confusion matrix

<table>
<tr><td></td><td></td><td colspan="2">Prediction</td></tr>
<tr><td></td><td></td><td>Negative</td><td>Positive</td></tr>
<tr><td rowspan="2">Actual</td><td>Negative</td><td>TN</td><td>FP</td></tr>
<tr><td>Positive</td><td>FN</td><td>TP</td></tr>
</table>

The entries in the confusion matrix have the following meaning:

- True Positive (TP): Number of correct positive predictions.
- True Negative (TN): Number of correct negative predictions.
- False Positive (FP): Number of incorrect positive predictions.
- False Negative (FN): Number of incorrect negative predictions.

Next evaluation metric is the Sum of Squared Error (SSE), which is used for clustering algorithms to validate. SSE is defined as follows [75]:

$$SSE(X,\Pi) = \sum_{i=1}^{k} \sum_{x_j \varepsilon c_i} ||x_j - c_i||^2 \tag{6.1}$$

Where, N is the feature vector and x is a data point in cluster c_i. The target object (x_i) of each group (c_i) relies on the euclidean distance measurement (m_i) to determine the clustering quality. Euclidean distance and m_i is centroid of cluster c_i, which can be calculated from the equation below:

$$m_i = \frac{1}{|c_i|} \sum_{x_j \varepsilon c_i} x_j \tag{6.2}$$

Accuracy [76] is also used as an evolution metric, which is the ratio of the number of samples appropriately classified to the total number of samples as given below:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} * 100 \tag{6.3}$$

The highest classification accuracy features are considered as the best features.

Other three evaluation metrics that are used in this experiment are precision, recall, and f-measure. Precision [77] denotes the ratio of observations that are correctly predicted to the total number of positive predicted observations.

$$Precision = \frac{TP}{TP+FP} \tag{6.4}$$

Recall [77] denotes the ratio of observations that are correctly predicted to all the views.

$$Recall = \frac{TP}{TP+FN} \tag{6.5}$$

F-measure [77] is the weighted average combination of both precision and recall.

$$F_{measure} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.6}$$

6.2 Result and Analysis

In Table 6.3, the confusion matrices for the Wisconsin Breast Cancer, Splice, Ionosphere, and SVMGuide1 datasets are given which provide us the information regarding the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) obtained for all four datasets.

Table 6.3: Result of confusion matrices from experiment

Wisconsin Breast Cancer

	Class 1	Class 2
Class 1	434	10
Class 2	25	214

Splice

	Class 1	Class 2
Class 1	458	25
Class 2	182	335

SVMGuide1

	Class 1	Class 2
Class 1	198	891
Class 2	38	1962

Ionosphere

	Class 1	Class 2
Class 1	198	891
Class 2	38	1962

The evaluation metrics such as precision, recall, and f-measure which are calculated from true positives, true negatives, false positives, and false negatives are given in Table 6.4.

Table 6.4: Various evaluation metrics deduced from confusion matrix

S. No.	**Dataset name**	**Precision**	**Recall**	**F-measure**
1	Wisconsin Breast Cancer	0.949	0.949	0.948
2	Splice	0.827	0.793	0.789
3	SVMGuide1	0.741	0.699	0.629
4	Ionosphere	0.869	0.867	0.869

The results of the experiment are compared with the methods presented in [68] based on the accuracy and the number of the selected features. Figure 6.1, 6.2, 6.3, and, 6.4 show the results of the experiment on four different datasets. Figure 6.1(a) and 6.1(b) compare the accuracy and the number of the selected features of Wisconsin Breast

Cancer dataset by the BBA with the k-means and Optimum-Path Forest (OPF) and other swarm intelligence algorithms, namely Firefly Algorithm (FA) [6], Gravitational Search Algorithm (GSA) [78], Harmony Search (GS) [79], and Particle Swarm Optimization (PSO) [80] against the proposed Modified Binary Bat algorithm (MBBA) with the k-means algorithm. The proposed method resulted higher accuracy of 94.87% and identified a minimal set comprising two features.

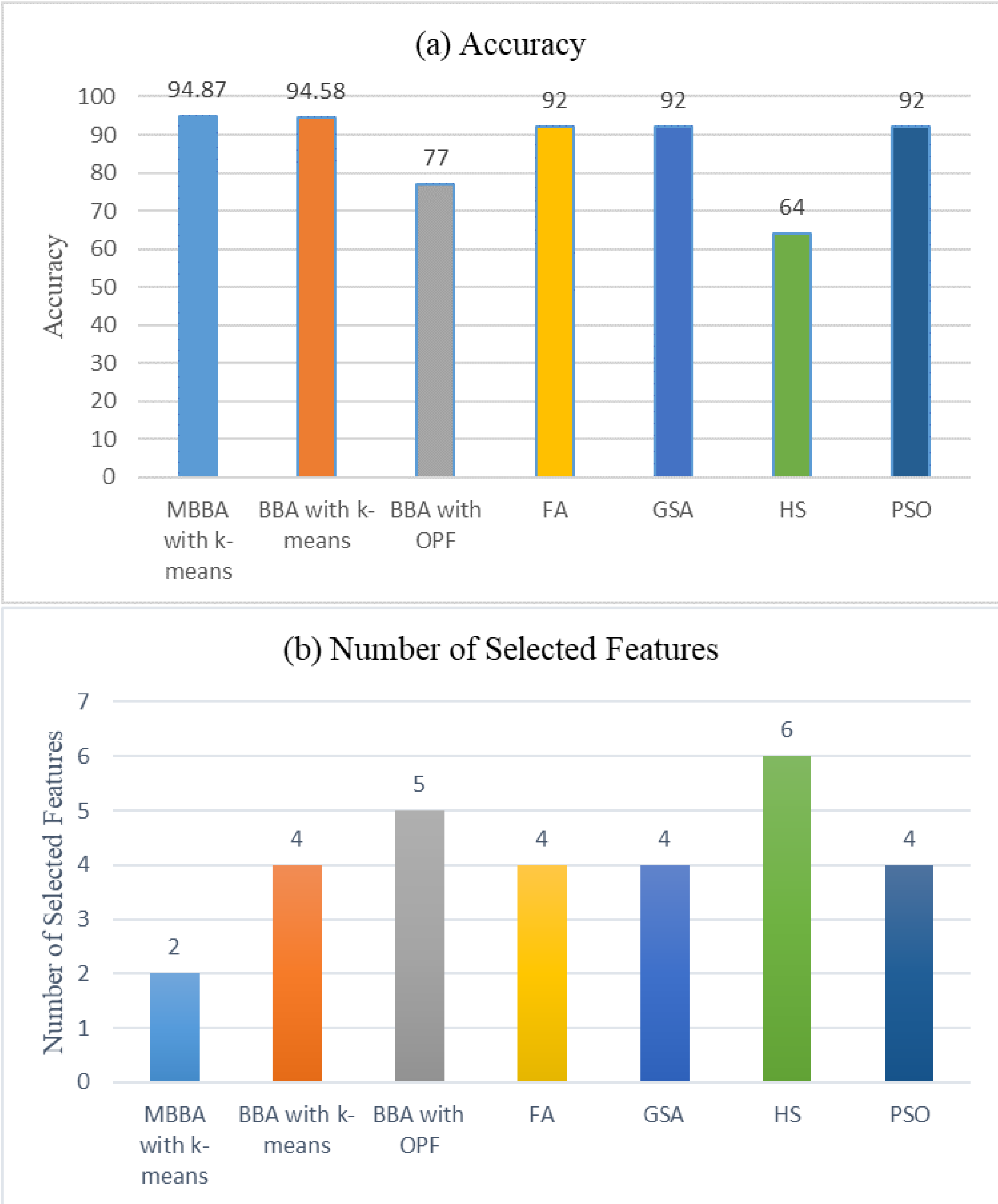

Figure 6.1: Results of Wisconsin Breast Cancer Dataset.

Figure 6.2(a) and 6.2(b) shows the accuracy of the MBBA with the k-means algorithm is 79.30% and the selected features is sixteen over Splice dataset, which is better compared to the BBA with the k-means algorithm, BBA with OPF, FA, GSA, HS and PSO.

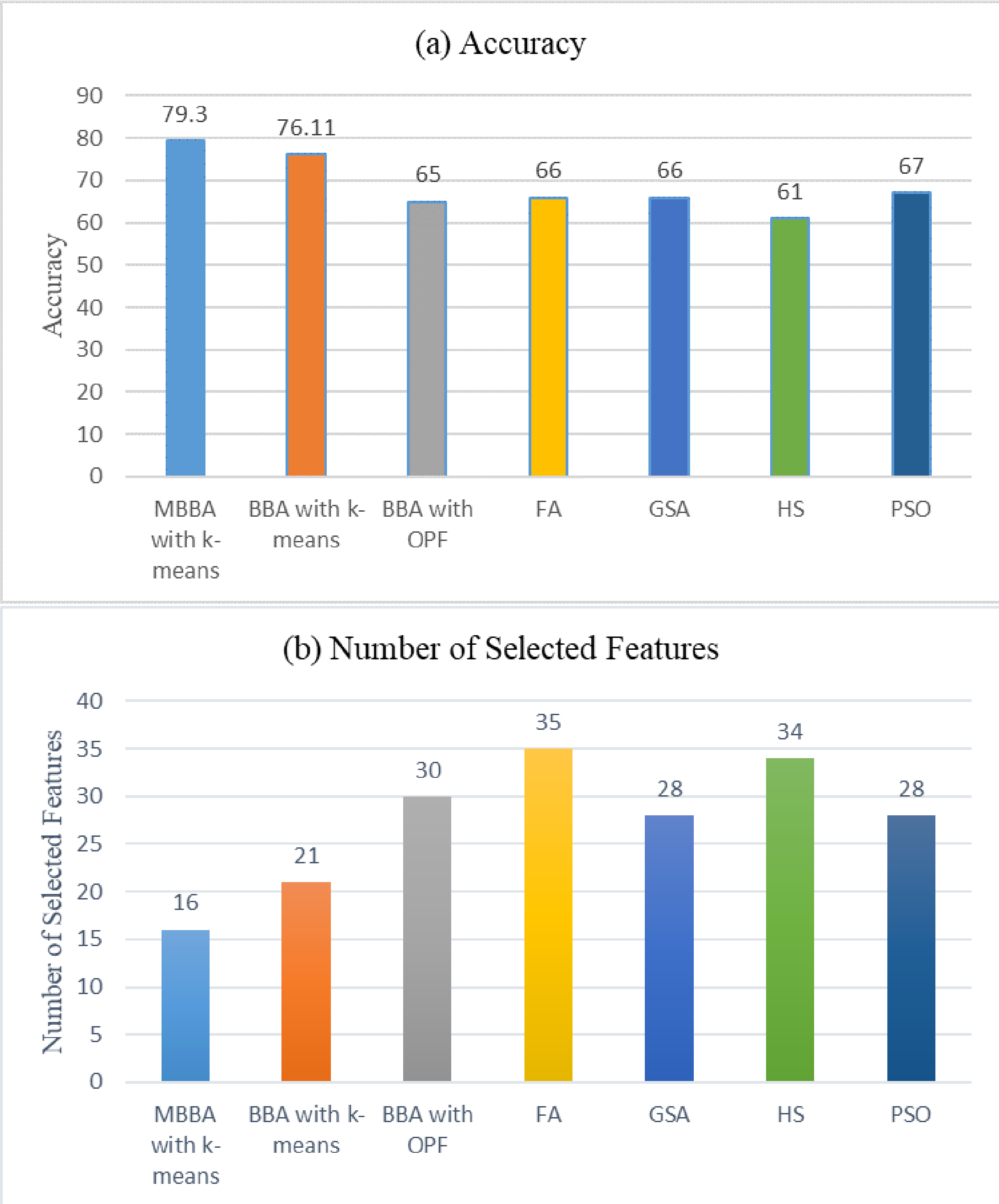

Figure 6.2: Results of Splice Dataset.

The accuracy of the MBBA with the k-means for Ionosphere is 86.89%, which is higher and the number of the selected features is six which is lesser when compared to BBA with the k-means, BBA with OPF, FA, GSA, HS, and PSO are shown in Figure 6.3(a) and 6.3(b) respectively.

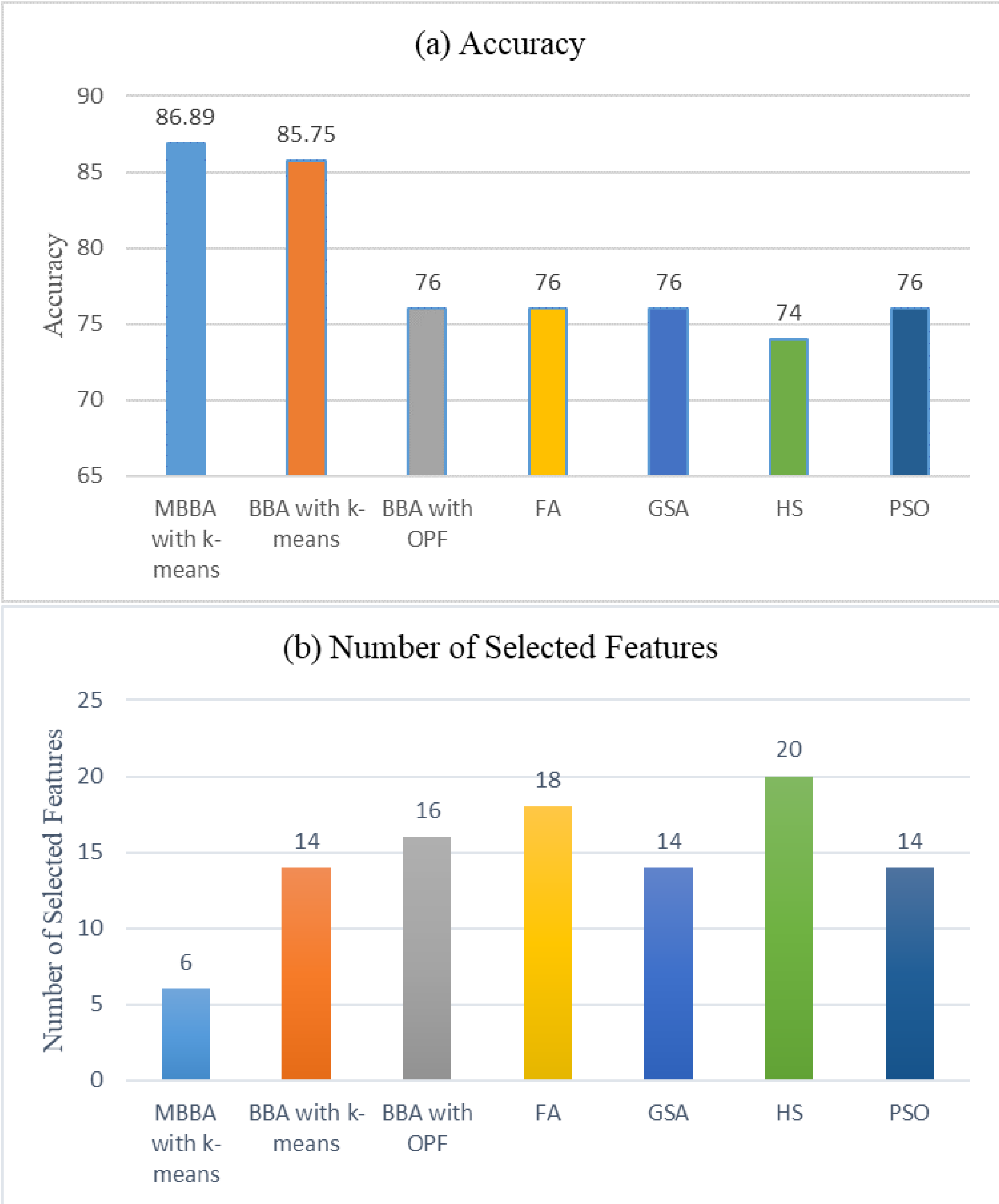

Figure 6.3: Results of Ionosphere Dataset.

Figure 6.4(a) compares the accuracy of SVMGuide1 by the MBBA with the k-means against other methods. Here, the proposed method gives 69.93% accuracy, which is less than BBA with the k-means, BBA with OPF, FA, GSA, HS, and PSO.

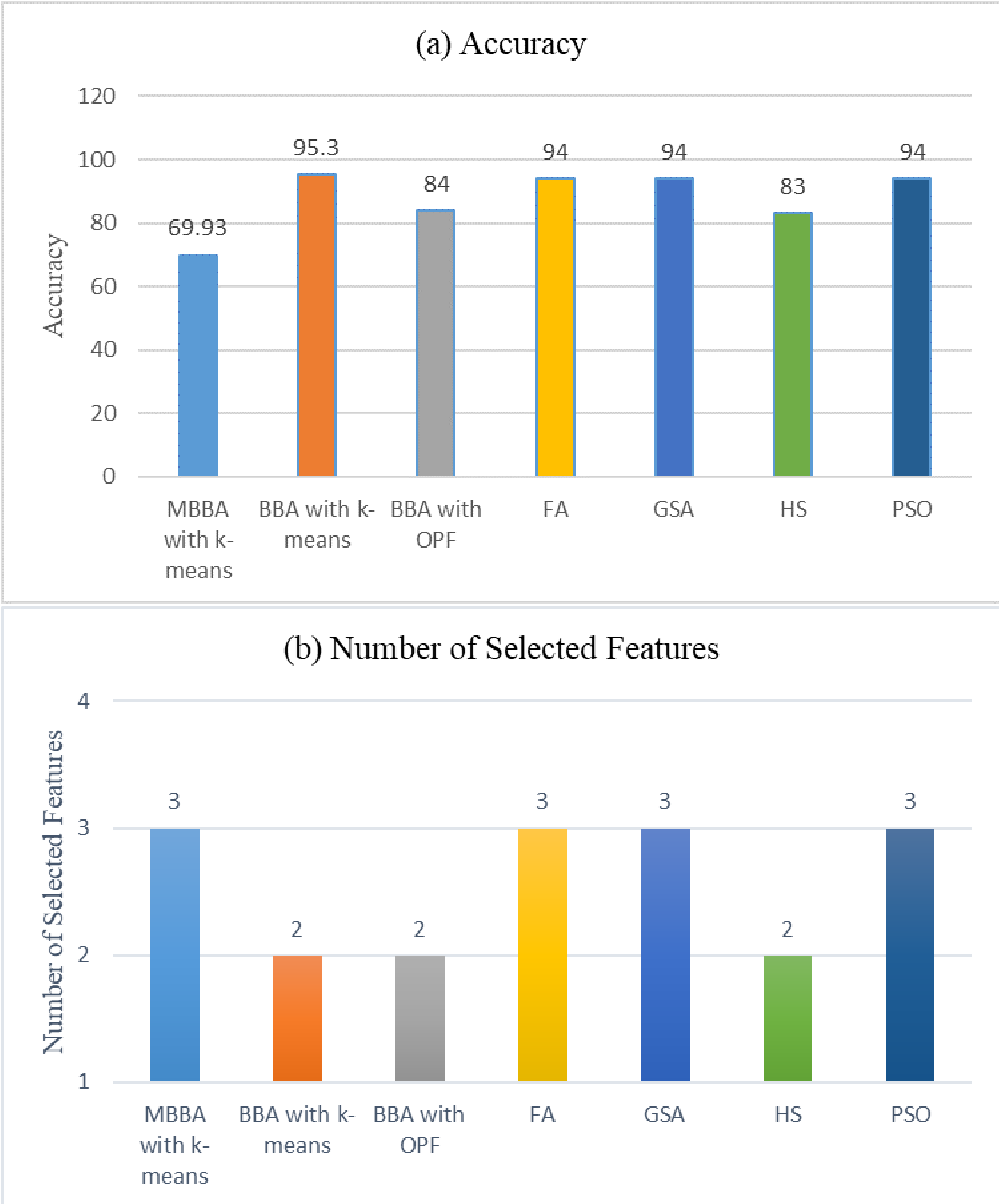

Figure 6.4: Results of SVMGuide1 Dataset.

Figure 6.4(b) compares the number of the selected features over the same dataset. The proposed method selected three features, which is the same as FA, GSA, and PSO. But, BBA with the k-means, BBA with OPF and HS selected fewer features, which is two.

The proposed method performed better in Wisconsin Brest Cancer, Splice and Ionosphere datasets in terms of accuracy and the number of the selected features compared to the other methods. The proposed method did not perform better with SVMGuide1 dataset. A possible reason behind this could be the lower number of features present in the SVMGuide1 dataset. It shows the the explorative quality of the proposed algorithm rather than being exploitative.

Chapter 7

Conclusion and Future Work

7.1 Conclusions

Solving feature selection problems to improve the performance of the machine-learning algorithms and provide better results is an important research topic. In this thesis, a modification has been done over the binary version of the Bat Algorithm (BA) called the Binary Bat Algorithm (BBA). The modification has been done by controlling the parameters (frequency, pulse rate, loudness). Later, the wrapper-based approach and the k-means clustering algorithm are used along with the proposed modified BBA (MBBA) to solve the feature selection problem in unsupervised learning where the dataset is provided without class label. The sum of squared error from k-means clustering worked as a fitness function in MBBA. The proposed method determines a new solution by selecting the loudness and pulse rate randomly, which improves the exploration and exploitation technique and also gives the best solution without being trapped in a local optimum, which shows that the proposed method produces better results than several other algorithms. Also selecting the number of bats based on half of the instances makes the modified algorithm perform faster. An experiment was performed to evaluate MBBA against several metaheuristic algorithms to identify how well the proposed method is performing. The results in the experiment show that the proposed method generates better quality solutions. What is also notable is that this method outperformed the existing methods for three out of four datasets used in the experiment by producing results with higher accuracy and minimal number of features in comparison with the existing methods.

7.2 Future Work

The importance of further research will increase as the feature selection problem continues to gain popularity. This thesis creates an opportunity for future work that will help to solve the feature selection problem more efficiently. Other unsupervised learning algorithms can be used to enhance performance. Swarm intelligence algorithms such as Ant Colony Optimization, Particle Swarm Optimization, Cuckoo Search, and Firefly can be combined with the Binary Bat Algorithm to make a new hybrid algorithm which can be used to solve the feature section problem. The proposed method can be applied to different real-world applications. Similarly, the performance of the proposed MBBA can also be investigated for other applications.

Bibliography

[1] Uysal, A. K. An improved global feature selection scheme for text classification. *Expert systems with Applications* **43**, 82–92 (2016).

[2] Tourassi, G. D., Frederick, E. D., Markey, M. K. & Floyd, C. E. Application of the mutual information criterion for feature selection in computer-aided diagnosis. *Medical physics* **28**, 2394–2402 (2001).

[3] Zhang, C., Li, Y., Yu, Z. & Tian, F. Feature selection of power system transient stability assessment based on random forest and recursive feature elimination. In *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 1264–1268 (IEEE, 2016).

[4] Liu, T., Liu, S., Chen, Z. & Ma, W.-Y. An evaluation on feature selection for text clustering. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, 488–495 (2003).

[5] Cheema, S., Henne, T., Koeckemann, U. & Prassler, E. Applicability of feature selection on multivariate time series data for robotic discovery. In *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 2, V2–592 (IEEE, 2010).

[6] Yang, X.-S. *Firefly Algorithms*, 111–127 (2014).

[7] Mitchell, M. *An introduction to genetic algorithms* (MIT press, 1998).

[8] Karaboga, D. & Basturk, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization* **39**, 459–471 (2007).

[9] Eberhart, R. & Kennedy, J. Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, vol. 4, 1942–1948 (Citeseer, 1995).

[10] Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 65–74 (Springer, 2010).

[11] Liang, J.-J., Suganthan, P. N. & Deb, K. Novel composition test functions for numerical global optimization. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, 68–75 (IEEE, 2005).

[12] Luke, S. Essentials of metaheuristics. lulu, 2009. *Available for free at http://cs. gmu. edu/sean/book/metaheuristics/. There is no corresponding record for this reference* (2011).

[13] Holland, J. H. *et al. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence* (MIT press, 1992).

[14] Razali, N. M., Geraghty, J. *et al.* Genetic algorithm performance with different selection strategies in solving tsp. In *Proceedings of the world congress on engineering*, vol. 2, 1–6 (International Association of Engineers Hong Kong, 2011).

[15] Goldberg, D. E. & Holland, J. H. Genetic algorithms and machine learning. *Machine learning* **3**, 95–99 (1988).

[16] Gaidhani, C. R., Deshpande, V. M. & Bora, V. N. Image steganography for message hiding using genetic algorithm. *International Journal of Computer Sciences and Engineering* **2**, 67–70 (2014).

[17] Radwan, A. A., Latef, B. A. A., Ali, A. M. A. & Sadek, O. A. Using genetic algorithm to improve information retrieval systems (2006).

[18] Zhang, G., Shao, X., Li, P. & Gao, L. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. *Computers & Industrial Engineering* **56**, 1309–1318 (2009).

[19] Shrivastava, A. & Rajawat, S. An implementation of hybrid genetic algorithm for clustering based data for web recommendation system. *Int J Comput Sci Eng* **2**, 6–11 (2014).

[20] Yang, J., Shi, X., Marchese, M. & Liang, Y. An ant colony optimization method for generalized tsp problem. *Progress in Natural Science* **18**, 1417–1422 (2008).

[21] Al-Enezi, J. R., Abbod, M. F. & Alsharhan, S. Artificial immune systems-models, algorithms and applications (2010).

[22] Bernardino, H. S., Fonseca, L. G. & Barbosa, H. J. Surrogate-assisted artificial immune systems for expensive optimization problems. In *Evolutionary Computation* (IntechOpen, 2009).

[23] Lones, M. Sean luke: essentials of metaheuristics (2011).

[24] Kennedy, J. & Eberhart, R. Particle swarm optimization, doi: 10.1109/icnn.1995.488968. In *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1942–1948 (1995).

[25] Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *science* **220**, 671–680 (1983).

[26] Griffin, D. Bats . By Phil Richardson. Published by Smithsonian Institution Press, Washington (DC), in association with The Natural History Museum, London. $16.95 (paper). 112 p; ill.; index. ISBN: 1–58834–020–1. 2002. . *The Quarterly Review of Biology* (2003).

[27] Yang, X.-S. Bat algorithm: literature review and applications. *arXiv preprint arXiv:1308.3900* (2013).

[28] Khan, K., Nikov, A. & Sahai, A. A fuzzy bat clustering method for ergonomic screening of office workplaces. In *Third International Conference on Software, Services and Semantic Technologies S3T 2011*, 59–66 (Springer, 2011).

[29] Reddy, V. U. & Manoj, A. Optimal capacitor placement for loss reduction in distribution systems using bat algorithm. *IOSR journal of Engineering* **2**, 23–27 (2012).

[30] Tamiru, A. & Hashim, F. Application of bat algorithm and fuzzy systems to model exergy changes in a gas turbine. In *Artificial intelligence, evolutionary computing and metaheuristics*, 685–719 (Springer, 2013).

[31] Yang, X.-S. Bat algorithm for multi-objective optimisation. *arXiv preprint arXiv:1203.6571* (2012).

[32] Lin, J.-H., Chou, C.-W., Yang, C.-H., Tsai, H.-L. *et al.* A chaotic levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. *Computer and Information Technology* **2**, 56–63 (2012).

[33] Gandomi, A. H. & Yang, X.-S. Chaotic bat algorithm. *Journal of Computational Science* **5**, 224–232 (2014).

[34] Mirjalili, S., Mirjalili, S. M. & Yang, X.-S. Binary bat algorithm. *Neural Computing and Applications* **25**, 663–681 (2014).

[35] Duda, R. O., Hart, P. E. & Stork, D. G. *Pattern classification and scene analysis*, vol. 3 (Wiley New York, 1973).

[36] Jamil, M., Zepernic, H. & Yang, X. Improved bat algorithm for global optimization. *Applied Soft Computing* (2013).

[37] Yilmaz, S. & Kucuksille, E. U. Improved bat algorithm (iba) on continuous optimization problems. *Lecture Notes on Software Engineering* **1**, 279 (2013).

[38] Komarasamy, G. & Wahi, A. An optimized k-means clustering technique using bat algorithm. *European Journal of Scientific Research* **84**, 263–273 (2012).

[39] Faritha Banu, A. & Chandrasekar, C. An optimized approach of modified bat algorithm to record deduplication (2013).

[40] Khan, K. & Sahai, A. A comparison of ba, ga, pso, bp and lm for training feed forward neural networks in e-learning context. *International Journal of Intelligent Systems and Applications* **4**, 23 (2012).

[41] Khan, K., Sahai, A. & Campus, A. A fuzzy c-means bi-sonar-based metaheuristic optomization algorithm. *IJIMAI* **1**, 26–32 (2012).

[42] Marichelvam, M. & Prabaharam, T. A bat algorithm for realistic hybrid flowshop scheduling problems to minimize makespan and mean flow time. *ICTACT Journal on soft Computing* **3**, 428–433 (2012).

[43] Bora, T. C., Coelho, L. d. S. & Lebensztajn, L. Bat-inspired optimization approach for the brushless dc wheel motor problem. *IEEE Transactions on magnetics* **48**, 947–950 (2012).

[44] Akhtar, S., Ahmad, A. & Abdel-Rahman, E. M. A metaheuristic bat-inspired algorithm for full body human pose estimation. In *2012 Ninth Conference on Computer and Robot Vision*, 369–375 (IEEE, 2012).

[45] Zhang, J. W. & Wang, G. G. Image matching using a bat algorithm with mutation. In *Applied Mechanics and Materials*, vol. 203, 88–93 (Trans Tech Publ, 2012).

[46] Lemma, T. A. & Hashim, F. B. M. Use of fuzzy systems and bat algorithm for exergy modeling in a gas turbine generator. In *2011 IEEE Colloquium on Humanities, Science and Engineering*, 305–310 (IEEE, 2011).

[47] Bahmani-Firouzi, B. & Azizipanah-Abarghooee, R. Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm. *International Journal of Electrical Power & Energy Systems* **56**, 42–54 (2014).

[48] Hasançebi, O., Teke, T. & Pekcan, O. A bat-inspired algorithm for structural optimization. *Computers & Structures* **128**, 77–90 (2013).

[49] Kashi, S., Minuchehr, A., Poursalehi, N. & Zolfaghari, A. Bat algorithm for the fuel arrangement optimization of reactor core. *Annals of Nuclear Energy* **64**, 144–151 (2014).

[50] Ge, H., Zhu, X. & Li, Y. Feature selection with variable interaction. *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE* **11**, 3983–3991 (2014).

[51] Arauzo-Azofra, A., Molina-Baena, J., Jiménez-Vílchez, A. & Luque-Rodriguez, M. Using individual feature evaluation to start feature subset selection methods for classification. In *ICAART (2)*, 607–614 (2017).

[52] Hall, M. A. & Smith, L. A. Feature subset selection: a correlation based filter approach (1997).

[53] Gu, Q., Li, Z. & Han, J. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725* (2012).

[54] Bradley, P. S. & Mangasarian, O. L. Feature selection via concave minimization and support vector machines. In *ICML*, vol. 98, 82–90 (1998).

[55] Guyon, I. & Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research* **3**, 1157–1182 (2003).

[56] Kim, Y., Street, W. N. & Menczer, F. Feature selection in unsupervised learning via evolutionary search. In *KDD*, 365–369 (2000).

[57] Tabakhi, S., Moradi, P. & Akhlaghian, F. An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence* **32**, 112–123 (2014).

[58] Forman, G. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research* **3**, 1289–1305 (2003).

[59] Song, D. & Tao, D. Biologically inspired feature manifold for scene classification. *IEEE Transactions on Image Processing* **19**, 174–184 (2010).

[60] Bins, J. & Draper, B. A. Feature selection from huge feature sets. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 159–165 (IEEE, 2001).

[61] Muštra, M., Grgić, M. & Delač, K. Breast density classification using multiple feature selection. *automatika* **53**, 362–372 (2012).

[62] Liu, C., Jiang, D. & Yang, W. Global geometric similarity scheme for feature selection in fault diagnosis. *Expert Systems with Applications* **41**, 3585–3595 (2014).

[63] Dessì, N., Pascariello, E. & Pes, B. A comparative analysis of biomarker selection techniques. *BioMed research international* **2013** (2013).

[64] Abusamra, H. A comparative study of feature selection and classification methods for gene expression data of glioma. *Procedia Computer Science* **23**, 5–14 (2013).

[65] Fister, I., Fister Jr, I., Yang, X. & Brest, J. On the representation of individual s using quaternions in swarm intelligence and evolutionary computation. *IEEE Trans. Evol. Computation,(2013, submitted)* (2013).

[66] Kong, X., Gao, L., Ouyang, H. & Li, S. A simplified binary harmony search algorithm for large scale 0–1 knapsack problems. *Expert Systems with Applications* **42**, 5337–5355 (2015).

[67] Sobeyko, O. & Mönch, L. Heuristic approaches for scheduling jobs in large-scale flexible job shops. *Computers & Operations Research* **68**, 97–109 (2016).

[68] Rani, A. S. S. & Rajalaxmi, R. Unsupervised feature selection using binary bat algorithm. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 451–456 (IEEE, 2015).

[69] Theodoridis, S., Pikrakis, A., Koutroumbas, K. & Cavouras, D. *Introduction to pattern recognition: a matlab approach* (Academic Press, 2010).

[70] Shahbaba, M. & Beheshti, S. Mace-means clustering. *Signal Processing* **105**, 216–225 (2014).

[71] Kwedlo, W. A clustering method combining differential evolution with the k-means algorithm. *Pattern Recognition Letters* **32**, 1613–1621 (2011).

[72] Tay, B., Hyun, J. K. & Oh, S. A machine learning approach for specification of spinal cord injuries using fractional anisotropy values obtained from diffu-

sion tensor images. *Computational and mathematical methods in medicine* **2014** (2014).

[73] Witten, I. H., Frank, E., Hall, M. A. & Pal, C. J. *Data Mining: Practical machine learning tools and techniques* (Morgan Kaufmann, 2016).

[74] Visa, S., Ramsay, B., Ralescu, A. L. & Van Der Knaap, E. Confusion matrix-based feature selection. *MAICS* **710**, 120–127 (2011).

[75] Thinsungnoena, T., Kaoungkub, N., Durongdumronchaib, P., Kerdprasopb, K. & Kerdprasopb, N. The clustering validity with silhouette and sum of squared errors. *learning* **3**, 7 (2015).

[76] Han, J., Pei, J. & Kamber, M. *Data mining: concepts and techniques* (Elsevier, 2011).

[77] Powers, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation (2011).

[78] de Moura Oliveira, P., Oliveira, J. & Cunha, J. B. Trends in gravitational search algorithm. In *International Symposium on Distributed Computing and Artificial Intelligence*, 270–277 (Springer, 2017).

[79] Askarzadeh, A. & Rashedi, E. Harmony search algorithm: Basic concepts and engineering applications. In *Intelligent Systems: Concepts, Methodologies, Tools, and Applications*, 1–30 (IGI Global, 2018).

[80] Kennedy, J. Particle swarm optimization. *Encyclopedia of machine learning* 760–766 (2010).