mathematics

MDPI

Article

Automatic Data Clustering by Hybrid Enhanced Firefly and Particle Swarm Optimization Algorithms

Mandakini Behera [1], Archana Sarangi [2], Debahuti Mishra [1], Pradeep Kumar Mallick [3], Jana Shafi [4], Parvathaneni Naga Srinivasu [5] and Muhammad Fazal Ijaz [6],*

[1] Department of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar 751030, Odisha, India
[2] Department of Electronics and Communication Engineering, Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar 751030, Odisha, India
[3] School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar 751024, Odisha, India
[4] Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdul Aziz University, Wadi Ad-Dawasir 11991, Saudi Arabia
[5] Department of Computer Science and Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada 520007, Andhra Pradesh, India
[6] Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Korea
* Correspondence: fazal@sejong.ac.kr

Abstract: Data clustering is a process of arranging similar data in different groups based on certain characteristics and properties, and each group is considered as a cluster. In the last decades, several nature-inspired optimization algorithms proved to be efficient for several computing problems. Firefly algorithm is one of the nature-inspired metaheuristic optimization algorithms regarded as an optimization tool for many optimization issues in many different areas such as clustering. To overcome the issues of velocity, the firefly algorithm can be integrated with the popular particle swarm optimization algorithm. In this paper, two modified firefly algorithms, namely the crazy firefly algorithm and variable step size firefly algorithm, are hybridized individually with a standard particle swarm optimization algorithm and applied in the domain of clustering. The results obtained by the two planned hybrid algorithms have been compared with the existing hybridized firefly particle swarm optimization algorithm utilizing ten UCI Machine Learning Repository datasets and eight Shape sets for performance evaluation. In addition to this, two clustering validity measures, Compact-separated and David–Bouldin, have been used for analyzing the efficiency of these algorithms. The experimental results show that the two proposed hybrid algorithms outperform the existing hybrid firefly particle swarm optimization algorithm.

Keywords: hybrid firefly particle swarm optimization algorithm; crazy firefly algorithm; variable step size firefly algorithm; compact-separated validity index; David–Bouldin validity index

MSC: 68T10

check for updates

Citation: Behera, M.; Sarangi, A.; Mishra, D.; Mallick, P.K.; Shafi, J.; Srinivasu, P.N.; Ijaz, M.F. Automatic Data Clustering by Hybrid Enhanced Firefly and Particle Swarm Optimization Algorithms. *Mathematics* **2022**, *10*, 3532. https://doi.org/10.3390/math10193532

Academic Editor: Ioannis E. Livieris

Received: 26 July 2022
Accepted: 20 September 2022
Published: 28 September 2022

1. Introduction

Clustering is a widely used unsupervised machine learning tool used to group data based on their similarity and dissimilarity properties [1]. The use of clustering technology is utilized for a wide range of application scenarios such as data mining, marketing, medicine, banking and finance, data science, machine learning, agriculture etc. In artificial intelligence and data mining, clustering data into meaningful clusters is a significant challenge. There are various aspects to influence the outcomes of clustering algorithms, such as the number of clusters that can be generated in a set of data, the standard and approach to clustering [2]. A variety of clustering techniques such as simulated annealing, k-means, k-medoids and fuzzy c-mean have been suggested to resolve data clustering problems. Such strategies are completely reliant on the initial solution; as a result, the chance of becoming easily stuck inside the local optima is high. Since these clustering algorithms are unable to handle the clustering job for large and complex datasets, several nature-inspired meta-heuristic optimization algorithms have been proposed to overcome the clustering problems by experimenting on several complex and high-dimensional datasets [3]. In the past few decades, to solve the data clustering problems, several evolutionary algorithms such as differential evolution algorithm and genetic algorithm along with several swarm intelligence algorithms such as particle swarm optimization, ant colony optimization, artificial bee colony optimization algorithms, firefly optimization have been applied.

In recent years, owing to its simplicity, effectiveness, robust and better performance, the firefly algorithm (FA) has gained more attention from many global optimization researchers. From the literature review, it is found that the limitations of FA can be overcome by hybridizing FA with different metaheuristic optimization algorithms. In addition to this, a strong balance between exploitation and exploration will be maintained [4]. The main purpose of this study is to build up an automatic superior data-clustering technique without prior knowledge regarding the characteristics of datasets. In this study two new modified FAs have been hybridized with particle swarm optimization algorithm (PSO). The results obtained by these two modified hybrid algorithms will be analyzed along with the existing hybrid firefly particle swarm optimization algorithm (HFAPSO). The two modified hybrid FAs include a hybrid crazy firefly algorithm particle swarm optimization algorithm (HCFAPSO) and a hybrid variable step size firefly algorithm (HVSFAPSO). In crazy FA, the searching diversity is well maintained by adding a craziness factor to the standard FA. So, it will perform better than the standard firefly algorithm [5]. In variable step size FA, a variable step size strategy is added to the standard FA to decrease the searching step size with the number of iterations; as a result, the detection, development and accuracy of the optimization can be improved. Furthermore, two cluster analysis validity measures, namely Davis–Bouldin (DB) and Compact-Separated (CS), have been used to check the validity of clustering solutions [6,7]. The experimental results are carried out based on ten UCI repository datasets and eight Shape sets to verify the better performance of proposed hybrid algorithms over other existing clustering algorithms. The traditional clustering algorithms are unable to handle the clustering task of high dimensional datasets; to overcome this problem many algorithms have been developed [8]. Sustainable automatic data clustering using a hybrid PSO algorithm with mutation was proposed by Sharma and Chhabra [9]. A novel sustainable hybrid clustering algorithm (HPSOM) has thus been developed by integrating PSO with a mutation operator for different network-generated datasets. In automatic data clustering, using the nature-inspired symbiotic organism search (SOS) algorithm, Zhou et al. [10] describe the need of SOS algorithms to overcome the clustering issues. The experimental results prove that the SOS algorithm outperforms other optimization algorithms with high accuracy and stability. Another clustering algorithm proposed by Rajah and Ezugwu [11] is a hybrid algorithm for automatic data clustering in which five novel hybrid symbiotic searching algorithms are used for automatic data clustering without having any primary information regarding the number of clusters. Researchers have identified the NP-difficult, automatic clustering issues; to overcome these issues a metaheuristic ant optimization algorithm has been proposed [12]. All the above-mentioned works demonstrate the need for studies on automatic data clustering problems and to perform certain improvements in data clustering tasks by developing two hybrid models with the help of swarm intelligence algorithms.

Most clustering approaches would require a clearly specified objective function. A Euclidean-based distance measure is chosen along with the two aforementioned validity indices utilized for computing the fitness function of each solution obtained. The majority of metaheuristic algorithms can manage noise or outlier identification related to the datasets and automatically split datasets into an ideal number of clusters. These algorithms start with a population of randomly generated individuals and try to optimize the population over a sequence of generations until the optimum solution is obtained. The proposed algorithms focus initially on searching for the best number of clusters and, after that, gradually move to globally optimal cluster centers. Two types of continuous fitness functions are designed on the basis of current clustering validation indices and the penalty functions designed for minimizing the amount of noise and to control the number of clusters [1]. The CS and DB validity index is used as fitness function to calculate the fitness of each firefly. The cluster center will be refined by the position firefly. The distance between two fireflies will be calculated using the Euclidean distance, where the position of firefly will be updated using the fitness parameter [3,4].

- Two variants of FA, namely crazy FA and variable step size FA, are hybridized individually with the standard PSO algorithm;
- The proposed hybrid algorithms are applied in an automatic data clustering task. The results obtained by the two planned hybrid algorithms have been compared with the existing HFAPSO;
- Ten UCI Machine Learning Repository datasets and eight Shape sets have been used for performance evaluation;
- Two clustering validity measures, CS and DB, have been used for analyzing the efficiency of these algorithms;
- The numerical result calculation is based on the CS as well as the DB validity index;
- The mean value and standard deviation for both the CS and DB validity index has been given;
- The main aim is to obtain the optimal clustering results.

The outline of this study is as follows: Section 2 gives an overview of related work. Section 3 describes the proposed methodology. Section 4 represents detailed knowledge regarding the results analysis. Section 5 presents a comparison study of HFASO, HCFAPSO and HVSFAPSO automatic clustering algorithms. Section 6 describes the findings of this work. Finally, Section 7 concludes the study with a concluding note and future research ideas.

2. Related Work

2.1. The Clustering Problem Description

In this research article, two different modified HFAPSO algorithms have been proposed to overcome the automatic data clustering issues. Automatic data clustering is adopted according to the method mentioned [13]. Let the data set given be $S = \{s_1, s_2, s_3, \ldots s_n\}$ be divided into clusters $M = \{m_1, m_2, m_3, \ldots m_n\}$ which are non-overlapping in nature, such that the dimension $X_i (i = 1, 2, 3, \ldots, n)$ is q. A cluster center (centroid) $c_i = (i = 1 \ldots\ldots D)$ is allocated for every cluster, i.e., $Y = (y_1, y_2, \ldots\ldots y_D)$ belongs to the centres of $M = \{m_1, m_2, \ldots\ m_d\}$. For an l-dimensional data vector, the following mentioned criteria should be considered [3]:

$$M_i \cap M_j = \phi \ where\ i, j = 1, 2, \ldots, D\ and\ i \neq j \tag{1}$$

$$M_1 \cup M_2 \cup \ldots \cup M_D = S \tag{2}$$

$$M_i \subseteq S\ and\ M_i \neq 0,\ i = 1, 2 \ldots, D \tag{3}$$

In the initialization step of every hybrid clustering algorithm, the swarm size R is defined as X $= (x_1, x_2, \ldots x_R)$. Assume every member X_i in the swarm size will be a D × l-dimensional vector and the $S_{n \times 1}$, which is described as $X_i = X_1^*, X_2^*, \ldots, X_D^* = (X_{11}, X_{12}, \ldots X_{1q}), (X_{21}, X_{22}, \ldots, X_{2q}), \ldots\ldots, (X_{D1}, X_{D2}, \ldots X_{Dq})$. The objective of the optimization process is performed by the proposed hybrid algorithms by using two performance measures: CS index and DB index for data clustering automatically by reducing the sum of distance between datasets $S_i (i = 1, 2, I, n)$ and centres $c_i (i = 1I \ldots D)$. The upper bound as well as the lower bound of the total number of groups in the Iwarm is represented like Var_{max} and Var_{min}, respectively. Here the Var_{max} is represented as $P_i^* = max\{S_1, S_2, \ldots\ S_q\}$ and the Var_{min} is represented as $R_i^* = min\{S_1, S_2, \ldots, S_q\}$. Generally, for the solution space, the lower boundary is $R = (R_1^*, R_2^*, \ldots, R_D^*)$ and the upper boundary is $P = \left(P_1^*, P_2^*, \ldots, P_D^*\right)$. To overcome the automatic data clustering issues, the ith particle X_i is calculated as Equation (4) [3]

$$X_i = random(1, D \times l) \times (P - R) + R \tag{4}$$

Here, $random\ (1,\ D \times l)$ is a vector of randomized number which is distributed in a uniform manner returns a value ranging between 0 and 1.

2.2. The Clustering Validity Measures

In this study, we consider two different validity measures, CS index and DB index, to calculate the efficiency as well as the cluster quality and analyse the performance of the clustering algorithms.

2.2.1. Compact-Separated Index (CS Index)

The CS index calculates the ratio between the total sum of within-cluster scattering and the between-cluster separation. It has been observed that the CS index performs better while working for automatic data clustering with different sizes or densities and dimensions. Therefore, the CS validity measure can be evaluated like a fitness function, as the following mentioned in Equation (5) (Chou et al. [7]).

$$f_{CS} = \frac{\sum_{i=1}^{D}\left[\frac{1}{|S_n|}\sum_{X_i \in Y_i} \max_{Z_j \in Y_i}\left\{V\left(X_i, X_j\right)\right\}\right]}{\sum_{i=1}^{D}\left[\min_{j \in D, j \neq i}\left\{V\left(s_i, s_j\right)\right\}\right]} \tag{5}$$

where $|S_n|$ denotes to the total number of data points in the cluster M, the function $V\left(X_i, X_j\right)$ represents to the distance between the intra-cluster scatter and the inter-cluster separation X_j and $V\left(s_i, s_j\right)$ represents to the distance between the datapoints s to their centroid.

2.2.2. Davis–Bouldin Index (DB Index)

The DB index calculates automatic data cluster results with the help of estimating intra-cluster (mean distance of all the data points in a cluster from the centroid) to inter-cluster (distance in between two different centroids) distance. Like the CS index, lower values of the DB index will result in good compactness or separation, while the opposite is true for a high value. The DB index can be evaluated like a fitness function, as per the following Equation (6) [6]

$$\begin{gathered} f_{DB} = \frac{1}{D}\sum_D^1 V_i \\ \text{Here } V_i = max\left\{\frac{G_i + G_j}{H_{ij}} \middle|\, 1\ \leq\ i, j \leq D,\ i \neq j\right\} \end{gathered} \tag{6}$$

V_i is the distance between the data point to the centroid, whereas G_i and G_j are represented as the mean for all data points intra clusters of their distance in between data points and their centroids, and also the H_{ij} denoted as the inter-cluster distance in between two centroids.

2.3. Firefly Algorithm (FA)

The FA is a population-based, stochastic, meta-heuristic, nature-inspired, swarm-related algorithm proposed by Yang [13]. This refers to stochastic algorithms that attempt to search for a collection of solutions using a randomization method. The intensity (I) of firefly is inversely proportional to the distance between two fireflies (r) [13]

$$I = \frac{1}{r^2} \tag{7}$$

Every firefly has its own unique and special attractiveness which determines how intensely a firefly excites other swarm members. However, the attraction β is still relative;

this will differ along with distance r_{ij}. In between two different fireflies, i and j presented at locations x_i and y_i, respectively [13]

$$r_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{8}$$

Firefly positions are the locations where the fireflies are present. Here, x_i and y_i are the positions where two different fireflies i and j presented.

The attractiveness function $\beta(r)$ of the firefly is evaluated as follows.

$$\beta(r) = \beta_0 e^{-\gamma r^2}, \text{ Here, } \beta_0 \text{ represents the attractiveness at } r = 0. \tag{9}$$

The firefly that has the finest fitness value will be chosen to perform in the succeeding optimizing process. In the case of equal brightness, the firefly arbitrarily migrates. By applying the current position, the firefly adjusts its position. The movement of the firefly i attracts another brighter firefly j and will be evaluated (Yang [14]).

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}\left(x_i - x_j\right) + \alpha\left(rand - \frac{1}{2}\right) \tag{10}$$

Algorithm 1 is data clustering using the FA algorithm. In this algorithm, the method by which clusters are encoded with FA algorithm is mentioned. Initially, every firefly and cluster centroid is initialized randomly. Then the fitness function is calculated as Equations (5) and (6). According to Equation (9), firefly attraction will be changed and the fireflies' positions will be updated based on ranking; in addition to this, the current best solution will be updated.

Algorithm 1. Pseudo code for FA

```
Start
    Initialized every firefly $F_{firefly}$ and $R$ random cluster centroid randomly;
for $i = 1$ $to$ $m$;
Calculate the fitness function $f_{CS}$ $and$ $f_{DB}$ as Equations (5) and (6) respectively and obtain the current best solution;
As Equation (7) Intensity of light will be calculated;
Define light absorption coefficient;
while $Iteration < Maximum\ Iteration$
    for $i = 1$ $to$ $m$;
        for $j = 1$ $to$ $m$;
            if $F_{firefly}(j) < F_{firefly}(i).cost$
                Move $F_{firefly}(i)$ towards $F_{firefly}(j)$ as Equation (10) to refine cluster centers;
    end if
    According to Equation (9), attraction of fireflies changes with distance;
    New solution will be calculated and light intensity will be updated;
    end for
end for
    Fireflies position will be updated based on ranking and the current best solution will be updated.
End for
end while
End
```

2.4. Particle Swarm Optimization Algorithm (PSO)

The PSO algorithm is a population oriented, stochastic, metaheuristic optimization algorithm based on the social behaviour of swarm or a group of individuals [15]. In this algorithm, the position as well as the velocity of each particle of swarm will be updated by using the objective function to achieve the best simulation results. The velocity as well as the position of every individual will be adjusted as Equations (11) and (12), respectively [16].

$$v_{ik}^{(t)} = w \times v_{ik}^{(t-1)} + c_{pa} \times random\left(p_{ik} - x_{ik}^{(t-1)}\right) + c_{ga} \times Random\left(p_{gk} - x_{ik}^{(t-1)}\right) \quad (11)$$

$$x_{ik}^{(t)} = x_{ik}^{(t-1)} + v_{ik}^{(t)} \quad (12)$$

Here, $v_{ik}^{(t)}$ is considered as the latest velocity of the particle, w is taken as the inertia weight, $v_{ik}^{(t-1)}$ is the present velocity of individual, c_{pa} and c_{ga} are personal and global acceleration coefficients respectively, *random* and *Random* are two uniformly distributed independent random variables in the range between $[0,1]$, p_{ik} is the earliest best position of ith particle,$x_{ik}^{(t-1)}$ is the position of particle i in the $t-1$ iteration, p_{gk} is represented as global best position of the population. $x_{ik}^{(t)}$ is represented as the latest position of a particle.

2.5. Hybrid Firefly Particle Swarm Optimization Algorithm (HFAPSO)

The hybrid clustering algorithm has been designed by integrating the advantages of both PSO and FA algorithms. While creating a hybrid clustering model, two major problems may arise. The first problem is created by adding two or more separate techniques into a single design and the second is the calculation of the best solution by using the process of individual solution searching. Here in this hybrid algorithm, the designing process is carried out by combining FA and PSO. The FA algorithm has strong intensification ability, whereas the PSO algorithm has strong exploration ability [17]. The FA algorithm is good for local searching whereas the PSO algorithm is good for global search solutions. So, for hybridization purposes, the FA algorithm will be taken as the base searching algorithm and then the PSO algorithm will be integrated for finding the optimal solution. Both FA and PSO have their own advantages. By combining FA and PSO, an excellent hybrid optimization algorithm which can be used for automatic data clustering can be developed [17].

The clustering results and efficiency of the HFAPSO clustering algorithm can be determined by using the CS validity measure and DB validity measure. In addition, these performance measures will also help to select the best perfect number of clusters and are also needed for finding the finest partitioning for the selected clusters. While carrying out a further global search for the optimum solution, Firefly does not require any prior previous information of the local best position. Additionally, Firefly does not suffer from the problems with velocity startup and instability for high-velocity particles. The working of HFAPSO will begin with a randomly initializing process by defining the initial firefly population. Then the fitness value of every solution of the FA will be calculated by applying the CS measure and the DB measure. Thereafter the population will be modified by the help of FA operators. Consequently, a similar approach will be repeated in an iterative manner for PSO operators still in the first cycle of the calculation stage of HFAPSO design. PSO makes use of the finest solution provided by FA as its initial search population. The position as well as the velocity of the newly generated solutions by PSO will be updated. In estimation, the previous local best value as well as the previous global best value will be compared with the new population, and the particle will also be updated with the finest fitness values as the global best or best solution. Likewise, the CS and the DB measures have been used by PSO for measuring every solution's ultimate fitness function. Then, it is used by HFAPSO to define the finest candidate solution and performs the required modifications. Ultimately, the finest solution will be evaluated, depending on the solution having the smallest CS index value or DB index value. The two stages of the HFAPSO algorithm will be repeated until the termination conditions are satisfied.

3. Proposed Methodology

3.1. Hybrid Crazy Firefly Particle Swarm Optimization Algorithm (HCFAPSO)

In swarm intelligence algorithms, birds, fish or fireflies can change their direction quickly. A term of craziness has been used in many swarm intelligence techniques to define the unexpected change of direction in optimization algorithm. The global searching ability of the traditional HFAPSO can be increased by introducing a craziness operator to the traditional FA, and this modification is considered as the best modification to obtain that each firefly must have a predefined probability of craziness for better diversification maintenance. The crazy FA will give superior results with higher convergence rate than the traditional FA. In addition to this this craziness operator will help to obtain improved exploration ability. Mathematically, the expression using craziness is as follows.

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_i - x_j) + \alpha\left(rand - \frac{1}{2}\right) + p(r)\mathrm{sgn}(r)v_d^{craziness} \tag{13}$$

Here $v_d^{craziness}$ and r are arbitrary parameters selected uniformly inside the interval of $\left[v_d^{min}, v_d^{max}\right]$ and [0, 1] respectively.

The $sgn(r)$ and $p(r)$ functions are defined as $sgn(r) = 1$ where $r \geq 0.5$
-1 where $r < 0.5$
$p(r) = 1$ where $r \leq p_{cr}$
$= 0$ where $r > p_{cr}$, respectively.

p_{cr} is represented as a predefined craziness probability [18]. Adding a crazy operator with the FA algorithm will improve the performance and the searching ability of the algorithm. For data clustering tasks, the crazy operator will give the best results. Therefore, in this hybrid automatic clustering algorithm, the crazy FA will be integrated with the PSO algorithm for finding better cluster results.

In Algorithm 2 the proposed HCFAPSO method for automatic data clustering is described properly. In this algorithm the stepwise procedure of the automatic data clustering using HCFAPSO is stated, in which the crazy FA algorithm is integrated with PSO algorithm. Whereas Figure 1 is the flowchart of the proposed HCFAPSO algorithm for automatic data clustering. In Figure 1 the working principle of the proposed HCFAPSO algorithm is described.

3.2. Hybrid Variable Step Size Firefly Particle Swarm Optimization Algorithm (HVSFAPSO)

The performance of standard FA can be improved by increasing the convergence speed [19]. To overcome the drawbacks of standard FA, the global exploration as well as the local exploitation should be maintained properly. For this purpose, the step size α should be adjusted dynamically. In standard FA, the step size α is constant and will not perfectly follow the searching process. In variable step size algorithms, the step size α is considered as a variable. To maintain balance in between the identification and development capacity of firefly algorithms, initially step size α should be a larger value. Subsequently, it decreases over iterations. Based on various searching space optimization issues, a large searching step size is needed if the definition space of the optimization target is large. Otherwise, a small searching step size is required, which will aid the algorithm's ability to adapt to a variety of optimization issues [20].

$$\alpha(t) = \frac{0.4}{\left(1 + \exp\left(0.015 \times \frac{(t - \mathrm{max}generation)}{3}\right)\right)} \tag{14}$$

Algorithm 2. Pseudo code for HCFAPSO

```
    Start
       Initialize every firefly $F_{firefly}$ and $R$ random cluster centroid randomly;
       Evaluate fitness value;
for $i = 1$ to $m$;
Calculate the fitness function $f_{CS}$ and $f_{DB}$ as Equations (5) and (6) respectively and obtain the current best solution;
if the current value of $F_{firefly}(i).Cost \leq Bestsolution.cost$, then modify the current $F_{firefly}(i)$ as best solution;
$Bestsolution = F_{firefly}(i)$;
end if
end for
while $Iteration < Maximum\ Iteration$
    for $i = 1$ to $m$;
    for $j = 1$ to $m$;
    if $F_{firefly}(j) < F_{firefly}(i).cost$
              Move $F_{firefly}(i)$ towards $F_{firefly}(j)$ as Equation (13);
              if $newsolution.cost \leq newF_{firefly}(i).cost$;
               $newF_{firefly}(i)$ will be the new solution;
                    if $newF_{firefly}(i).cost \leq Bestsolution.cost$;
                    Modify $new\ F_{firefly}(i)$ as new solution;
                    end if
              end if
end if
    Initialized $F_{PSO}(i) \leftarrow newF_{firefly}(i)$ randomly;
Calculate the $P_{best}$ and $G_{best}$ position of every particle $F_{PSO}(i)$;
    Evaluate $F_{PSO}(i)$ fitness value by taking the function $f_{CS}$ and $f_{DB}$;
    if $F_{PSO}(i).cost \leq newF_{firefly}(i).cost$
        $F_{P_{best}} \leftarrow F_{PSO}(i)$;
else if $newF_{PSO}(i)$ the fitness value is lesser than the overall best fitness value, then modify the new value as the global best value. $F_{G_{best}} \leftarrow newF_{PSO}(i)$;
Modify centroids of cluster following velocity and coordinates modifying Equations (11) and (12);
end if
    end if
end for
end for
end while
End
```

Here t = number of existing iterations and max generation = maximum number of iterations. For data clustering tasks, the variable step size FA will work better than the standard FA algorithm by adjusting the step size dynamically. Therefore, in this hybrid automatic clustering algorithm, the variable-step-size firefly algorithm will be integrated with the particle swarm optimization algorithm to obtain better cluster results.

Algorithm 3 is the proposed HVSFAPSO method for automatic data clustering. In Algorithm 3, the variable-step-size FA algorithm is hybridized with PSO for automatic data clustering; Figure 2 shows the flowchart for the proposed HVSFAPSO algorithm. The working principle of the automatic data clustering task is described in detail in Figure 2.

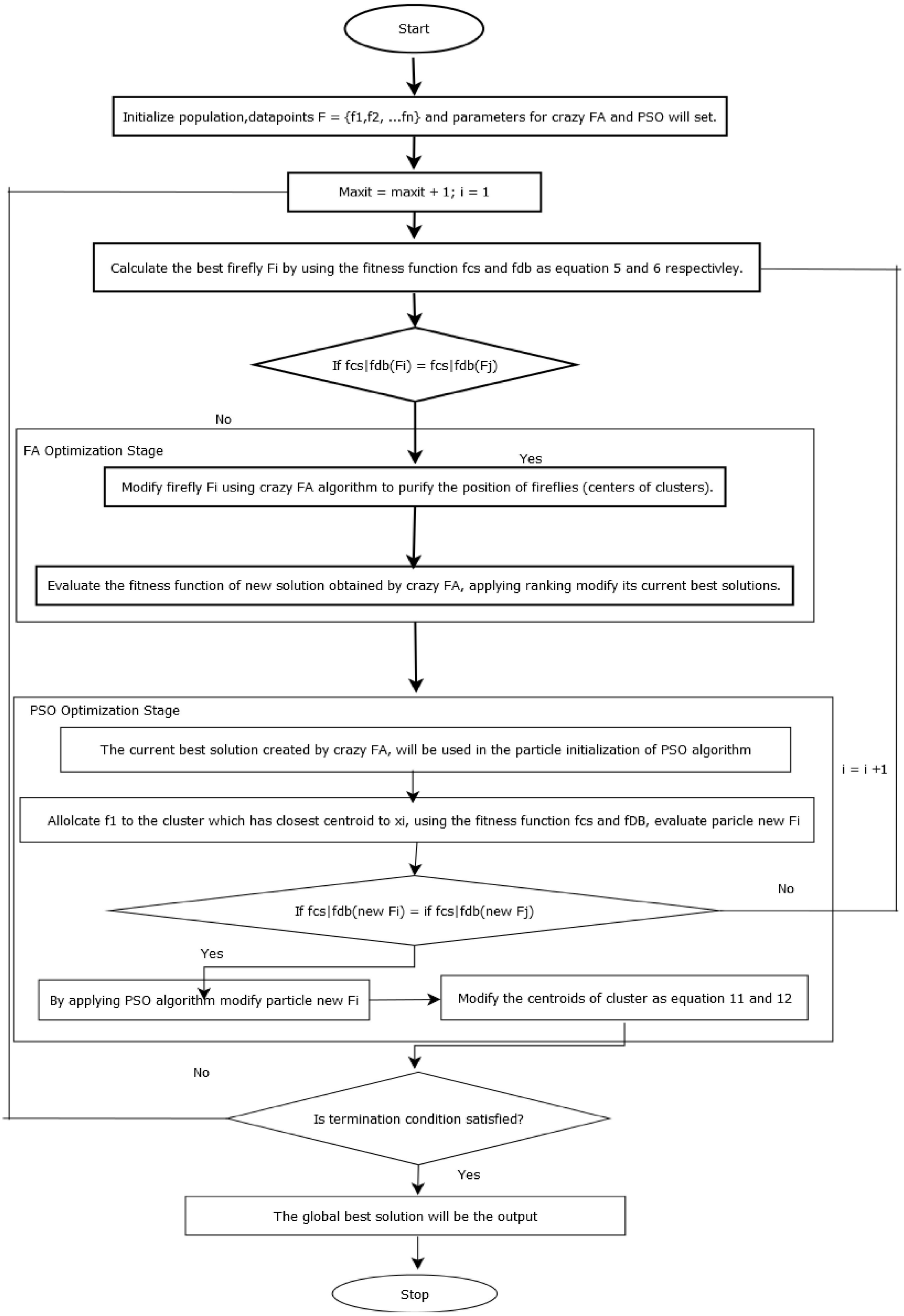

Start
Initialize population,datapoints F = {f1,f2, ...fn} and parameters for crazy FA and PSO will set.
Maxit = maxit + 1; i = 1
Calculate the best firefly Fi by using the fitness function fcs and fdb as equation 5 and 6 respectivley.
If fcs|fdb(Fi) = fcs|fdb(Fj)
No
FA Optimization Stage
Yes
Modify firefly Fi using crazy FA algorithm to purify the position of fireflies (centers of clusters).
Evaluate the fitness function of new solution obtained by crazy FA, applying ranking modify its current best solutions.
PSO Optimization Stage
The current best solution created by crazy FA, will be used in the particle initialization of PSO algorithm
i = i +1
Allolcate f1 to the cluster which has closest centroid to xi, using the fitness function fcs and fDB, evaluate paricle new Fi
No
If fcs|fdb(new Fi) = if fcs|fdb(new Fj)
Yes
By applying PSO algorithm modify particle new Fi
Modify the centroids of cluster as equation 11 and 12
No
Is termination condition satisfied?
Yes
The global best solution will be the output
Stop

Figure 1. Flowchart for HCFAPSO automatic clustering.

Algorithm 3. Pseudo code for HVSFAPSO

```
    Start
  Initialized every firefly $F_{firefly}$ and $R$ random cluster centroid randomly;
  Evaluate fitness value;
for $i = 1$ to $m$;
  Calculate the fitness function $f_{CS}$ and $f_{DB}$ as Equations (5) and (6) respectively and obtain the current best solution;
if the current value of $F_{firefly}(i).Cost \leq Bestsolution.cost$, then modify the Current $F_{firefly}(i)$ as best solution;
$Bestsolution = F_{firefly}(i)$;
end if
end for
while $Iteration < Maximum\ Iteration$
  for $i = 1$ to $m$;
  for $j = 1$ to $m$;
if $F_{firefly}(j) < F_{firefly}(i).cost$
        Move $F_{firefly}(i)$ towards $F_{firefly}(j)$ as Equations (8) and (9);
          Set the value of step size $\alpha$ as Equation (14)
          if $newsolution.cost \leq newF_{firefly}(i).cost$;
         $newF_{firefly}(i)$ will be the new solution;
          if $newF_{firefly}(i).cost \leq Bestsolution.cost$;
          Modify $new\ F_{firefly}(i)$ as new solution;
          end if
            end if
  end if
Initialized $F_{PSO}(i) \leftarrow newF_{firefly}(i)$ randomly;
Calculate the $P_{best}$ and $G_{best}$ position of every particle $F_{PSO}(i)$;
  Evaluate $F_{PSO}(i)$ fitness value by taking the function $f_{CS}$ and $f_{DB}$;
  if $F_{PSO}(i).cost \leq newF_{firefly}(i).cost$
      $F_{P_{best}} \leftarrow F_{PSO}(i)$;
        else if $newF_{PSO}(i)$ the fitness value is lesser than the overall best fitness value, then modify the new value as the global best value. $F_{G_{best}} \leftarrow newF_{PSO}(i)$;
  Modify centroids of cluster following velocity and coordinates modifying Equations (11)and (12);
end if
  end if
end for
end for
end while
End
```

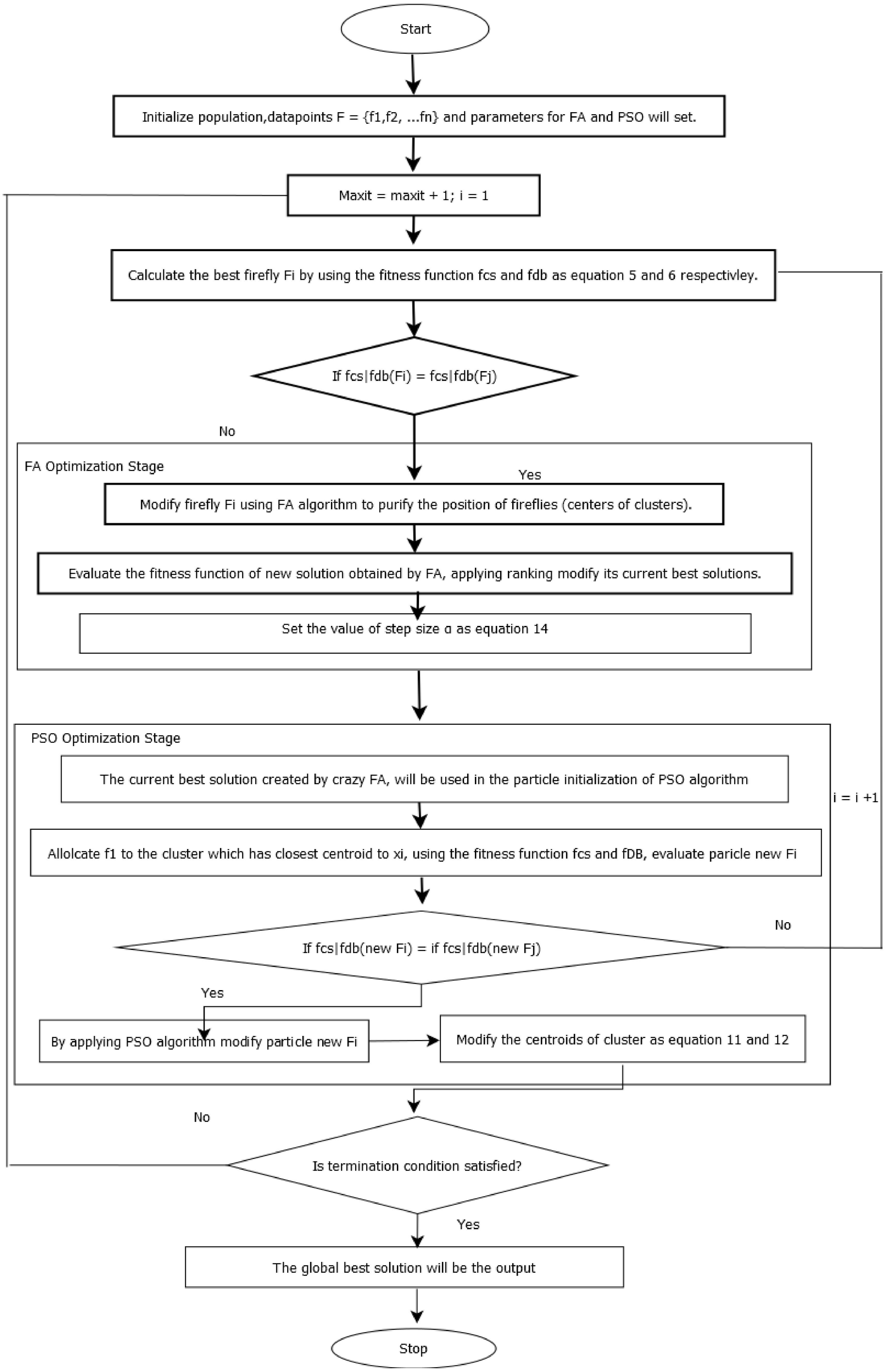

Start
Initialize population,datapoints F = {f1,f2, ...fn} and parameters for FA and PSO will set.
Maxit = maxit + 1; i = 1
Calculate the best firefly Fi by using the fitness function fcs and fdb as equation 5 and 6 respectivley.
If fcs|fdb(Fi) = fcs|fdb(Fj)
No
FA Optimization Stage
Yes
Modify firefly Fi using FA algorithm to purify the position of fireflies (centers of clusters).
Evaluate the fitness function of new solution obtained by FA, applying ranking modify its current best solutions.
Set the value of step size α as equation 14
PSO Optimization Stage
The current best solution created by crazy FA, will be used in the particle initialization of PSO algorithm
i = i +1
Allolcate f1 to the cluster which has closest centroid to xi, using the fitness function fcs and fDB, evaluate paricle new Fi
No
If fcs|fdb(new Fi) = if fcs|fdb(new Fj)
Yes
By applying PSO algorithm modify particle new Fi
Modify the centroids of cluster as equation 11 and 12
No
Is termination condition satisfied?
Yes
The global best solution will be the output
Stop

Figure 2. Flow chart for HVSFAPSO automatic clustering.

4. Results Analysis

This section presents detailed information regarding the simulation experiments on automatic data clustering by using three hybrid algorithms and describes in detail the system configuration along with the design of the datasets; in addition to this, the results obtained will be discussed.

4.1. System Configuration

The experiments have been implemented on MATLAB by using a 1.80 GHz Intel(R) Core (TM) i5-8265U processor with 4.00 GB RAM on a Windows 10 operating system. The parameter setting for the HFAPSO implementation is given in Table 1. The parameter values of the firefly algorithm have already been explored in past studies [3,4,14,17]. From those works, the optimal range values have been chosen, rather than a specific fixed value to obtain better clustering results. The experiments have been conducted on HFAPSO by using 200 iterations with 25 population size for 20 independent runs. Furthermore, 10 UCI repository datasets and 8 Shape sets have been used for experimental purposes. The details regarding the datasets are mentioned in Table 2.

Table 1. Parameter setting.

HFAPSO		HCFAPSO		HVSFAPSO	
Value	**Parameters**	**Value**	**Parameters**	**Value**	**Parameters**
nPop	25	nPop	25	nPop	25
gamma	1	gamma	1	gamma	1
beta0	2.0	beta0	2.0	beta0	2.0
alpha	0.2	alpha	0.2	alpha	0.2
alpha_damp	0.98	alpha_damp	0.98	alpha_damp	0.98
Maxit	200	Maxit	200	Maxit	200
W	1	W	1	W	1
Wdamp	0.99	Wdamp	0.99	Wdamp	0.99
C1	1.5	C1	1.5	C1	1.5
C2	2.0	C2	2.0	C2	2.0
		Pr	1		
		Sgnr	−1		
		Vdcraziness	0.95		

4.2. Datasets Design

Eighteen datasets used for experimental study were taken from the UCI Machine Learning Repository and Shape sets [21,22]. The detailed descriptions of the datasets have been mentioned in Table 2, including the number of data points, datasets' dimensions and number of clusters. The experiments for the task of automatic data clustering have been performed by considering 200 iterations with 25 populations on 20 independent runs. The numerical result calculation is based on the CS validity index as well as the DB validity index, which are presented in Table 3. In the results table, the clustering results are given. The Best, Worst, Average, StDev represent the best clustering solution, worst clustering solution, average clustering solution and the standard deviation respectively [23].

Table 2. Characteristics of the eighteen datasets.

Sl. No	Datasets Used	Type of Dataset Used	Total Number of Data Points(N)	Dimensions of Datasets (D)	Existing Number of Clusters
1	Iris	UCI dataset	150	4	3
2	Yeast	UCI dataset	1484	8	10
3	Wine	UCI dataset	178	13	3
4	Thyroid	UCI dataset	215	5	2
5	Spiral	Shape set	312	2	3
6	Path based	Shape set	300	2	3
7	Jain	Shape set	373	2	2
8	Hepatitis	UCI dataset	155	19	2
9	Heart	UCI dataset	270	13	2
10	Glass	UCI dataset	214	9	7
11	Flame	Shape set	240	2	2
12	Compound	Shape set	399	2	6
13	Breast	UCI dataset	699	9	2
14	Wdbc	UCI dataset	569	32	2
15	R15	Shape set	600	2	15
16	Leaves	UCI dataset	1600	64	100
17	D31	Shape set	3100	2	31
18	Aggregation	Shape set	788	2	7

4.3. Results Discussion

In this section, the numerical results obtained by automatic data clustering for all three above mentioned hybrid algorithms by taking two validity indices: the CS index and the DB index, over 20 independent runs, have been discussed clearly. In Table 3 the results of the HFAPSO automatic data clustering algorithm for all datasets is clearly presented. The four decimal values are in bold, representing that they constitute the best value for this set. The main aim is to get quality results with less execution time to execute every algorithm to obtain the optimal clustering results. In Table 3, it is shown that CS index outperforms the DB index in some datasets: Spiral, Path based, Jain, Flame, Breast, R15, Leaves, and D31. Similarly, the DB index outperforms than CS index in some datasets, namely Iris, Yeast, Wine, Thyroid, Hepatitis, Heart, Glass, Compound, Wdbc, and Aggregation. In Figure 3, the execution times of HFAPSO for both the CS index and DB index are presented clearly. In the Figure 3 the blue bar represents the CS index whereas the orange bar represents the DB index. However, the figure shows that the CS index will take longer to execute in comparison to the DB index to obtain the clustering results. The clustering illustrations of each individual datasets of HFAPSO, HCFAPSO and HVSFAPSO automatic data clustering based on both CS index and DB index is presented in Figures 4 and 5 respectively.

Table 3. Numerical results of HFAPSO based on CS and DB validity indices over 20 independent runs.

Dataset Used	CS-Index				DB-Index			
	Best	**Worst**	**Average**	**StaDev.**	**Best**	**Worst**	**Average**	**StDev.**
Iris	0.7191	0.8488	0.7438	0.0382	0.57	0.9155	0.6739	0.1168
Yeast	0.5204	0.7421	0.5395	0.0578	0.4382	1.1832	0.8143	0.2385
Wine	0.8828	1.2226	0.9433	0.0891	0.8002	1.222	0.9835	0.1782
Thyroid	0.6408	0.6408	0.6408	0.0000	0.4813	1.0118	0.6765	0.2003
Spiral	0.5541	0.979	0.7766	0.1149	0.7273	0.8173	0.7633	0.0340
Path based	0.4441	0.9713	0.6976	0.1757	0.6249	0.8409	0.6975	0.0475
Jain	0.4848	0.8126	0.6523	0.0729	0.6478	0.7226	0.6268	0.0359
Hepatitis	0.5298	0.5298	0.5298	0.0000	0.4318	0.5236	0.4529	0.0263
Heart	0.5974	0.5974	0.5974	0.0000	0.4515	0.6333	0.5150	0.0617
Glass	0.0607	0.0607	0.0607	0.0000	0.334	0.9849	0.7447	0.7379
Flame	0.3846	1.0101	0.5195	0.1860	0.63	0.8024	0.7359	0.0501
Compound	0.5032	0.7732	0.7019	0.0989	0.4931	0.5878	0.5170	0.0281
Breast	0.5996	1.1514	0.8844	0.2382	0.6519	1.4911	0.9730	0.3284
Wdbc	0.0712	0.0712	0.0712	0.0000	0.0507	0.5459	0.0801	0.1035
R15	0.6876	0.9129	0.7235	0.0685	0.714	0.8957	0.7860	0.0565
Leaves	0.4919	0.6994	0.5124	0.0530	0.5833	1.5194	1.0337	0.4390
D31	0.7127	1.1947	0.8822	0.1481	0.7929	0.9043	0.8350	0.0394
Aggregation	0.7352	1.0031	0.8272	0.1027	0.7199	0.7751	0.7354	0.0185

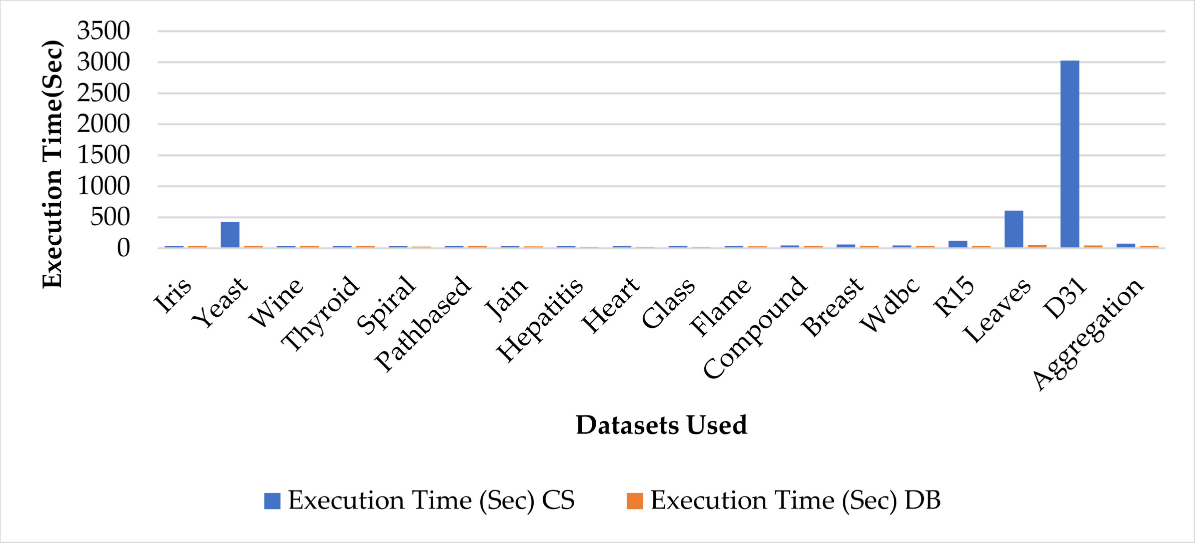
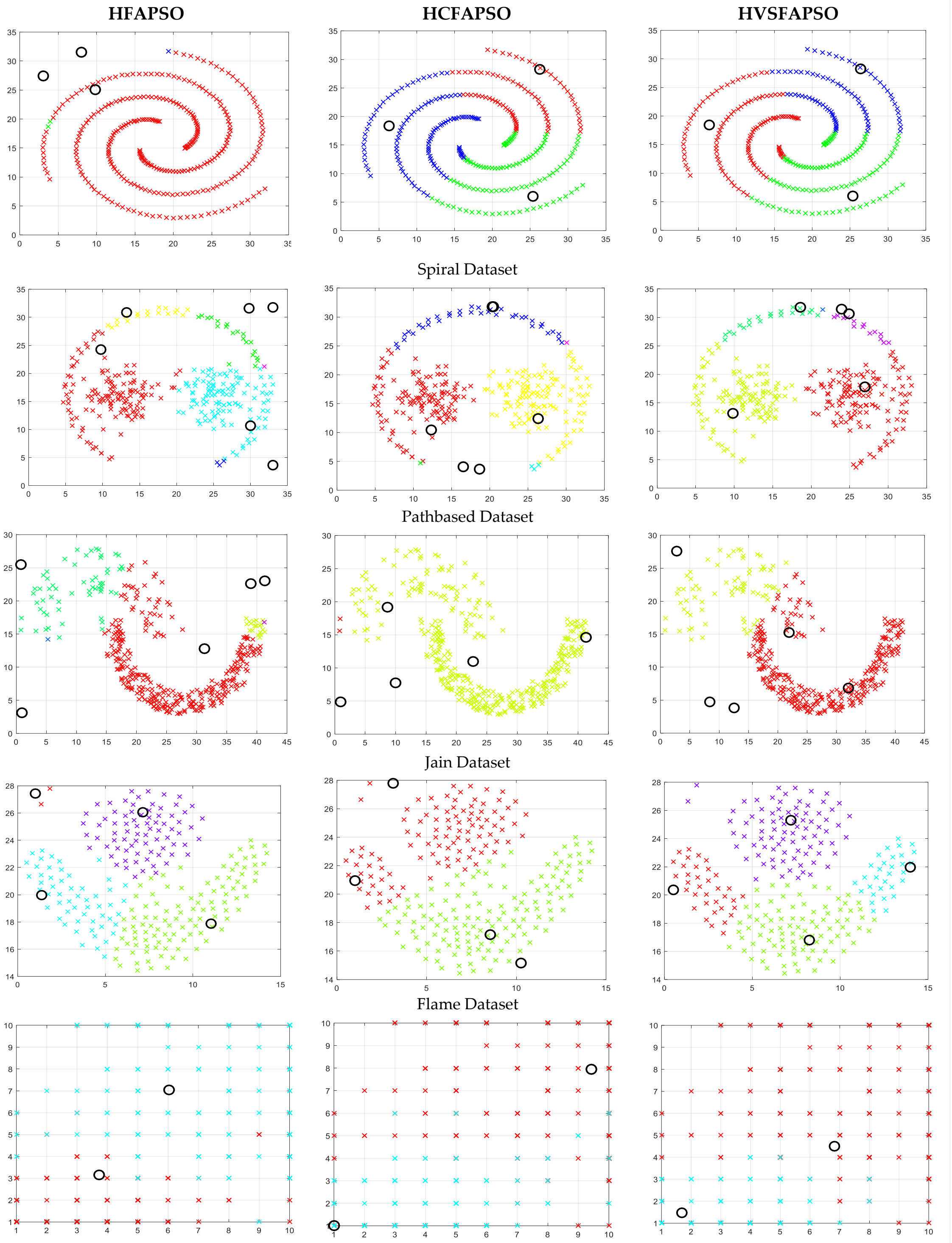


Figure 3. Average execution time taken by HFAPSO on CS and DB indices for all the datasets used over 20 independent runs.



HFAPSO

HCFAPSO

HVSFAPSO

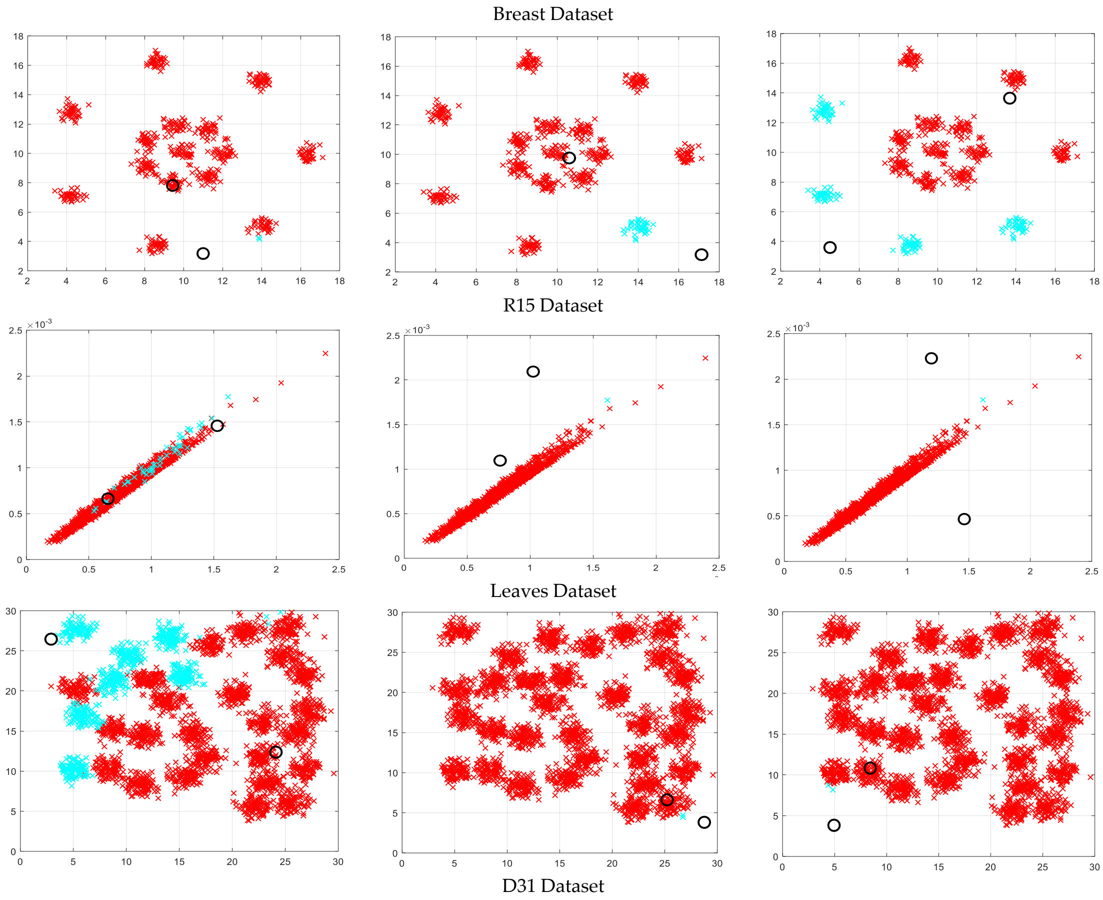
Spiral Dataset

Pathbased Dataset

Jain Dataset

Flame Dataset

Figure 4. *Cont.*



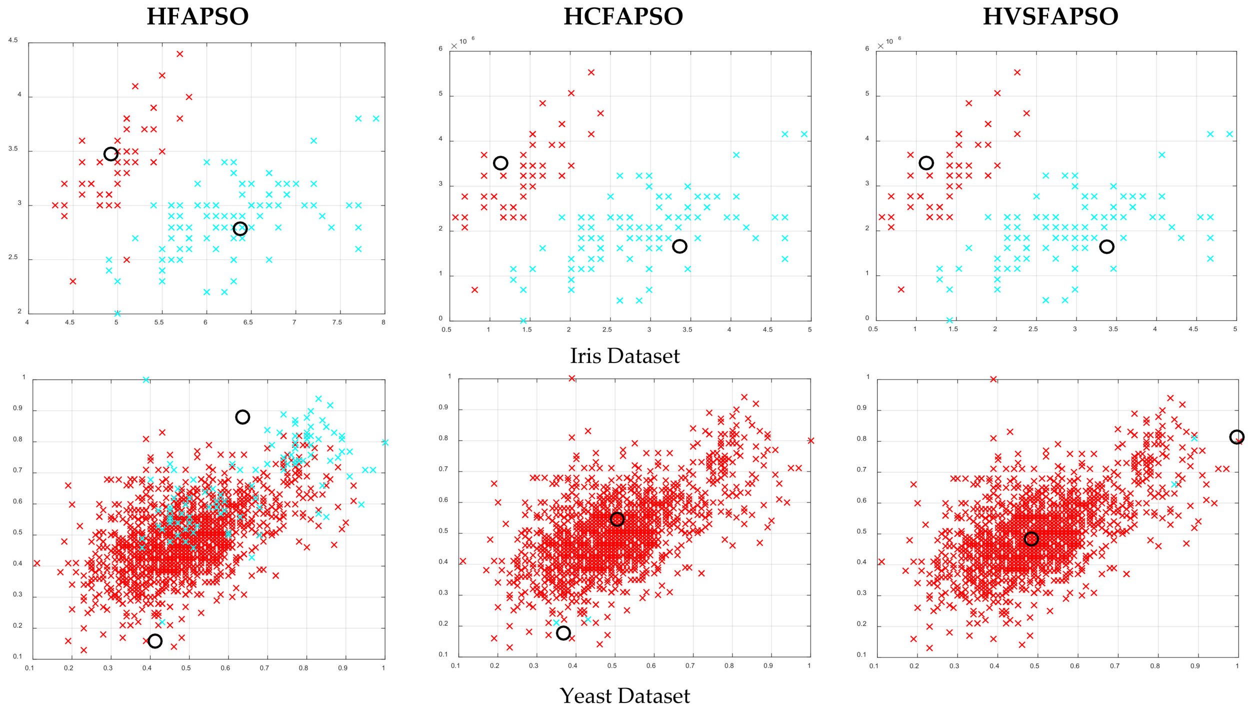
Breast Dataset

R15 Dataset

Leaves Dataset

D31 Dataset

Figure 4. Clustering illustrations for HFAPSO, HCFAPSO and HVSFAPSO clustering algorithms on some selected datasets based on CS index where the black hollow circle represents the number of clusters formed.



HFAPSO

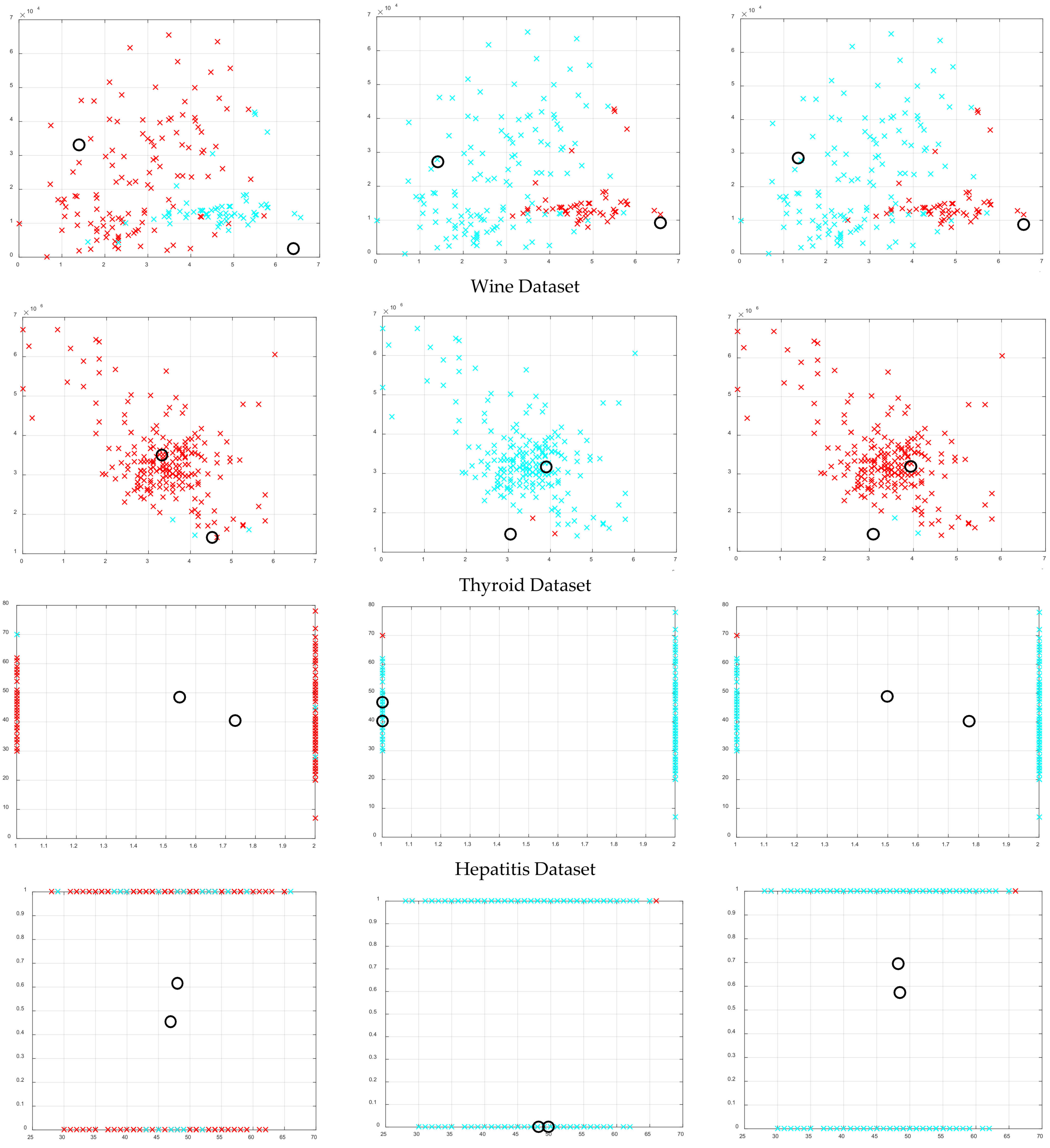
HCFAPSO

HVSFAPSO

Iris Dataset

Yeast Dataset

Figure 5. *Cont.*



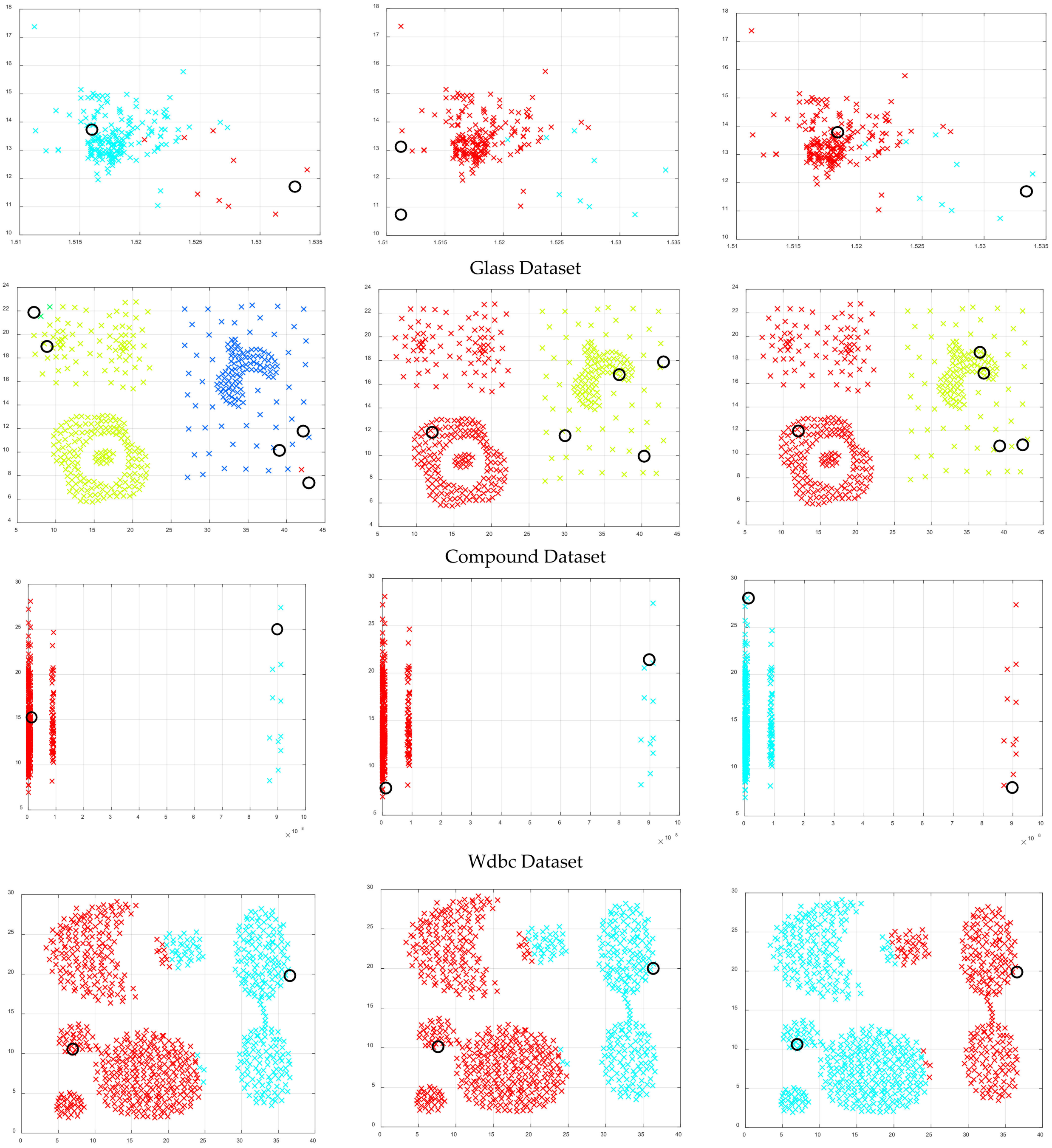
Wine Dataset

Thyroid Dataset

Hepatitis Dataset

Heart Dataset

Figure 5. *Cont.*



Glass Dataset

Compound Dataset

Wdbc Dataset

Aggregation Dataset

Figure 5. Clustering illustrations for HFAPSO, HCFAPSO and HVSFAPSO clustering algorithms on some selected datasets based on DB index, where the black hollow circle represents the number of clusters formed.

In Table 4 the results of the HFAPSO automatic data clustering algorithm for all datasets is clearly presented. The four decimal values are in bold, representing that they constitute the best value. The main aim is to obtain quality results with less execution time to execute every algorithm to obtain the optimal clustering results. In Table 4 it is shown that the CS index outperforms the DB index in some datasets: Spiral, Path based, Flame, R15, Leaves, and D31. Similarly DB index outperforms than CS index in some datasets namely, Iris, Yeast, Wine, Thyroid, Jain, Hepatitis, Heart, Glass, Compound, Breast, Wdbc and Aggregation. In Figure 6 the execution time of HCFAPSO for both CS index and DB index is presented clearly. In the Figure 6 the blue bar represents the CS index whereas the orange bar represents the DB index. However, from the figure it shows that the CS index will take longer to execute in comparison to the DB index to obtain the clustering results. The clustering illustrations of each individual datasets of HFAPSO, HCFAPSO and HVSFAPSO automatic data clustering based on both CS index and DB index is presented in Figures 4 and 5 respectively.

Table 4. Numerical results of HCFAPSO based on CS and DB validity indices over 20 independent runs.

Dataset Used	CS-Index				DB-Index			
	Best	Worst	Average	StDev.	Best	Worst	Average	StDev.
Iris	0.7191	0.9801	0.7478	0.0478	0.57	0.933	0.6960	0.1335
Yeast	0.5204	0.6904	0.5799	0.0810	0.443	1.3793	0.9660	0.2619
Wine	0.8828	1.0289	0.9334	0.0814	0.8002	1.2419	0.9908	0.1299
Thyroid	0.6408	0.6408	0.6408	0.0000	0.4814	1.0111	0.6660	0.1863
Spiral	0.6861	0.9457	0.7653	0.0999	0.7502	0.8092	0.7990	0.0092
Path based	0.6493	1.0068	0.7685	0.1301	0.673	0.7303	0.6854	0.0191
Jain	0.6546	0.7724	0.6781	0.0418	0.65	0.6778	0.6555	0.0078
Hepatitis	0.5298	0.5298	0.5298	0.0000	0.4319	0.4865	0.4500	0.0212
Heart	0.5974	0.5974	0.5974	0.0000	0.456	0.6291	0.5249	0.0616
Glass	0.0607	0.0607	0.0607	0.0000	0.4017	0.9068	0.7263	0.1342
Flame	0.3948	1.0767	0.5336	0.2206	0.7682	0.8142	0.7818	0.0107
Compound	0.7179	0.7732	0.7672	0.0144	0.4931	0.5568	0.5101	0.0201
Breast	0.6862	1.1681	1.0525	0.0962	0.653	1.3761	0.9408	0.2951
Wdbc	0.0712	0.0712	0.0712	0.0000	0.0507	0.0796	0.0554	0.0073
R15	0.6876	0.9395	0.7171	0.0528	0.7299	0.9042	0.7997	0.0591
Leaves	0.4919	0.8196	0.6049	0.1299	0.6205	1.6482	1.0986	0.4523
D31	0.7713	0.9634	0.8076	0.1157	0.793	0.9184	0.8455	0.0368
Aggregation	0.735	1.0323	0.8577	0.0936	0.72	0.8068	0.7348	0.0200

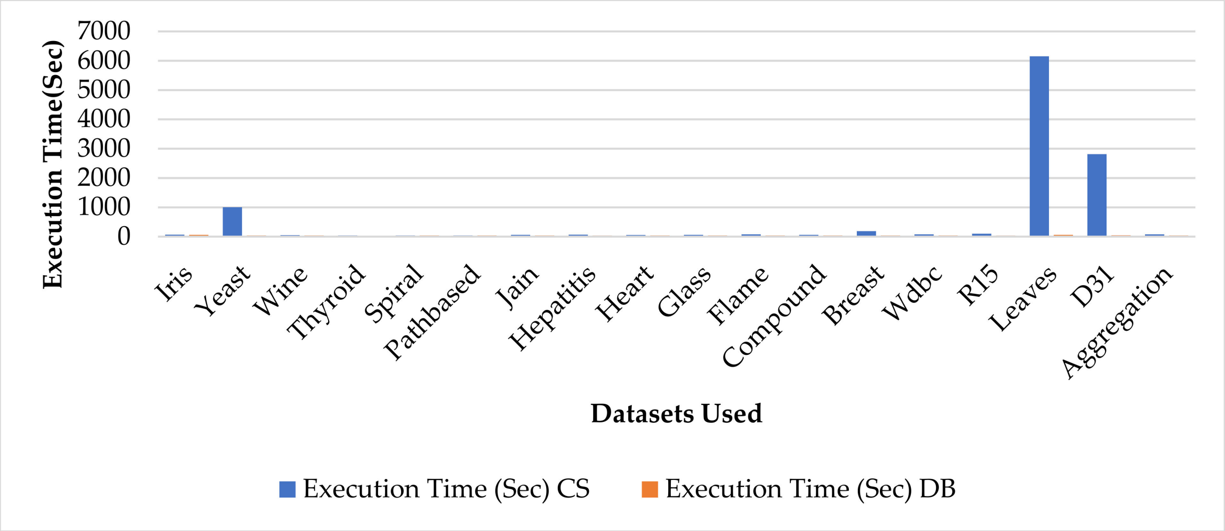


Figure 6. Average execution time taken by HCFAPSO on CS and DB indices for all the datasets used over 20 independent runs.

In Table 5 the results of the HVSFAPSO Automatic data clustering algorithm for all datasets are clearly presented. The four decimal values are in bold, representing that they constitute the best value. The main aim is to obtain quality results with less execution time to execute every algorithm to obtain the optimal clustering results. In Table 5 it is shown that the CS index outperforms than DB index in some datasets: Spiral, Path based, Jain, Glass, Flame, Breast, R15, Leaves, and D31.Similarly the DB index outperforms than the CS index in some datasets, namely, Iris, Yeast, Wine, Thyroid, Hepatitis, Heart, Compound, Wdbc, Aggregation. In Figure 7 the execution time of HVSFAPSO for both CS index and DB index is presented clearly. In the Figure 7 the blue bar represents the CS index whereas the orange bar represents the DB index. However, from the figure it shows that the CS index will take longer to execute in comparison to the DB index to obtain the clustering results. The clustering illustrations of each individual datasets of of HFAPSO, HCFAPSO and HVSFAPSO automatic data clustering based on both the CS index and the DB index are presented in Figures 4 and 5, respectively.

Table 5. Numerical results for HVSFAPSO based on the CS and DB validity indices over 20 independent runs.

Dataset Used	CS-Index				DB-Index			
	Best	**Worst**	**Average**	**StDev.**	**Best**	**Worst**	**Average**	**StDev.**
Iris	0.7191	0.8828	0.7421	0.0322	0.57	0.9491	0.6694	0.1162
Yeast	0.5204	0.7888	0.5553	0.0770	0.4381	1.1058	0.7978	0.2459
Wine	0.8828	1.2393	0.9252	0.0776	0.5733	1.2492	0.9904	0.2010
Thyroid	0.6408	0.6408	0.6408	0.0000	0.4814	0.9486	0.6433	0.1638
Spiral	0.5287	1.0292	0.7678	0.1251	0.7964	0.8188	0.7647	0.0335
Path based	0.5	0.9482	0.6992	0.1692	0.6261	0.7618	0.6858	0.0301
Jain	0.4905	0.6546	0.6414	0.0608	0.6372	0.6562	0.6263	0.0356
Hepatitis	0.5298	0.5298	0.5298	0.0000	0.4318	0.4909	0.4488	0.0201
Heart	0.5974	0.5974	0.5974	0.0000	0.4518	0.6548	0.5250	0.0660
Glass	0.0607	0.0607	0.0608	0.0009	0.3337	1.0041	0.7568	0.7642
Flame	0.3712	0.9792	0.5640	0.2007	0.6705	0.8154	0.7330	0.0468
Compound	0.5032	0.7732	0.7309	0.0840	0.4931	0.5705	0.5144	0.0267
Breast	0.5996	1.1565	0.9094	0.2375	0.6519	1.3643	0.9608	0.3128
Wdbc	0.0712	0.0712	0.0712	0.0000	0.0507	0.0598	0.0531	0.0034
R15	0.6876	0.9077	0.7126	0.0440	0.714	0.9096	0.7866	0.0605
Leaves	0.4919	0.6965	0.5132	0.0613	0.5854	1.5006	1.0215	0.4290
D31	0.711	0.9539	0.8062	0.0852	0.7929	0.9297	0.8392	0.0436
Aggregation	0.7352	1.0737	0.8469	0.1043	0.7199	0.7739	0.7338	0.0163

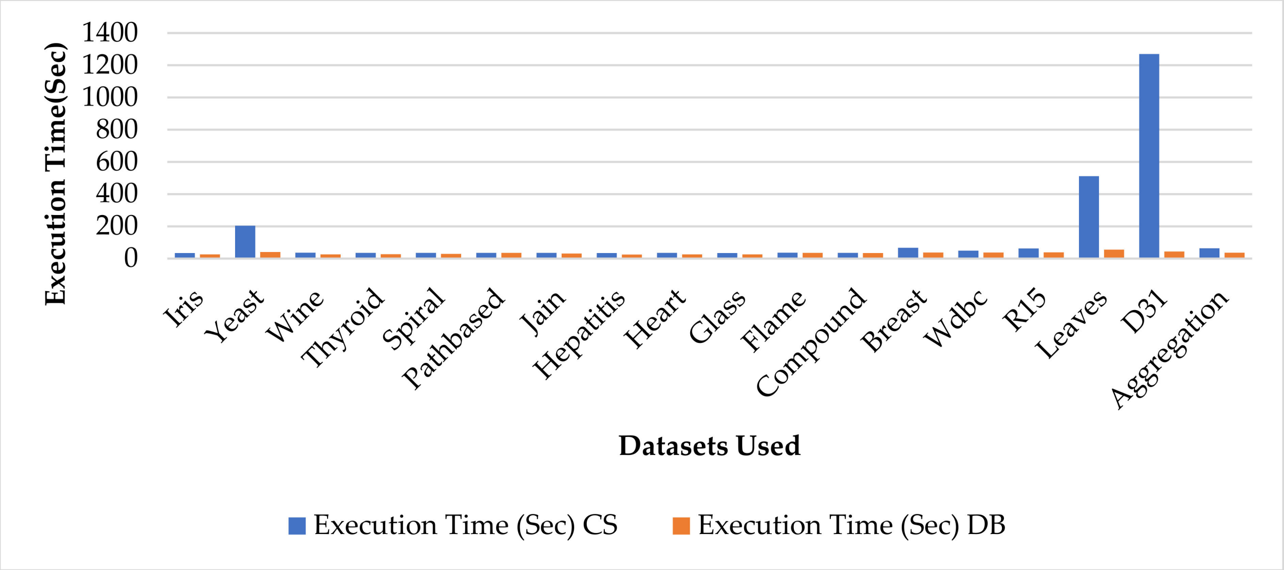


Figure 7. Average execution time taken by HVSFAPSO on CS and DB indices for all the datasets used over 20 independent runs.

5. Comparison Study of HFASO, HCFAPSO and HVSFAPSO Automatic Clustering Algorithms

In this section the comparison results of three hybrid clustering algorithms have been discussed. In Table 6, the mean value and standard deviation for both CS and DB indexes have been given. The clustering illustration of all eighteen datasets for both CS index and DB index has been given in Figures 4 and 5 respectively. The convergence curves for all three-hybrid algorithms for both the CS and DB indexes have been shown in Figures 8 and 9, respectively.

Table 6. Result Comparison of HFAPSO, HCFAPSO and HVSFAPSO for Automatic Clustering.

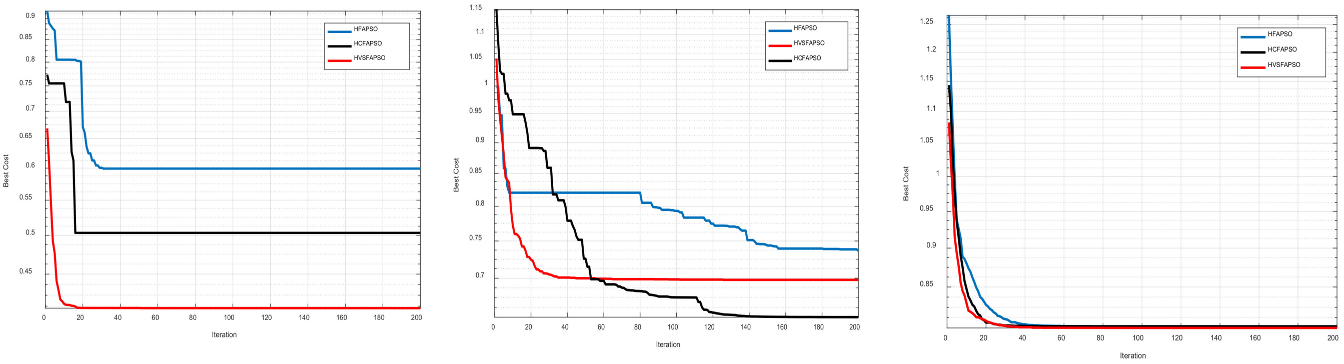
Dataset Used	Algorithm	CS Index		DB Index	
	Methods	Average Value	Standard Deviation	Average Value	Standard Deviation
Iris	HFAPSO	0.7438	0.0382	0.06739	0.1168
	HCFAPSO	0.7478	0.0478	0.6960	0.1335
	HVSFAPSO	0.7421	0.0322	0.6694	0.1162
Yeast	HFAPSO	0.5395	0.0578	0.8143	0.2385
	HCFAPSO	0.5799	0.0810	0.9660	0.2619
	HVSFAPSO	0.5553	0.0770	0.7978	0.2459
Wine	HFAPSO	0.9433	0.0891	0.9835	0.1782
	HCFAPSO	0.9334	0.0814	0.9908	0.1299
	HVSFAPSO	0.9252	0.0776	0.9904	0.2010
Thyroid	HFAPSO	0.6408	0.0000	0.6765	0.2003
	HCFAPSO	0.6408	0.0000	0.6660	0.1863
	HVSFAPSO	0.6408	0.0000	0.6433	0.1638
Spiral	HFAPSO	0.7766	0.1149	0.7633	0.0340
	HCFAPSO	0.7653	0.0999	0.7990	0.0092
	HVSFAPSO	0.7678	0.1251	0.7647	0.0335

Table 6. *Cont.*

Dataset Used	Algorithm	CS Index		DB Index	
Pathbased	HFAPSO	0.6976	0.1757	0.6975	0.0475
	HCFAPSO	0.7658	0.1301	0.6854	0.0191
	HVSFAPSO	0.6992	0.1692	0.6858	0.0301
Jain	HFAPSO	0.6523	0.0729	0.6268	0.0359
	HCFAPSO	0.6781	0.0418	0.6555	0.0078
	HVSFAPSO	0.6414	0.0608	0.6263	0.0356
Hepatitis	HFAPSO	0.5298	0.0000	0.4529	0.0263
	HCFAPSO	0.5298	0.0000	0.4500	0.0212
	HVSFAPSO	0.5298	0.0000	0.4488	0.0201
Heart	HFAPSO	0.5974	0.0000	0.5150	0.0617
	HCFAPSO	0.5974	0.0000	0.5249	0.0616
	HVSFAPSO	0.5974	0.0000	0.5250	0.0660
Glass	HFAPSO	0.0607	0.0000	0.7447	0.7379
	HCFAPSO	0.0607	0.0000	0.7263	0.1342
	HVSFAPSO	0.0608	0.0009	0.7568	0.7642
Flame	HFAPSO	0.5195	0.1860	0.7359	0.0501
	HCFAPSO	0.5336	0.2206	0.7818	0.0107
	HVSFAPSO	0.5640	0.2007	0.7330	0.0468
Compound	HFAPSO	0.7019	0.0989	0.5170	0.0281
	HCFAPSO	0.7672	0.0144	0.5101	0.0201
	HVSFAPSO	0.7309	0.0840	0.5144	0.0267
Breast	HFAPSO	0.8844	0.2382	0.9730	0.3284
	HCFAPSO	1.0525	0.0962	0.9408	0.2951
	HVSFAPSO	0.9094	0.2375	0.9608	0.3128
Wdbc	HFAPSO	0.0712	0.0000	0.0801	0.1035
	HCFAPSO	0.0712	0.0000	0.0554	0.0073
	HVSFAPSO	0.0712	0.0000	0.0531	0.0034
R15	HFAPSO	0.7235	0.0685	0.7860	0.0565
	HCFAPSO	0.7171	0.0528	0.7997	0.0591
	HVSFAPSO	0.7126	0.0440	0.7866	0.0605
Leaves	HFAPSO	0.5124	0.0530	1.0337	0.4390
	HCFAPSO	0.6049	0.1299	1.0986	0.4523
	HVSFAPSO	0.5132	0.0613	1.0215	0.4290
D31	HFAPSO	0.8822	0.1481	0.8350	0.0394
	HCFAPSO	0.8076	0.1157	0.8455	0.0368
	HVSFAPSO	0.8062	0.0852	0.8392	0.0436
Aggregation	HFAPSO	0.8272	0.1027	0.7354	0.0185
	HCFAPSO	0.8577	0.0936	0.7348	0.0200
	HVSFAPSO	0.8469	0.1043	0.7338	0.0163

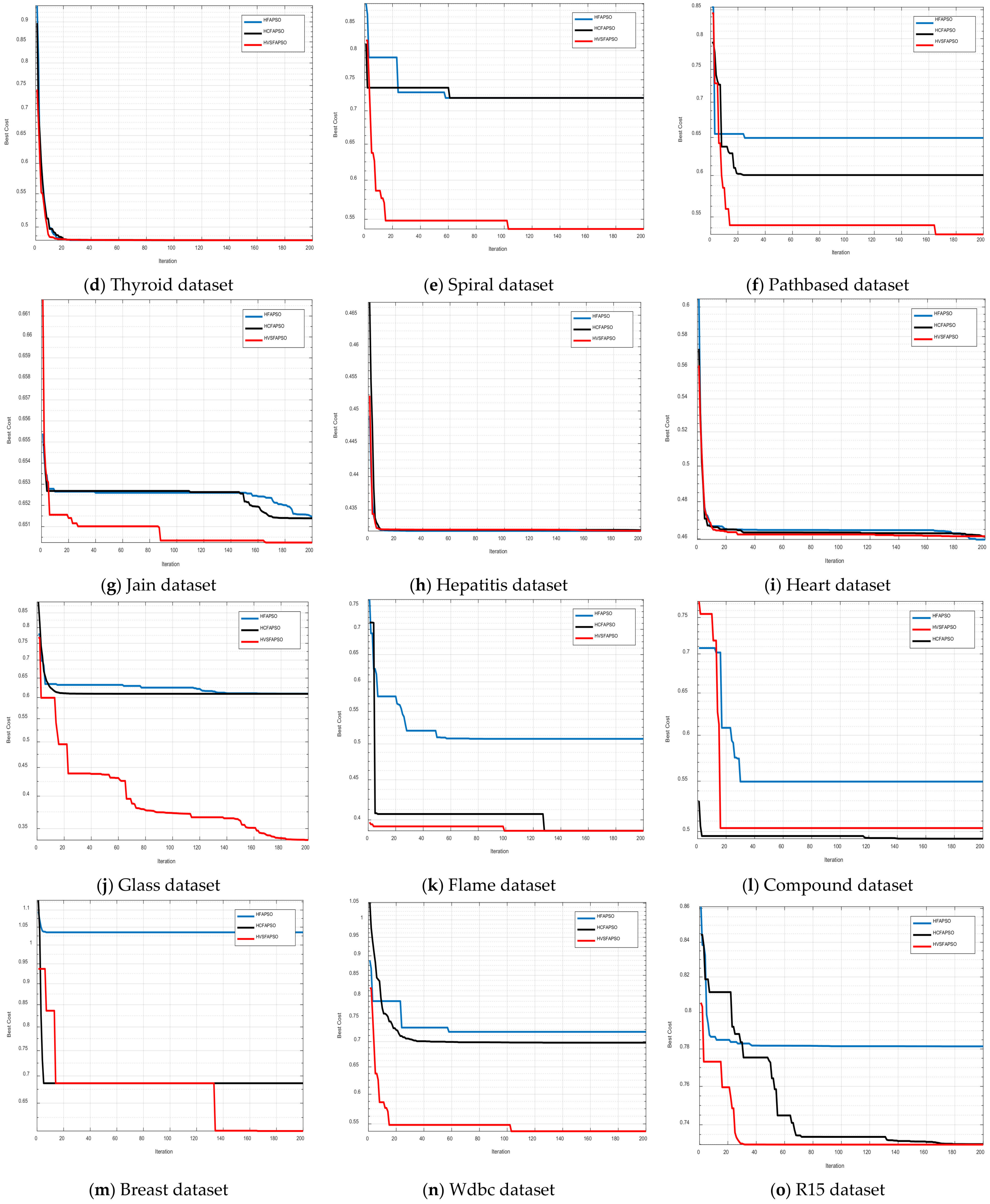
The clustering illustrations for HFAPSO, HCFAPSO and HVSFAPSO on some selected datasets based on the CS index have been presented in Figure 4. For the Spiral dataset from the above figure, it shows that in using the FAPSO clustering algorithm, we have some outliers in the clustering, whereas both HCFAPSO and HVSFAPSO clustering algorithms give superior results with three perfect clusters and no outliers. For the Path-based dataset the HFAPSO clustering algorithm gives four clusters with few magenta and blue outliers and by using HCFAPSO clustering algorithm, three distinct clusters are found with very few outliers, whereas in HVSFAPSO, four clearly separated clusters are generated. For the Jain dataset, HCFAPSO and HVSFAPSO give good clustering in comparison to HFAPSO clustering algorithms. For the Flame dataset, both HCFAPSO and HVSFAPSO clustering algorithms are superior to the HFAPSO clustering algorithm, having two and four clear clusters, respectively. For the breast dataset, by using both HCFAPSO and HVSFAPSO clustering algorithms, better clustering results obtained than the existing HFAPSO clustering algorithm. For R15 dataset the HFAPSO clustering algorithm gives clusters with outliers, whereas by using both HCFAPSO and HVSFAPSO clustering algorithms, good clustering results are observed with no outliers. For the Leaves dataset both HFAPSO and HVSFAPSO clustering algorithms outperform the HFAPSO clustering algorithm with good clusters, having far fewer outliers which are not noticeable. Similarly, for D31 dataset both HCFAPSO and HVSFAPSO clustering algorithms outperform the HFAPSO clustering algorithm with one cluster each.

The clustering illustrations for HFAPSO, HCFAPSO and HVSFAPSO on some selected datasets based on the DB index have been presented in Figure 5. For Iris, Wine, Yeast, Thyroid, Hepatitis and Heart datasets, HCFAPSO and HVSFAPSO clustering algorithms outperform the HFAPSO clustering algorithm, having two clusters each with very few blue and red outliers which can be ignored. For the Compound dataset, the HCFAPSO and HVSFAPSO algorithm give superior results than the HFAPSO clustering algorithm having three clusters without outliers. The exception cases are Glass, Wdbc and Aggregation datasets, in which all three algorithms give good clustering results, having two clusters each, but for better clustering results, both HCFAPSO and HVSFAPSO clustering algorithms can be considered [24].



(**a**) Iris dataset (**b**) Yeast dataset (**c**) Wine dataset

Figure 8. *Cont.*



(**d**) Thyroid dataset

(**e**) Spiral dataset

(**f**) Pathbased dataset

(**g**) Jain dataset

(**h**) Hepatitis dataset

(**i**) Heart dataset

(**j**) Glass dataset

(**k**) Flame dataset

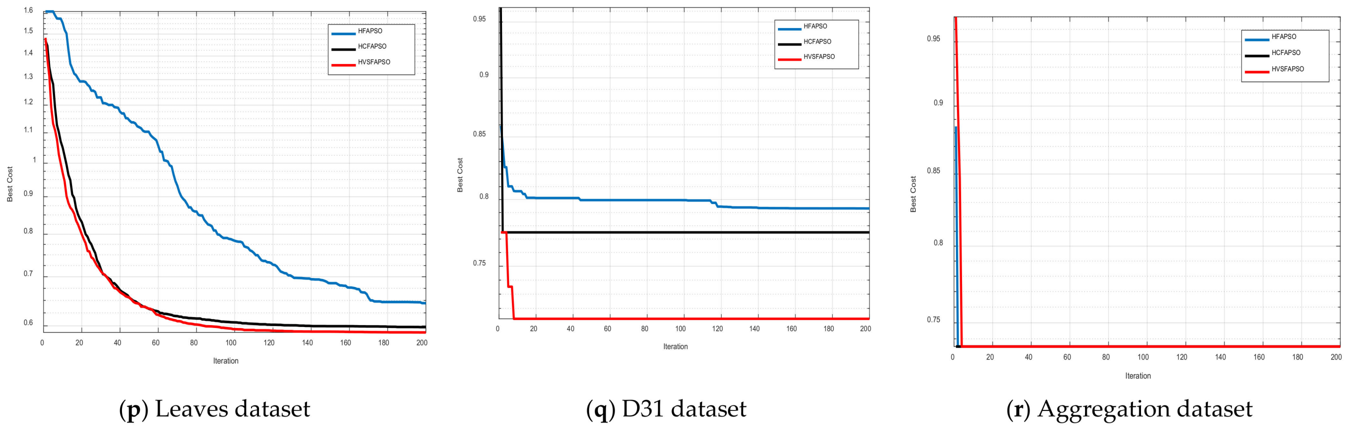
(**l**) Compound dataset

(**m**) Breast dataset

(**n**) Wdbc dataset

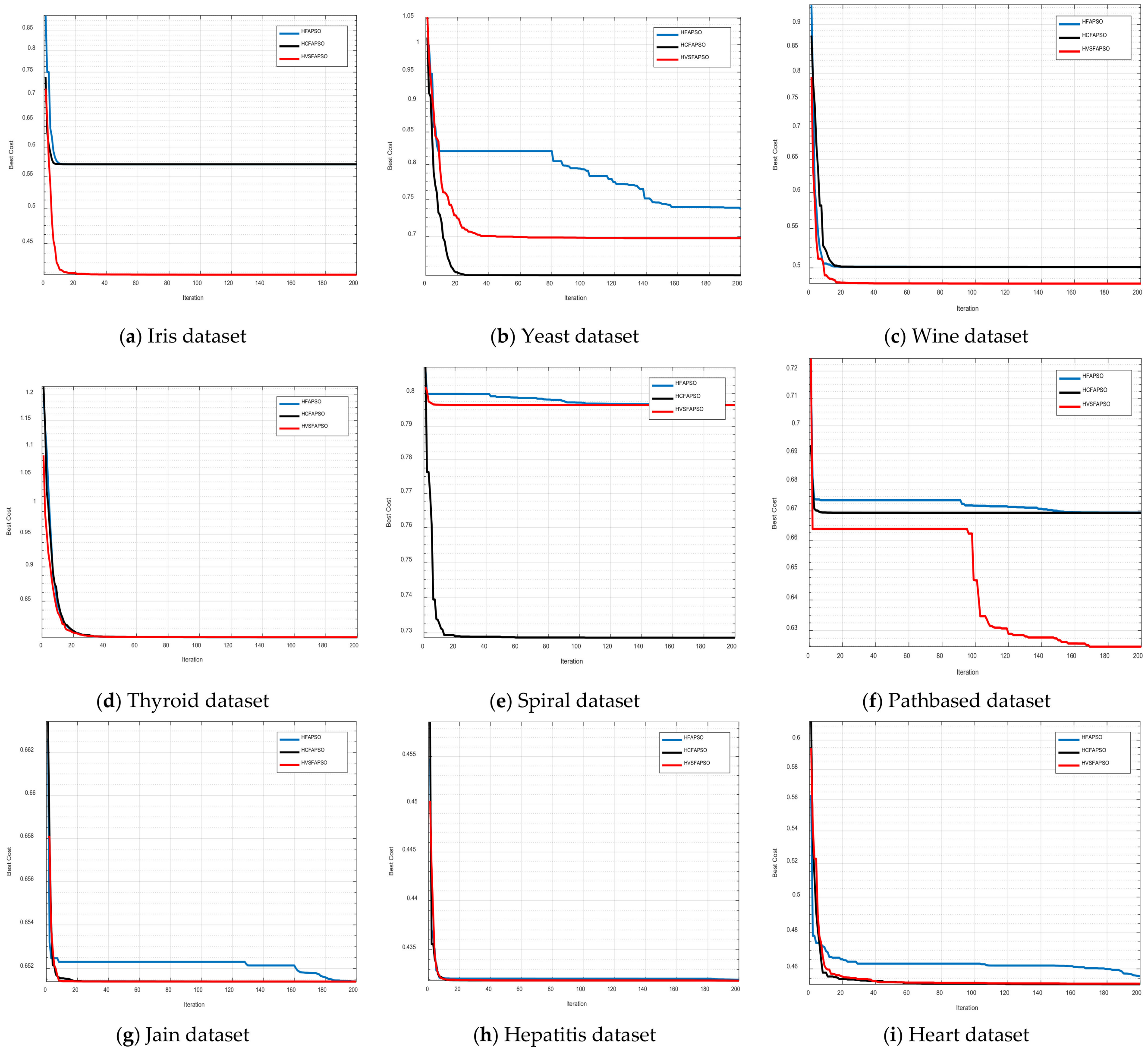
(**o**) R15 dataset

Figure 8. *Cont.*



(**p**) Leaves dataset (**q**) D31 dataset (**r**) Aggregation dataset

Figure 8. Convergence curves for all used datasets on CS-index.

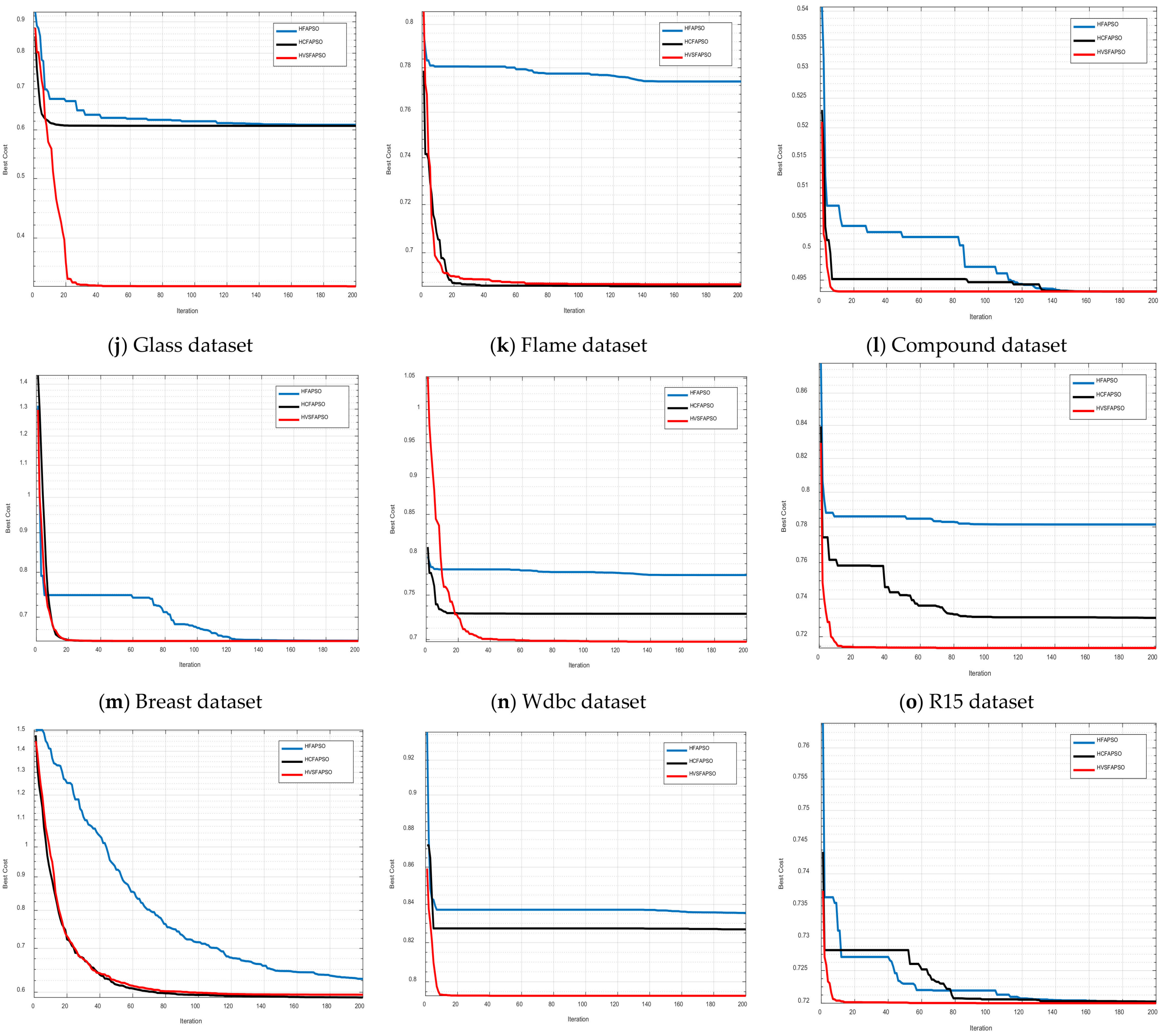


(**a**) Iris dataset (**b**) Yeast dataset (**c**) Wine dataset

(**d**) Thyroid dataset (**e**) Spiral dataset (**f**) Pathbased dataset

(**g**) Jain dataset (**h**) Hepatitis dataset (**i**) Heart dataset

Figure 9. *Cont.*



(**j**) Glass dataset (**k**) Flame dataset (**l**) Compound dataset

(**m**) Breast dataset (**n**) Wdbc dataset (**o**) R15 dataset

(**p**) Leaves dataset (**q**) D31 dataset (**r**) Aggregation dataset

Figure 9. Convergence curves for all used datasets on DB index.

In Figures 8 and 9 the equivalent graphical convergence comparison curves of the three hybrid clustering algorithms discussed above for eighteen datasets based on the CS index and DB index are presented, respectively. For both the CS index and the DB index, the HCFAPSO clustering algorithm and HVSFAPSO clustering algorithm give better convergence results than the existing HFAPSO clustering algorithm. However, the HVSFAPSO clustering algorithms converge faster and also give smoother convergence curves than HCFAPSO.

To further find a better experimental result, the Wilcoxon rank-sum test has been taken. In Table 7 the p-values for both CS and DB indexes are presented in pairwise analysis in terms of HCFAPSO vs. HFAPSO and HVSFAPSO vs. HFAPSO. The Wilcoxon rank-sum test also contrasts with the null hypothesis, which holds that two values are samples from continuous distributions with equal medians. Almost all values in Table 7 are less than 0.05, which shows significance level of 5%. This significance level provides better support over the null hypothesis and demonstrates the statistical importance of the proposed HCFAPSO and HVSFAPSO clustering results.

Table 7. *p*-Values generated by using Wilcoxon rank-sum test for equal medians.

Datasets	CS-Index		DB-Index	
	HCFAPSO vs. HFAPSO	HVSFAPSO vs. HFAPSO	HCFAPSO vs. HFAPSO	HVSFAPSO vs. HFAPSO
Iris	0	0	0.222	0.222
Yeast	0	0	0.050	0.011
Wine	0	0	0.031	0.012
Thyroid	0	0	0.012	0.015
Spiral	0.002	0.011	0	0
Pathbased	0.212	0	0	0
Jain	0.411	0	0	0
Hepatitis	0.003	0	0.310	0.311
Heart	0.008	0.025	0	0
Glass	1	0.022	0	0
Flame	0	0	0	0.001
Compound	0	0.011	0.453	0.156
Breast	0.012	0	0.178	0
Wdbc	0	0	0	0
R15	0.255	0.012	0.004	0
Leaves	0	0	0.006	0.012
D31	0	0.016	0	0.121
Aggregation	0.021	0	0	0

6. Discussions

1. In this work, two hybrid algorithms have been proposed to automatically cluster datasets by exploring the FA and PSO. Those algorithms start with a population of randomly generated individuals and try to optimize the population over a sequence of generations until the optimum solution is obtained.
2. The proposed algorithms initially focus on searching for the best number of clusters and gradually move to obtain the globally optimal cluster centers.
3. Two types of continuous fitness functions are designed on the basis of current clustering validation indices and the penalty functions designed for minimizing the amount of noise and to control the number of clusters [1].
4. The CS and DB validity indexes are used as fitness functions to calculate the fitness of each firefly. The cluster center has been changed by the position of each firefly. The distance between two fireflies has been calculated using the Euclidean distance, where the position of each firefly has been updated using the fitness parameter.
5. The effectiveness of the proposed clustering strategy has shown its efficiency with respect to the convergence graph, mean value, standard deviation value and the *p*-value generated by the Wilcoxon rank-sum test in comparison with HFAPSO to establish its effectiveness.

7. Conclusions and Future Scope

In past, many traditional clustering algorithms have been designed by researchers, but these have been unable to solve complex real-time data clustering problems. Most clustering approaches would require a clearly specified objective function. Two modified hybrid automatic clustering algorithms, namely HCFAPSO and HVSFAPSO, have been proposed to overcome the data clustering issues in real time. Subsequently the performance analysis of both the proposed automatic clustering algorithms has been performed based on two validity indices: CS index and DB index [6,7]. The CS index proves itself the better clustering performance measure than the DB index, with little more execution time taken than the DB index. The results obtained using both the proposed algorithms have been compared with the existing HFAPSO automatic clustering algorithms. In terms of better convergence speed, better diversification, and ability to adapt to a variety of optimization issues, both the proposed modified automatic clustering algorithms will outperform the existing HFAPSO automatic data clustering algorithm. In addition to this, both the proposed HCFAPSO and HVSFAPSO automatic data clustering algorithms will give superior results to HFAPSO based on the optimal number of clusters. The proposed HCFAPSO and HVSFAPSO algorithms can be used efficiently to solve automatic data clustering issues. However, in certain problems a little more emphasis must be given to the proposed HVSFAPSO to obtain better results. In future, HCFAPSO and HVSFAPSO can be applied efficiently in different complex optimization areas for better performance.

Author Contributions: Conceptualization, M.B., A.S., D.M., P.K.M., J.S., P.N.S. and M.F.I.; data curation, M.B., A.S. and D.M.; formal analysis, M.B., A.S., D.M., P.K.M. and J.S.; funding acquisition, M.F.I. and J.S.; investigation, P.K.M., J.S., P.N.S. and M.F.I.; methodology, M.B., A.S., D.M., P.K.M., J.S. and P.N.S.; project administration, M.F.I., P.N.S. and J.S.; resources, M.F.I. and J.S.; software, M.B., A.S., D.M., P.K.M., J.S. and P.N.S.; supervision, J.S., P.N.S. and M.F.I.; validation A.S., D.M. and M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Sejong University Research fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study.

Acknowledgments: Jana Shafi would like to thank the Deanship of Scientific Research, Prince Sattam bin Abdul Aziz University, for supporting this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guan, C.; Yuen, K.K.F.; Coenen, F. Particle swarm Optimized Density-based Clustering and Classification: Supervised and unsupervised learning approaches. *Swarm Evol. Comput.* **2019**, *44*, 876–896. [CrossRef]
2. Majhi, S.K.; Biswal, S. Optimal cluster analysis using hybrid K-Means and Ant Lion Optimizer. *Karbala Int. J. Mod. Sci.* **2018**, *4*, 347–360. [CrossRef]
3. Agbaje, M.B.; Ezugwu, A.E.; Els, R. Automatic Data Clustering Using Hybrid Firefly Particle Swarm Optimization Algorithm. *IEEE Access* **2019**, *7*, 184963–184984. [CrossRef]
4. Ezugwu, A.E.-S.; Agbaje, M.B.; Aljojo, N.; Els, R.; Chiroma, H.; Elaziz, M.A. A Comparative Performance Study of Hybrid Firefly Algorithms for Automatic Data Clustering. *IEEE Access* **2020**, *8*, 121089–121118. [CrossRef]
5. Sarangi, S.K.; Panda, R.; Sarangi, A. Crazy firefly algorithm for function optimization. In Proceedings of the 2nd International Conference on Man and Machine Interfacing (MAMI), Bhubaneswar, India, 21–23 September 2017; pp. 1–5.
6. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *1*, 224–227. [CrossRef] [PubMed]
7. Chou, C.-H.; Su, M.-C.; Lai, E. A new cluster validity measure and its application to image compression. *Pattern Anal. Appl.* **2004**, *7*, 205–220. [CrossRef]
8. Deeb, H.; Sarangi, A.; Mishra, D.; Sarangi, S.K. Improved Black Hole optimization algorithm for data clustering. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 5020–5029. [CrossRef]
9. Sharma, M.; Chhabra, J.K. Sustainable automatic data clustering using hybrid PSO algorithm with mutation. *Sustain. Comput. Inform. Syst.* **2019**, *23*, 144–157. [CrossRef]
10. Zhou, Y.; Wu, H.; Luo, Q.; Abdel-Baset, M. Automatic data clustering using nature-inspired symbiotic organism search algorithm. *Knowl.-Based Syst.* **2019**, *163*, 546–557. [CrossRef]
11. Rajah, V.; Ezugwu, A.E. Hybrid Symbiotic Organism Search algorithms for Automatic Data Clustering. In Proceedings of the Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 9–10 March 2022; pp. 1–9.
12. Pacheco, T.M.; Gonçalves, L.B.; Ströele, V.; Soares, S.S.R.F. An Ant Colony Optimization for Automatic Data Clustering Problem. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
13. Yang, X.S. *Firefly Algorithm, Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Cambridge, MA, USA, 2008; pp. 79–90.

14. Yang, X.S. Firefly Algorithms for Multimodal Optimization. In *Stochastic Algorithms: Foundations and Applications*; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5792, pp. 169–178.
15. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
16. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
17. Xia, X.; Gui, L.; He, G.; Xie, C.; Wei, B.; Xing, Y.; Wu, R.; Tang, Y. A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm. *J. Comput. Sci.* **2018**, *26*, 488–500. [CrossRef]
18. Samal, S.; Sarangi, S.K.; Sarangi, A. Analysis of Adaptive Mutation in Crazy Particle Swarm Optimization. In Proceedings of the 2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE), Keonjhar, India, 29–31 July 2020; pp. 1–5.
19. Alhmouz, O.; Abrar, S.; Iqbal, N.; Zerguine, A. A Variable Step-Size Blind Equalization Algorithm Based on Particle Swarm Optimization. In Proceedings of the 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 1357–1361.
20. Zhao, J.; Chen, W.; Ye, J.; Wang, H.; Sun, H.; Lee, I. Firefly Algorithm Based on Level-Based Attracting and Variable Step Size. *IEEE Access* **2020**, *8*, 58700–58716. [CrossRef]
21. Bache, K.; Lichman, M. *UCI Machine Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2013.
22. Clustering Basic Benchmark. Available online: http://cs.joensuu.fi/sipu/datasets/ (accessed on 25 November 2019).
23. Samanta, S.R.; Mallick, P.K.; Pattnaik, P.K.; Mohanty, J.R.; Polkowski, Z. (Eds.) *Cognitive Computing for Risk Management*; Springer: Cham, Switzerland, 2022.
24. Mukherjee, A.; Singh, A.K.; Mallick, P.K.; Samanta, S.R. Portfolio Optimization for US-Based Equity Instruments Using Monte-Carlo Simulation. In *Cognitive Informatics and Soft Computing. Lecture Notes in Networks and Systems*; Mallick, P.K., Bhoi, A.K., Barsocchi, P., de Albuquerque, V.H.C., Eds.; Springer: Singapore, 2022; Volume 375. [CrossRef]