Towards Optimized K means Clustering using Nature-Inspired Algorithms for Software Bug Prediction

Tameswar Kajal¹, Geerish Suddul² and Kumar Dookhitram³

¹ University of technology

Received: 1 January 1970 Accepted: 1 January 1970 Published: 1 January 1970

Abstract

In today’s software development environment, the necessity for providing quality software products has undoubtedly remained the largest difficulty. As a result, early software bug prediction in the development phase is critical for lowering maintenance costs and improving overall software performance. Clustering is a well-known unsupervised method for data classification and finding related patterns hidden in datasets.


1 Introduction

In an era of technological disruption, the demand for software adoption has accelerated. They are a part of our society and play an important role in shaping it. Our modern society is becoming increasingly reliant on complex software systems. Thus, it is critical to build reliable and trustworthy systems in a cost-effective and timely manner. The presence of defective modules in a software drives up development and maintenance expenses, leading to customer dissatisfaction. The need for quality assurance has inevitably remained the biggest challenge in today’s software development environment. Hence, software bug prediction is an important task to help developers locate bugs more efficiently.

Software bug prediction is an imperative task in Software Development Life Cycle (SDLC) as it pertains to the overall success of software. One method in this direction is to use machine learning (ML) methods to predict defects in software. In addition, implementing this method earlier in the SDLC process enhances quality of the product and lowers the cost of software maintenance. Many researchers have applied different theories and methodologies in the field of software bug prediction. Two things are clear from the literature when it comes to defect prediction. Initially, no single prediction approach dominates (Lessmann et al., 2008), and next, the employment of various set of data, data pre-processing, validation systems, and performance statistics makes it challenging to make sense of the multiple prediction outcomes (Myrtveit et al., 2005). There are two common ML model used for prediction based on dataset availability. The first, known as supervised approach, in which a software defect prediction model is built from training set of data and then tested on a testing dataset. Secondly, unsupervised approach, in which the defect prediction model for software is built from scratch using the present testing dataset without training the dataset.

Clustering algorithms have been commonly used to evade the lack of training datasets available being a constraint. Cluster analysis groups things into clusters based on their similarity to create a visual representation of data (Jain and Dubes, 1998). As pointed out by Kaur (2016), one of the better instances of unsupervised learning is K-means clustering. Clustering is beneficial because it makes it easier to obtain or locate relevant information at a faster rate. Among the different clustering approaches that already exist, the K-means methodology is obviously fairly popular. (Gayathri et al., 2015). The preliminary values of the initial centroids, which are generated randomly each time the algorithm is run, have a significant impact on the performance of k-means. K-means frequently fall into local optima that produce poor clustering results. Obtaining a globally optimal clustering result involves a time-consuming, exhaustive approach that tests all partitioning choices. A heuristic
3 RELATED WORKS

3 Related Works

K-means clustering is a well-known partitioned clustering algorithm that has been used in a variety of applications. In the literature, several variations of K-means have been proposed to improve its performance for the broad clustering problem. Annong et al. (2012) studied the integration of bio-inspired optimization methods into K-means clustering for software bug prediction in order to assess clustering performance. The main optimization algorithms tested include the Firefly algorithm, Artificial Bee Colony, Particle Swarm Optimization (PSO), and Harmony Search. In addition, Catal et al. (2009) used the x-means clustering algorithm to create faulty and non-faulty clusters based on software metrics. Lines of code, cyclomatic complexity, and metrics are the operators. If the metric values are complex than the threshold, the software entity is predicted to be defective, and vice versa. Almayyan, 2021 used dataset from the NASA repository and used three clustering algorithms tested include the Firefly algorithm, Artificial Bee Colony, Particle Swarm Optimization (PSO), and Harmony Search. In order to evaluate different feature selection algorithms. The Farthest First clustering algorithm was found to be effective in predicting software faultiness, and Bat and Cuckoo were found to be useful in comparison to all other metaheuristic algorithms.

Jensi and Jiji, 2015). In addition, only a few bio-inspired optimization methods that are integrated with K-means are provided in the previous studies. Only 7 of the 28 NIAs hybridized with K-means (Genetic Algorithm, Particle Swarm Optimization, Bat Algorithm, Artificial Bee Colony, Differential Evolution, Harmony Search, and Symbiotic Organism Search) dedicated their hybridization to solving automatic clustering problems, accounting for 20.6 percent of the total (Ikotun et al., 2021). In general, it can be seen that the rate of publishing on K-means hybridization with specific NIAs is minimal. More research is needed in this area to see if there are any other ways to improve the performance of the existing hybridization algorithm. This suggests that combining K-means with these other NIAs to solve automatic clustering problems should be investigated.

The purpose of this research is to look into the mechanics of incorporating certain NIAs into the K-means clustering algorithm. The optimization function adds to the existing best solution by progressively improving it with a new solution from an unknown fragment of the search space. When a new solution is identified to be better than the present one, the searching agents replace the solutions and continue searching until some stopping criteria are fulfilled.
5 Methodology a) K means Clustering Algorithm

The K-means clustering algorithm is a partitioned clustering technique that divides a dataset into k number of clusters using a certain fitness measure. Due to the large amount of data objects in real-world datasets, distributing data items into appropriate clusters to obtain an ideal cluster outcome is computationally expensive and time-consuming (Iktotu et al.2021).

Given a dataset \( X = \{x_i\} \), where \( i = 1, 2, \ldots, n \) of d-dimension data points of size n, \( X \) is partitioned into \( k \) clusters such that \( J(c_k) = \sum \|x_i - \mu_k\|^2 \) (1)

With the objective function: minimize the sum of the square error over all the \( k \) clusters. That is, minimize \( J(C) \)

\[ J(C) = \sum_{i=1}^{n} \min_{k} \| x_i - \mu_k \|^2 \] (2)

b) Nature-inspired algorithms (NIAs) Nature-inspired computation has gained popularity in the previous two decades and has been used in practically every field of research and engineering (??Yang et al.2013). NIAs are global optimization strategies for solving difficult real-world issues (Okwu et al. 2020). NIAs have successfully provided suboptimal solutions to automatic clustering problems in a reasonable amount of time (Hruschka et al. 2009). The population is used for the exploration of search space in the nature-inspired metaheuristic, ensuring a higher possibility of finding optimal cluster partitions (Nanda and Panda, 2014). It has been discovered that combining K-means with NIAs for automatic clustering improves the performance of algorithms when dealing with cluster analysis. In most circumstances, the automatic cluster number determination aids in the selection of near-optimal starting cluster centroids for the clustering process rather than the normal random selection (Zhou et al. 2017).

6 c) Combination of k-means with Nature-Inspired Algorithms (NIAs)

Clustering using NIAs is now as simple as assigning combinations of centroids to the searching agents, allowing them to heuristically find the best answer. Though the specifics of conducting a heuristic search vary depending on which nature-inspired optimization algorithm technique is used, the initialization stage and the finishing step, where the quality of the discovered solution is evaluated as a stopping condition, are both comparable.

\( S \) is defined as the solution space that contains a finite number of \( x_i \), where \( i \) is the solution’s index, in the initialization construct. The search agents represent the solutions \( x \), each of which holds a set of centroids, regardless of the types of bio-inspired optimization methods used. Typically, a large population of searching agents, \( N \), is utilized to collaboratively search for the best feasible cluster configurations (as expressed by the locations of the optimal centroids). \( K \) is the number of clusters that must be formed, which is generally a user-defined figure. \( D \) is the dimension of the search space, which is the number of attributes a data point possesses.

To find the optimal configuration of centroids we let \( c_{j,v} \) be the centroids at the \( j \) th cluster and the \( v \) th attribute. To obtain the centroid location, the following formula is used: \( c_{j,v} = \sum c_{j,v} / \sum i \)

\[ c_{j,v} = \frac{\sum_i x_i \cdot c_{j,v}}{\sum_i x_i} \] (3)

In our concept, the matrix \( c_{j,v} \) contains all of the cluster centers and is a two-dimensional matrix with \( K \times D \) characteristics. \( F(cen) = k \) \( j=1 \) \( S_i=1 \) \( W_{i,j} \) \( K\times D = \sum_{(X_{iv}, -cen_{j,v})}^2 \) (5)

The calculation method loops \( K \times D \) times to analyze the values of all the attributes of \( x \) in each cluster \( v \) to calculate the distance between each \( x \) and the centroid.

Cluster centers can be designated by data points. For example, in a two-cluster clustering task, the objective function requires three variables. As a result, there are three dimensions.

Three variables, and hence three-dimensional spaces, are required, and the \( x \) th data point may be written as \( x_i = (i, x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}, x_{i,5}, x_{i,6}) \). The clustering strategy can be formulated as follows:

\( i,j = \min_k \| x_i - \mu_k \| \) (6)

Sets of functional parameters must be defined in order to execute the bio-inspired optimization algorithms. Despite the fact that some of their parameters are shared, each set of parameters for the hybrid bio-inspired clustering algorithms is designed independently. The six models investigated are K means with Genetic Algorithm, K means with Bat algorithm, K means with Ant colony algorithm, K means with Cuckoo Search Algorithm, K means with Firefly Algorithm and K means with Coral reefs algorithm. The most significant variations are in how the global optimal exploration is carried out for all these algorithms. The evaluation stage comes right after the exploration construct, and it compares if the new solution is better than the current best one.

7 d) Genetic Algorithm

Genetic Algorithm (Ga) are randomized heuristic search algorithms that are based on natural selection and genetic principles (Goldberg, 1989). The genetic operators used in the combination of K-means and GA are selection, distance-based mutation, and the K-means operator. The parameters have been set according to the
study of Bouhmala et al. 2013. P (0) is chosen at random as the starting population. Each allele in the population can be given a cluster number from the uniform distribution over the set \( \{ 1, \ldots, K \} \) at random.

According to the distribution given by, the selection operator selects a chromosome from the preceding population at random as follows:
\[
P(s_{i}) = f(s_{i}) / N \quad \text{if} \quad s_{i} \neq s_{j}
\]

The possibility of solutions surviving in the future population is ranked in the current population. Each solution in the population must be assigned a figure of merit or a fitness value:
\[
F(s_{W}) = \begin{cases} 
0 & \text{if } g(s_{W}) > 0 \\
g(s_{W}) & \text{otherwise}
\end{cases}
\]

The parameters have been set according to the study ??Huang and Ma, 2020).

The frequency, velocity, and position for each bat are initialized. The virtual bats' movement is described by updating their velocity and position using the equations below for each time step \( t \), where \( T \) is the iteration limit.
\[
V_{i}^{t+1} = V_{i}^{t} + \Delta V_{i}^{t} f_{i}^{t+1} = f_{i}^{t} + V_{i}^{t + X*} f_{i}^{t+1} = X_{i}^{t} + V_{i}^{t+1}
\]

A random number is generated when the bat positions are updated; if the random number is greater than the pulse emission rate, a new location is formed around the current best solutions, as shown in the equation below.

\[
X_{new} = X_{old} + EA_{t}
\]

8 f) Ant Colony Optimization (ACO)

The ACO heuristic was inspired by investigations of ant foraging behavior in real colonies, which indicated that ants can often find the shortest path between food source and nest [Zheng et al. 2003]. The parameters have been set according to the study ??Tang et al. 2012).

When the ant moves from \( i \) to \( j \), the path node at the start can set as \( A = \{ 0, 1, \ldots, n-1 \} \). This reflects the role of pheromones accumulated by ants during exercise during ant migration and reveals the relative relevance of the trajectory. The larger \( ? \) is, it indicates the high probability for subsequent ants to choose this path.

The probability of the ant moving from \( i \) to \( j \) is computed using the following formula:
\[
P_{ij}^{t} = \frac{r_{ij}^{t}}{\sum_{j=1}^{N} r_{ij}^{t}}
\]

9 g) Firefly Algorithm (FA)

Firefly algorithm is a very strong technique for solving restricted optimization and NP-hard problems (Apostolopoulos and Vlahos, 2011). The parameters have been set according to the study ??Tang et al. 2012).

The attractiveness of a firefly \( i \) on a firefly \( j \) is determined by the degree of the firefly \( i \)'s brightness and the distance \( r_{ij} \) between the firefly \( i \) and the firefly \( j \), as shown below:
\[
I_{i} = I_{s} / r_{ij}^{2}
\]

Consider the case when there are \( n \) fireflies and the solution for firefly \( i \) is \( x_{i} \). The brightness of the firefly \( i \) is linked to the objective function \( f(x_{i}) \).
\[
I_{i} = f(x_{i})
\]

Each firefly has an attraction value, and the less dazzling (attractive) one is drawn to the brighter one and transferred there. The attractiveness value \( ? \) is relative based on the distance between fireflies. Where pheromone is \( ? \), which is a constant that represents weight. The time of iteration is \( Nc \) and the initial setting is \( ? \). The predicted heuristic factor is \( \alpha \), which demonstrates the relevance of visibility relative to other factors. It also represents the significance of the heuristic component in the entire path of the ant’s movement.

\[
E_{i} = ? / r_{ij}^{2}
\]

Where \( V_{it} \) and \( X_{it} \) are the velocity and position at time \( t \), \( V_{i}^{t+1} \) and \( X_{i}^{t+1} \) are the velocity and position at time \( t+1 \), and is a random number between 0 and 1. Where \( ? \) is the firefly attraction value at \( r = 0 \) and \( ? \) is the media light absorption coefficient.

Where \( E \) is a random number \( A \) represents the average loudness of all bats at time \( t \).

An initial population of \( n \) nests is randomly generated at the positions, \( X = \{ x_{0} \} \), to evaluate the objective values to find the current global best solution.

The new position is updated accordingly by performing a \( L?vy \) flight:
\[
x_{i}^{t+1} = x_{i}^{t} + \sum_{j=1}^{N} c_{0} x_{j}^{t} + \sum_{j=0}^{N} \alpha x_{j}^{t}
\]

10 i) Coral Reefs Optimization Algorithm (CRO)

CRO is another nature-inspired algorithm, based on an artificial simulation of the process of coral reef formation and reproduction (Sanz et al. 2014). The CRO algorithm has never been utilized in the realm of software bug detection to our knowledge. Corals reproduce at each iteration step in the CRO algorithm, producing new individuals. The parameters have been set according to the study (Medeiros et al., 2015).

By allocating a coral to each square \( (i, j) \), the CRO algorithm generates a \( N \times M \) square grid in which each square \( (i, j) \) may represent an alternate solution to a problem (or colony of corals). The formation of coral is the second phase. After three phases, the entire collection of existing corals in the reef is graded according to their level of healthiness (broadcast spawning, brooding, and larvae setting).

11 j) Particle Swarm Optimization (PSO)

The behavior of particles in a swarm is the central concept of the PSO. Each particle has its own location in a multidimensional space and communicates with the others. To move about in space, the particles employ social
The hardware configuration used for all experiments in this study is as follows: Corei7-6500U CPU @2.50 GHz and Groza, 2014. The parameters have been set according to the study (Rana et al., 2010).

The inertia weight balances the algorithm’s local and global search abilities. The proportional contribution of the prior velocity to the current velocity is defined by the inertia weight. $v_{i k+1} = w v_{i k} + c1 \text{ rand } (p_{best} - x_{i k}) + c2 \text{ rand } (g_{best} - x_{i k})$ (18) $X_{i k+1} = X_{i k} + v_{i k+1}$ (19)

### 12 k) Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) is a simple, population-based, flexible, and derivative-free metaheuristic optimization method that intelligently avoids stagnation in local optima spots of the search space. It simulates the social behaviors of grey wolves in the aspects of their hierarchical leadership and hunting movement. Grey wolves’ leadership and haunting mechanism help to design a new metaheuristic algorithm with three steps: searching prey, encircling prey, and attacking prey.

During the GWO operation, the position of the wolves is continuously updated, with appropriate mathematical formulas (Hou et al., 2022). The parameters have been set according to the study (Wang et al., 2019).

### 13 Proposed Method

The purpose of clustering is to discover a proper set of centroids using the metaheuristic of the nature-inspired method as a guide. The metaheuristic will always insist on centroids being moved in a progressive manner in each phase, with the goal of finding the best grouping. The ideal group’s ultimate result should be that the data points inside each cluster are closest to their centroid. During the search, the centroids move around in the search space, following the swarming pattern of the nature-inspired optimization method, until no further progress is seen. It comes to a halt when there is no other possible relocation that will yield a better result. Along with the success of employing nature-inspired metaheuristic algorithms to solve automatic clustering problems, it has been discovered that combining two or more metaheuristics for the same objective improves clustering performance. The performance of hybrid algorithms, according to Nanda and Panda 2014, is superior to that of separate algorithms in terms of robustness, effectiveness, and accuracy.

### 14 Dataset and Data Processing

The dataset was collected from the online PROMISE repository. AR1, AR3, AR4, AR5, AR6, KC1, KC2, JM1, CM1, PC1 and PC5 were used respectively. With reference to the paper, Shepperd et al. 2013, data cleaning is mandatory before using any datasets available. Indeed, we noted a huge class imbalance issue with the available datasets (faulty, non-faulty), and all data inconsistencies, missing and null values were removed. Each dataset selected represents a NASA software system that includes various metrics. Each dataset is made up of a number of software modules and attributes. Modules with defects are classified as prone to faults, whereas those without defects are classified as non-prone, and thus considered to be global optimum clusters. The suggested method uses the k-means algorithm together with a range of NIAs for software bug prediction. By adding an exploration function to the k-means algorithm, the combination of these strategies may improve the model. The exploration function improves the existing solution by examining regions outside of its immediate vicinity, and if a new, better solution than the current best one is discovered, the search agents will move toward it. The exploring procedure will continue until certain stopping criteria are met. Nature-inspired algorithms are metaheuristic algorithms, which means they have the ability to explore the combinatorial search space heuristically rather than exhaustively. The integration methods are based on representing the search agents as a combination of centroid locations, then the search agents explore the search space for the best solution.

Where $> 0$ denotes the step size, which should be connected to the problem’s scales. In most circumstances, we can use $\beta = 1$. classified as non-fault prone. For the training purpose, the entire dataset is used except for the last column (output column), only columns consisting of numerical values were considered.

### 15 Evaluation a) Experimental Setup

The main goal of this research is to demonstrate the utility of the k-means algorithm with different NIAs, which we accomplished using Tensorflow to train the model. Tensorflow is an open-source machine learning platform to build and deploy prediction models. Google Colab was also used to run the results, which allowed the code to run with no configuration and free GPU access. Each dataset is performed 10 times in the trials to find the average CPU time and objective function values/best fitness value.

The clustering results of the new hybrid clustering algorithms are compared to the K-means, which serve as a benchmarking reference. The full dataset is used for training, and cluster formation is referred to until perfection is attained using the entire set of data. The ultimate clustering result’s quality is determined by each cluster’s integrity, which is represented by the objective function’s final fitness value.

The hardware configuration used for all experiments in this study is as follows: Corei7-6500U CPU @2.50 GHz, 2.60 GHz, Windows 10, 64-bit operating system, x64 based processor, RAM: 8 GB DDR4, and Hard Disk: SSD.
16 b) Performance Evaluation Measures

In order to assess the effectiveness of combining the k-means algorithm and optimization algorithms in the prediction of software bugs, the evaluation metrics, accuracy and F-measure have been calculated accordingly as shown in the Equation (1):%

\[ \text{F-measure} = \frac{TP + TN}{TP + TN + FN + FP} \]

Where TP = true positive, TN = true negative, FN = false negative and FP = false positive.

On the other hand, the external metric used to determine the accuracy of the clustering findings, known as the F-measure, is also computed.

The F-measure, which is the average of precision and sensitivity performance, is calculated as follows:

\[ \text{F} = \frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \]

Where P refers to precision and sensitivity is calculated by finding the non-defective modules that were accurately categorized. From the table above, K-means clustering is optimized using the various NIAs. We can see that all of the proposed algorithms perform better than the traditional standalone k-means algorithm. K-means appears to take the shortest computation time in any of the tests, maybe because it stops early in local optima (Table ??). This is evident from the accuracy obtained from the table above. NIAs speed up the process of clustering centroids and illustrate that all partitioning clustering methods can be linked with the natural search process to prevent local optima. Secondly, simple Kmeans were applied to the robust nature of GA, which shows adequate prediction accuracy for all datasets. Even though GA may converge to the global optimum due to mutation, GA faces the issue in terms of computational challenges. The application of k means with the Bat algorithm apparently yields the same accuracy. This hybrid algorithm improves the convergence speed of BA and helps the k means algorithm independent of the initial centers. Next, K means is combined with PSO. The PSO method is used to start the process because of its fast convergence, and then the K-Means algorithm is used to refine the PSO algorithm’s outcome to near-optimal solutions. The hybridization of these two methods yields effective results in terms of efficiency and precision. The PSO algorithm can be used to generate good initial cluster centroids for the K-Means.

17 VII.

18 Results and Discussions

19 Practical Implications

Metaheuristics algorithms have shown to be effective optimizers. This research found that each of the hybrid K means based-nature inspired optimization algorithm models outperformed the standalone K means algorithm in terms of accuracy and F1 score. Following the intrinsic limitations of K-means design and the virtues of Nature-inspired optimization techniques, it seems feasible to integrate them, allowing them to complement and function together. The algorithms’ successful integration gives reason to believe that more advanced optimization mining techniques can be developed. This study can be used as a roadmap for researchers who want to incorporate other new emerging NIAs into improved clustering methods in the field of software bug detection.

20 IX.

21 Conclusion and Future Works

Prediction of defect-prone software modules is an important goal in software engineering. The traditional clustering algorithm usually gets trapped in the problem of local optima. As a result, the natureinspired method provides an alternative technique for solving clustering problems using its searching capabilities. This study’s main contribution is combining the clustering algorithm with the different NIAs for software bug detection. To the authors’ knowledge, only PSO, Cuckoo, Bat, and GWO (Grey Wolf Optimizer) algorithms were applied with clustering algorithms for software bug detection (Almayyan, 2021). The results are improved significantly when clustering algorithms are combined with bio-inspired optimization methods, apparently for the hybrid model of k means clustering withCoral reefs algorithm, achieving an accuracy of 96%. For future work, this work can be replicated with other related datasets for the analysis of bug prediction in software.

---

1

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Modules</th>
<th>Defective modules</th>
<th>Software metrics (Attributes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR1</td>
<td>121</td>
<td>9</td>
<td>29</td>
</tr>
<tr>
<td>AR3</td>
<td>63</td>
<td>8</td>
<td>29</td>
</tr>
<tr>
<td>AR4</td>
<td>107</td>
<td>20</td>
<td>29</td>
</tr>
<tr>
<td>AR5</td>
<td>36</td>
<td>8</td>
<td>29</td>
</tr>
<tr>
<td>AR6</td>
<td>101</td>
<td>15</td>
<td>29</td>
</tr>
<tr>
<td>KC1</td>
<td>2109</td>
<td>1783</td>
<td>22</td>
</tr>
<tr>
<td>KC2</td>
<td>522</td>
<td>107</td>
<td>21</td>
</tr>
<tr>
<td>JM1</td>
<td>7782</td>
<td>1672</td>
<td>21</td>
</tr>
<tr>
<td>CM1</td>
<td>327</td>
<td>42</td>
<td>37</td>
</tr>
<tr>
<td>PC1</td>
<td>705</td>
<td>61</td>
<td>37</td>
</tr>
<tr>
<td>PC5</td>
<td>1711</td>
<td>471</td>
<td>38</td>
</tr>
</tbody>
</table>

Figure 1: Table 1:

2

<table>
<thead>
<tr>
<th>Datasets</th>
<th>AR1</th>
<th>AR3</th>
<th>AR4</th>
<th>AR5</th>
<th>AR6</th>
<th>KC1</th>
<th>KC2</th>
<th>JM1</th>
<th>CM1</th>
<th>PC1</th>
<th>PC5</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-Means</td>
<td>88.90</td>
<td>88.00</td>
<td>89.01</td>
<td>88.85</td>
<td>88.43</td>
<td>89.10</td>
<td>89.00</td>
<td>88.80</td>
<td>89.00</td>
<td>89.19</td>
<td>89.99</td>
</tr>
<tr>
<td>K-Means +GA</td>
<td>90.50</td>
<td>90.58</td>
<td>91.28</td>
<td>91.55</td>
<td>90.11</td>
<td>90.00</td>
<td>90.54</td>
<td>90.53</td>
<td>91.25</td>
<td>90.00</td>
<td>90.05</td>
</tr>
<tr>
<td>K-Means +BAT</td>
<td>90.00</td>
<td>91.59</td>
<td>91.00</td>
<td>92.34</td>
<td>92.00</td>
<td>92.98</td>
<td>91.34</td>
<td>90.00</td>
<td>91.25</td>
<td>92.56</td>
<td>92.00</td>
</tr>
<tr>
<td>K-Means +PSO</td>
<td>92.50</td>
<td>92.65</td>
<td>92.87</td>
<td>93.01</td>
<td>93.00</td>
<td>92.99</td>
<td>94.10</td>
<td>92.67</td>
<td>92.89</td>
<td>93.10</td>
<td>93.58</td>
</tr>
<tr>
<td>K-Means +Coral Reefs</td>
<td>94.00</td>
<td>94.54</td>
<td>94.56</td>
<td>94.87</td>
<td>94.00</td>
<td>95.96</td>
<td>95.66</td>
<td>96.88</td>
<td>95.01</td>
<td>95.04</td>
<td>95.54</td>
</tr>
<tr>
<td>K-Means +Cuckoo</td>
<td>94.50</td>
<td>94.58</td>
<td>94.58</td>
<td>94.00</td>
<td>94.56</td>
<td>95.45</td>
<td>95.88</td>
<td>95.67</td>
<td>95.44</td>
<td>94.56</td>
<td>94.78</td>
</tr>
<tr>
<td>K-Means ACO</td>
<td>94.00</td>
<td>93.56</td>
<td>93.50</td>
<td>94.10</td>
<td>93.78</td>
<td>93.03</td>
<td>93.56</td>
<td>93.44</td>
<td>93.89</td>
<td>94.01</td>
<td>94.52</td>
</tr>
<tr>
<td>K-Means +Firefly</td>
<td>92.56</td>
<td>92.67</td>
<td>93.00</td>
<td>93.44</td>
<td>93.02</td>
<td>93.56</td>
<td>94.78</td>
<td>93.67</td>
<td>94.88</td>
<td>94.34</td>
<td>94.54</td>
</tr>
<tr>
<td>K-Means GWO</td>
<td>90.09</td>
<td>92.47</td>
<td>94.65</td>
<td>93.22</td>
<td>92.00</td>
<td>92.60</td>
<td>93.00</td>
<td>92.50</td>
<td>94.50</td>
<td>94.12</td>
<td>94.13</td>
</tr>
</tbody>
</table>

Figure 2: Table 2:
### Table 4:

<table>
<thead>
<tr>
<th>Datasets</th>
<th>AR1</th>
<th>AR3</th>
<th>AR4</th>
<th>AR5</th>
<th>AR6</th>
<th>KC1</th>
<th>KC2</th>
<th>JM1</th>
<th>CM1</th>
<th>PC1</th>
<th>PC5</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-Means</td>
<td>0.66</td>
<td>0.79</td>
<td>0.82</td>
<td>0.80</td>
<td>0.75</td>
<td>0.81</td>
<td>0.80</td>
<td>0.81</td>
<td>0.82</td>
<td>0.82</td>
<td>0.80</td>
</tr>
<tr>
<td>K-Means+GA</td>
<td>0.84</td>
<td>0.83</td>
<td>0.83</td>
<td>0.80</td>
<td>0.83</td>
<td>0.84</td>
<td>0.84</td>
<td>0.85</td>
<td>0.82</td>
<td>0.81</td>
<td>0.85</td>
</tr>
<tr>
<td>K-Means +BAT</td>
<td>0.83</td>
<td>0.81</td>
<td>0.83</td>
<td>0.86</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.87</td>
<td>0.87</td>
<td>0.85</td>
</tr>
<tr>
<td>K-Means +PSO</td>
<td>0.85</td>
<td>0.85</td>
<td>0.87</td>
<td>0.87</td>
<td>0.86</td>
<td>0.85</td>
<td>0.87</td>
<td>0.85</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
</tr>
<tr>
<td>K-Means +Coral Reefs</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.85</td>
<td>0.86</td>
<td>0.87</td>
<td>0.88</td>
<td>0.86</td>
<td>0.87</td>
<td>0.87</td>
<td>0.88</td>
</tr>
<tr>
<td>K-Means +Cuckoo</td>
<td>0.89</td>
<td>0.85</td>
<td>0.88</td>
<td>0.89</td>
<td>0.86</td>
<td>0.89</td>
<td>0.86</td>
<td>0.89</td>
<td>0.89</td>
<td>0.87</td>
<td>0.88</td>
</tr>
<tr>
<td>K-Means +ACO</td>
<td>0.84</td>
<td>0.83</td>
<td>0.86</td>
<td>0.85</td>
<td>0.84</td>
<td>0.86</td>
<td>0.85</td>
<td>0.86</td>
<td>0.85</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>K-Means +Firefly</td>
<td>0.86</td>
<td>0.85</td>
<td>0.83</td>
<td>0.87</td>
<td>0.87</td>
<td>0.85</td>
<td>0.83</td>
<td>0.85</td>
<td>0.86</td>
<td>0.88</td>
<td>0.85</td>
</tr>
<tr>
<td>K-Means+ GWO</td>
<td>0.82</td>
<td>0.82</td>
<td>0.81</td>
<td>0.86</td>
<td>0.84</td>
<td>0.83</td>
<td>0.79</td>
<td>0.85</td>
<td>0.84</td>
<td>0.84</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Figure 3: Table 4:
This study received no formal support from public, private, or not-for-profit funding organizations.
21 CONCLUSION AND FUTURE WORKS


