Int. J. Advance Soft Compu. Appl, Vol. 9, No. 3, Nov 2017
ISSN 2074-8523

An Enhanced Binary Particle Swarm Optimization (EBPSO) Algorithm Based A V-shaped Transfer Function for Feature Selection in High Dimensional data

Idheba Mohamad Swesi[1] and Azuraliza Abu Bakar[1]

[1]Centre for Artificial Intelligence Technology,
Faculty of Information Science and Technology,
University Kebangsaan Malaysia, Selangor Darul Ehsan, Malaysia
e-mail: ana180611@yahoo.com, azuraliza@ukm.edu.my

Abstract

A general problem faced in classification is a small number of samples compared with the large number of genes. In this case, the feature selection (FS) process becomes a challenging task to improve the classification performance by reducing the dimensionality. Particle swarm optimisation (PSO) is a powerful method for solving FS problems. The main ingredient of the binary PSO is its transfer function that allows mapping a continuous space to a discrete space. A sigmoid function was used to update positions in BPSO. However, due to its way of updating positions, this function is not very effective to dodge local minima and speed up the convergence. Thus, this paper suggests an enhanced BPSO, through the FS approach. The study employs a V type transfer function and a special method for updating positions. In addition, a hybrid FS method that integrates the information gain (IG) as a filter approach with the wrapper approach (EBPSO) is proposed. In a hybrid model, feature subsets are ranked based on their significance in a decreasing order. The EBPSO is applied to these subsets and then FS is performed. The proposed algorithm is tested on six microarray datasets and the results verify its superiority. Also, the results of the proposed hybrid FS model, supports its effectiveness because the model produced a small feature subset that showed high classification performance.

Keywords: *Feature Selection, Gene Expression Data, Hybrid Features Selection Particle Swarm Optimisation, Transfer Function.*

1 Introduction

In fields such as machine learning and data mining, a large number of features are associated with the datasets. One such example is the cancer microarray data that consist of a large number of features (genes). However, a majority of the features are irrelevant or redundant, which may bring down the classification performance. The aim here is to maintain only those genes that would contribute to arriving at a decision regarding the output classes. Feature selection is responsible for selecting a subset with the relevant features that would help achieve a similar or better performance than the complete features could [11]. Also, it can significantly improve the classification accuracy, reduce data dimensionality, minimise computational costs, and enhance output comprehensibility by removing the irrelevant and redundant features [12]. The current feature selection methods are classified into two: filter and wrapper. The filter approach depends on several parameters such as information, distance, dependence and consistency [19] to assess the classification of feature subsets. The wrapper approach employs the algorithm's classification performance to assess the goodness of the feature subsets [19]. As the wrapper approach involves repeated learning steps, it is more expensive than filter approach.

Due to the low commotional costs, the literature is flooded with many effective filtering methods that have already been applied. Among these, the most popular one is the information measures due to its efficiency and wide application. The information measures can be employed in four main ways: (1) Ranking the individual features prior to employing an EC technique. (2) In the local search, make use of the information measure for the memetic algorithm. (3) Integrate an information measure into the updating/search mechanism. (4) Employ information theory to develop a fitness function in an EC algorithm. Feature selection (FS) is considered a NP-hard problem. This is because with the rise in the number of features, the search space expands exponentially, 2^N, where N is the number of features [4]. As the number of features grows, the task of performing FS becomes more challenging as selecting an optimal feature subset from among the given datasets requires exhaustive searching. However, the exhaustive search is a time-consuming process [18].

Over the past few decades, for feature selection, various search techniques such as greedy search, complete search, random search ,and heuristic search have been proposed [11, 13–15]. However, the problem of local optima and/or high computational costs continues to plague most of the current feature selection methods [16, 17]. Therefore, to solve the feature selection problems, an efficient global search technique is needed. Evolutionary computation (EC) techniques are well known for their global search ability and are employed in many optimisation problems. These techniques include ant colony optimisation (ACO) [7], genetic algorithm (GA) [5] and particle swarm optimisation (PSO) [9, 12].

Among the EC techniques employed, particle swarm optimisation (PSO) is the most popular as it is used in solving numerous optimisation problems like multi-objective optimisation [38], constrained optimisation [40], feature selection [7] and classification [39]. In this paper, the capacity of PSO is evaluated in feature selection and a few improvements are suggested that would help enhance the results. There are many research studies in the literature validating the advantages of employing PSO to solve issues related to high dimensionality and FS [8, 12, 19, 22]. For example, Tran et al. [19] put forward a new PSO-based FS approach, in which their algorithm's performance is evaluated and compared with three other PSO-based methods on five datasets comprising varied number of features ranging from 5,469 to 12,600. Experimental results support the greater computational ability of the proposed algorithm than others. Next, based on hamming distance, the binary particle swarm optimisation for feature selection in gene expression data is proposed in Ref. [12]. The fact that BPSO-HD was able to achieve higher classification accuracy than other methods was clear from the results on three benchmark high-dimensional dataset consisting of varying feature cardinality from 2,000 to 7,000. However, to validate the effectiveness of the proposed algorithm, employing only three datasets may not be sufficient. Moreover, the transfer function employed for BPSO is not very effective. This could result in problems related to slow convergence speed and trapping in of local minima. Therefore, to efficiently handle these issues and improve the result's accuracy for gene expression data, the PSO as a feature selection method needs to be enhanced further. In addition, it was found that by combining the filter and wrapper approaches in the hybrid model, better results than BPSO could be yielded. The hybridisation based on the filter approach with PSO was found to speed up the learning process, reduce the data's dimensionality, and improve the classification performance.

Based on the above arguments, the paper first proposes an enhanced binary PSO named EBPSO to enhance the exploitation and exploration of the algorithm. Second, a hybrid feature selection model is demonstrated that combined a well-known filter method with EBPSO. In the first stage of the FS model, a filter method named information gain (IG) is employed to evaluate the importance of the features [28]. In the second stage, the EBPSO is employed to perform feature selection for top ranked features that were selected by using the filter method. The remaining paper is arranged as follows: section 2 explains the related works, section 3 describes the improved BPSO and the hybrid FS model in details, section 4 presents the experiments and results, and section 6 concludes the paper.

2 Background

2.1 Particle Swarm Optimisation (PSO)

Kennedy & Eberhart originally proposed the PSO [1]. The algorithm is based on the population-based social behaviour of fish schools or bird flocks. In PSO, a swarm is defined as a set of random particles that keep moving in search of optimal solutions. In this, every particle draws a potential solution to a problem being solved, and it is related to a position which is characterised by a vector $(x_i = x_{i1}, x_{i2}, \dots, x_{iD})$. The particle velocity is defined by $(v_i = v_{i1}, v_{i2}, \dots, v_{iD})$, where D represents the dimensionality of the search space. When the particle is in motion, the velocity and position are updated automatically based on their own and neighbours' experience. A fitness function is assigned to each particle in the swarm, which is assessed and compared with the *previous best fitness (pbest)* for a particular particle and with the best fitness achieved by the swarm (gbest). After determining the two best fitness's (pbest and gbest), each particle updates its velocity and position based on the following equations:

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_1 * [p_{id} - x_{id}^{t}] + c_2 * r_2 * [p_{gd} - x_{id}^{t}] \tag{1}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{2}$$

where v_{id}^{t} and x_{id}^{t} represent the velocity and position, respectively, of particle i at iteration t in the dimension d . p_{id} and p_{gd} represent pbest and gbest positions, respectively; in dimension d. W signifies the inertia weight factor that is employed to regulate the balance between exploitation and exploration. c_1 and c_2 represent the acceleration constants, and r_1 and r_2 signify the random numbers that are uniformly distributed between 0 and 1. PSO was originally developed to handle the continuous optimisation problems. Eberhart & Kennedy [2] designed another version of PSO called BPSO to further extend its application. BPSO can tackle discrete optimisation problems as well as perform feature selection. A binary bit string is employed to encode the positions of the particle in BPSO, where each bit characterises a feature, i.e., the bit value 1 signifies a selected feature, while the bit value 0 signifies a non-selected feature. The transformation of the real-valued velocities to probability values in the range of (0, 1) requires a sigmoid transfer function, whereas the following formulae are employed to update the position of each particle:

$$x_{id}^{t+1} = \begin{cases} 1, & if\ rand() < s(v_{id}^{t+1}) \\ 0, & otherwise \end{cases} \tag{3}$$

where

$$s(v_{id}^{t+1}) = \frac{1}{1 + e^{-v_{id}^{t+1}}} \tag{4}$$

where $s(v_{id}^{t+1})$ represents a sigmoid transformation, *rand* signifies a random number picked from a uniform distribution in [0,1]. The following are the general steps of the BPSO algorithm:

a. Initialisation of all the particles containing random values.
b. Calculate the particles velocities using Eq. (1).
c. Define the probabilities for changing the elements of position vectors by employing Eq. (4).
d. Use the rules in Eq. (3) to update position vectors accordingly.

e. Repeat steps b–d until the termination criterion is met.

2.2 Related Works on BPSO for Feature Selection

To address the FS issues, several evolutionary computation (EC) techniques such as genetic algorithms (GA) [5], ant colony optimisation (ACO) [7], genetic programming (GP) [6] and particle swarm optimisation [8] have been applied. When compared with other experimental techniques, PSO provides benefits such as high accuracy, quick convergence and low computational costs [12]. This algorithm can create many feature subsets by its own to select the best subset with most informative features. Several modified techniques have been proposed to enhance the performance of PSO for FS. The work of [29] for example suggested new updating mechanisms and new initialisation strategies to enhance PSO performance. Experiments conducted on 14 UCI datasets reveal that the proposed PSO method, when compared to traditional methods, could achieve greater classification performance with smaller number of features. Other study mostly concentrates on the classification performance applicability in the fitness function is proposed in [17]. Using 10 UCI data sets, the proposed method's performance is compared with that of the genetic algorithm (GA), standard PSO, and the grid search methods. Based on the numerical results, the proposed algorithm was found to perform considerably better than the other three methods when it comes to the number of selected features and classification accuracy. However, these approaches focused more on datasets that contain comparatively small number of features (less than 700 features).

In recent years, PSO for FS in gene expression data has gained much popularity. This is because the data used include only a small number of informative genes out of a large pool of genes in thousands, which may have a role in the occurrence of cancers. [30] suggested an approach called PSODT that combined PSO with C4.5 decision tree classifier. Seven popular classification algorithms were employed on 11 gene expression cancer datasets that ranged from 83 to 12,601 features to evaluate the integrated PSO's performance. Experimental results revealed that the PSODT algorithm outperformed when compared with other algorithms in terms of classification accuracy. However, the proposed algorithm, in terms of computation, involves high costs than others, especially for large datasets. An improved BPSO is proposed in [31] to select a small subset of informative genes, which would prove significant for the cancer classification. To increase the probability of unselected features, a speed concept was employed to update the positions of the particles. Through this method, much smaller feature subsets could be identified by PSO when compared with other compared methods. However, the method suffers from few drawbacks such as results with high standard deviations for some data sets, low accuracy in few cases and the lack of statistical significance test for the results.

Of late, some researchers have carried out studies on the hybridisation of filter methods for PSO [10, 31-34]. These hybrid methods offer low computational costs and provide enhanced efficiency and performance as the wrapper component used in the hybrid method focuses on feature dependencies. Thus, the aim of selecting a hybrid method is to reap benefits of both the techniques to improve classification accuracy. For instance, [10] presented a hybrid feature selection method (HFSM) that offered advantages of both wrapper and filter approaches. Eight UCI data sets were considered for the experiment, which demonstrated that the performance of the proposed method was greater than the existing methods when it comes to calculation efficiency, classification accuracy, and the number of relevant features. [32] presented yet another filter feature selection approach that considered BPSO and information theory. Four benchmark datasets were selected from the UCI machine learning repository to perform analysis of the proposed method. The results of this analysis revealed that the number of features could be reduced substantially with the proposed algorithms, which also allowed achieving similar, or in some cases, even higher classification accuracy. [34] proposed a feature selection scheme that was based on five filter measures with PSO to combine the strengths of both wrappers and filters approaches. The results of the experiment demonstrated that the yield of the proposed method was slightly better than the PSO-based filter algorithm in terms of performance. However, the proposed algorithm’s performance was not compared with the results of the wrapper approach, which typically yielded better results than the filter approach.

For feature selection methods, various modified versions of BPSO have been suggested in the literature; however, most of them still face problems such as low convergence speed and getting stuck in local optima. Despite the main element of BPSO being the transfer function, the literature does not have sufficient investigation information to address its significance. Therefore, this paper focuses on the investigation of the proposed v-shaped transfer function and its impact on the BPSO for FS [20]. It also evaluates the effectiveness of the function to suggest approaches to improve both exploitation and exploration of the algorithm. Although different optimisation problems employ this function [35,36], its use in the FS task is however limited. Also, the process consumes a long time to create feature subsets, which leads to high computational costs. Thus, the hybridisation approach applied to filter methods with PSO can decrease feature dimensionality and accelerate the classification processes and feature subset generation. The proposed approach is explained briefly with details in the following section.

3 The Proposed Approach

PSO is considered a powerful technique when it comes to handling reduction of dimensionality and FS. This is due to its inherent features such as simplicity, lower computational cost and involvement of fewer parameters. However, a major disadvantage of the PSO approach is its low resistance to getting trapped in the local minima. Also, researchers consider high dimensional data as a complex and challenging task as the data contains many redundant and irrelevant features. These may deteriorate the machine learning algorithms' performance by causing overfitting, reduced accuracy as well as longer time consuming. Therefore, to address the mentioned issues, an EBPSO approach for FS is suggested. Also, we introduced a hybrid FS approach that could combine the filter FS method with

3.1 EBPSO for Feature Selection

Stagnation is a common primary problem that may occur when employing the swarm intelligence technique in which the swarm agents may fall in the local minima. In such a case, the agents may face prematurely converged issues, and their chances of moving into the search space become even lower. Hence, finding the optimum solution for swarm algorithms is extremely difficult. In this paper, BPSO was provided with the capability to overcome this situation by enhancing the transfer function.

In BPSO, the velocities of individuals that are closed to the local minima should approximately be close to zero. Thus, according to Eqs. (3) and (4), the probability for shifting the position of individuals becomes zero. This will result in low power of exploitation and exploration, which in turn may bring in the stagnation situation. In situation where PSO is trapped in the local minima, incrementation of the probability of individual's movement in the search space can be performed by replacing Eqs. (3) and (4) with the help of new updating position and new transfer function. This function enhances the power of exploitation and exploration. Therefore, by strengthening the exploitation and exploration, the BPSO could avoid local optimum and also attain a good convergence to the global optimum. The following subsections discuss the building of a distinct table, the formulation of fitness functions, the proposed transfer function, and finally the recommended EBPSO algorithm.

3.1.1 Building a Distinction Table

How to get the first population remains the foremost important question here, which relies on the type of problem. In the case of a travelling salesman problem, any permutation of city numbers as an individual produces the first population, whereas it is much more different in the current case. A distinction table is created in an explicit way, where a number of individuals are randomly selected from the created distinction table. Also, we engage the distinction table with the fitness score computation for each individual as illustrated in Eq. (6). The distinction table, a binary matrix, consists of dimensions $\frac{(T^2-T)}{2} \times N$, where N represents the number of features and T represents the objects. An element $b(i,(k,j))$ of the matrix corresponding to the pair of objects (x_k, x_j) and attribute a_i is given by:

$$b((k,j),i)=\begin{cases}1, & if\ a_i(x_k) \neq a_i(x_j).\\ 0, & if\ a_i(x_k) = a_i(x_j).\end{cases} \tag{5}$$

The presence of "1" signifies that the attribute a_i has the capability to discern (or distinguish) between the pair of objects (x_k,x_j). As the aim here is to determine the reducts that have the potential to discern between objects, even if they belong to different classes, we may assume the distinction table to have *N* number of columns and rows that correspond to exclusively to those object pairs (x_k,x_j) such that $d(x_k) = d(x_j)$.

(i) Since object pairs that are from the same class are not fit to represent a row of the distinction table, the size gets reduced substantially, which results in a decrease in computational cost.
(ii) In the matrix, the entries '1' signify the attributes of interest that may aid in arriving at a classification decision.

According to [12], the following must be satisfied when the distinction table is created: Let us assume that the number of objects initially in the two classes to be T_1 and T_2. Then, in the *d*-distinction table, the number of rows becomes $(T_1 * T_2) < \frac{T*(T-1)}{2}$, in which $T_1 + T_2 = T$. This results in the reduction of the complexity of fitness computation to $O(N * T_1 * T_2)$. Once the distinction table is built, a certain number of strings (each row is a string) can be selected from the table to develop the first population of the BPSO. It should be noted that the proposed EBPSO approach is not just limited to solving binary class problems; it can also be extended to solve multi-class problems. However, the present work concentrates on both multi-class and binary problems that employ benchmark datasets.

3.1.2 Fitness Function

The following objective function can be employed to perform feature selection. It consists of two sub-functions: F_1 and F_2. F_1 allows identifying the number of selected features (i.e., number of 1's), while F_2 helps to decide whether the feature can distinguish between the object's pairs. The fitness function introduced is as follows:

$$F = \alpha_1 F_1(\vec{v}) + (1-\alpha_1)F_2(\vec{v}) \tag{6}$$

Where the first sub-function $F_1(\vec{v}) = \frac{N-O_{\vec{v}}}{N}$, and the second sub-function $F_2(\vec{v}) = \frac{R_{\vec{v}}}{T_1 * T_2}$ are described by the condition $0 < \alpha_1 < 1$. Here, $\vec{v}$ represents the reduct candidate, $O_{\vec{v}}$ shows the number of 1's in $\vec{v}$, T_1 and T_2 represent the number of objects in the two classes, and $R_{\vec{v}}$ represents the number of object combinations that can be distinguished by rows in the distinction table $\vec{v}$. The fitness function F_1 provides the candidate credit for those that have less number of features in $\vec{v}$, while F_2 helps on deciding the extent of discerning level by the candidates for given object pairs in the distinction table.

3.1.3 Proposed Transfer Function

Flipping the positions of the particles is achieved by a transfer function. The function outlines the probability to map the elements of the position vector from 0 to 1 and vice versa. $s(v_{id})$ defines the probability for the change of a binary bit to 1, while the probability for that bit to change to 0 is defined by $1 - s(v_{id})$. In general, larger velocities have very high chances of flipping, while those with smaller values of velocity have lesser chances. In the BPSO approach, the particle changes its bits randomly even if the particle's velocity is small. This indicates a stronger exploration characteristic, even though a good solution has been reached. This seems unreasonable, as the value of velocity close to 0 indicates that the binary bits and the convergence of algorithm should get changed with a lower exploration. So, the BPSO approach has a few drawbacks when operating the binary positions. Also, it cannot remember the already obtained high-quality positions. It is challenging for the BPSO to cover the entire optimal solution as randomness keeps increasing with the iteration process. Furthermore, the BPSO suffers from getting trapped in the local minima [2, 20]. To address the above mentioned problems, a new type of transfer function consisting of different position updating formulas could be employed [20]. According to [20], selecting a transfer function requires considering of a few concepts as mentioned in the following:

- The transfer function's range needs to be limited corresponding to the interval [0,1] as it gives the probability when a particle would likely change its position.
- A transfer function should allow achieving a high probability for changing X_{id}^{t} for a large absolute velocity value. The particles showing large absolute values are probably not the best solution, and so the positions need to be switched in the next iteration.
- A transfer function should give a small probability for changing X_{id}^{t} for a small absolute velocity value.
- The return value of a transfer function needs to be increased with the rise in the velocity. The particles that are moving away from the best solution need to have a higher probability of changing their position vectors so that they can later resume their earlier positions.
- The return value of a transfer function should reduce with decline in velocity.

Following these concepts will guarantee an optimum transfer function that can map continuous search space to binary search space, while maintaining similar concepts of the search (Gbest and Pbest in PSO) for a specific evolutionary algorithm. As will be shown later, all the above mentioned concepts are fulfilled by the proposed transfer function. These were named v-shaped transfer function because of the shapes of these functions [20]. In this study, a v-shaped transfer function is employed to improve BPSO's performance in terms of convergence rate and avoiding local minima. In addition, the v-shaped transfer function [20] presents substantial advantages [20, 37] when compared with s-shaped transfer function. Here, we employed a v-shaped transfer function as suggested by Mirjalili and Lewis [20] to map a continuous velocity with a binary position. It is designed as shown in Eq. (7) :

$$T(v_{id}^{t+1}) = \left|\frac{2}{\pi}\ \arctan\left(\frac{\pi}{2}\, v_{id}^{t+1}\right)\right| \tag{7}$$

where

$$v_{id}^{t+1} = w * V_{id}^{t} + c1 * r1 * HD[P_{id}(t), X_{id}(t)] + c2 * r2 * HD[P_{gd}(t), X_{id}(t)] \tag{8}$$

where HD represents the hamming distance between two binary vectors. As the above transfer function (7) is distinctly different from the original one (4), the applied rules for position updating would also get changed. Here, Eq. (9) is employed for updating the position values that are based on Eq. (7):

$$X_{id}^{t+1} = \begin{cases} (X_{id}^{t})^{-1} & , \ if\ rand() < T(v_{id}^{t+1}) \\ X_{id}^{t} & , \ if\ rand() \geq T(v_{id}^{t+1}) \end{cases} \tag{9}$$

Where the position and velocity of particle i at iteration t given the $d-th$ dimension are represented by X_{id}^{t} and v_{id}^{t}, respectively, and $(X_{id}^{t})^{-1}$ represents the complement of X_{id}^{t}. This method is beneficial mainly because its transfer function does not impose the particles to assume values of 0 or 1. According to (9), a decreasing velocity encourages the particle position to stay in its current location and try a nearby search (exploitation). In contrast, a larger velocity corresponds to a higher probability of changing position. In other words, larger velocities are indicative of a particle being far from the best solution. Therefore, the particle has to switch bits to its complement in the position vector with the goal of finding promising regions within the search space (exploration). Here, the memory of the particle plays an important role since it lessens the randomness of the binary positions during iteration and improves the algorithm's convergence speed. Furthermore, the new position derived from Eq. (9) maintains the swarm diversity and reduces the probability of the EBPSO getting stuck in local optimum.

Overall, the BPSO's shortcomings can be addressed by the enhancement of BPSO resulting from the new updating position and transfer function. The EBPSO can potentially be used to address different optimisation problems. Some of its advantages can be summarised as follows: 1) the presented transfer function greatly enhances the convergence speed. 2) The new position updating formula avoids the problem of local minima and increases the diversity of the population.

3.1.4 Proposed Algorithm

The following steps outline the major phases of the proposed feature selection algorithm using EBPSO:

Step 1) From a dataset of either binary or multi classes, a d-distinction table is produced.
Step 2) A population of size N called particles are randomly initialized.
Step 3) The fitness value F for each particle is calculated using Eq. (6).
Step 4) The p_{best} (i.e. p_{id}) and g_{best} (i.e. p_{gd}) at every iteration are updated based on the fitness function.
Step 5) Based on the number of different bits between two particles, the hamming distance is computed.
Step 6) The velocity of the particle is updated using Eq. (8)
Step 7) Given the new transfer function using Eq. (7), another particle update is created.
Step 8) The position of the particle is also updated using Eqs. (9).
Step 9) Steps 2 to 8 are then performed repeatedly for the finite number of iterations.

3.2 Proposed Hybrid FS Approach Based on EBPSO

The main purpose of a hybrid approach is to find subsets of informative features using different feature selection methods (filter and wrapper). Compared to individual algorithms, these methods have smaller number of features and better classification performance. Moreover, more time is consumed if the FS only uses the wrapper approach in the microarray gene expression data. On the other hand, finding gene subsets which have interaction could be impossible if only filter approach is used. To achieve the benefit of the filter and wrapper, a hybrid approach combining both feature selection methods is proposed. This could improve the classification performance and lessen the computing time. There are two stages in the building of the proposed hybrid approach, with each stage containing an approach, as seen in fig 1. In the first stage, a specific number of features are selected based on their ranking using information gain (IG) that uses a certain statistical criterion. In the second stage, an optimised feature subset is generated. It uses the feature subsets selected in the first stage as EBPSO inputs.

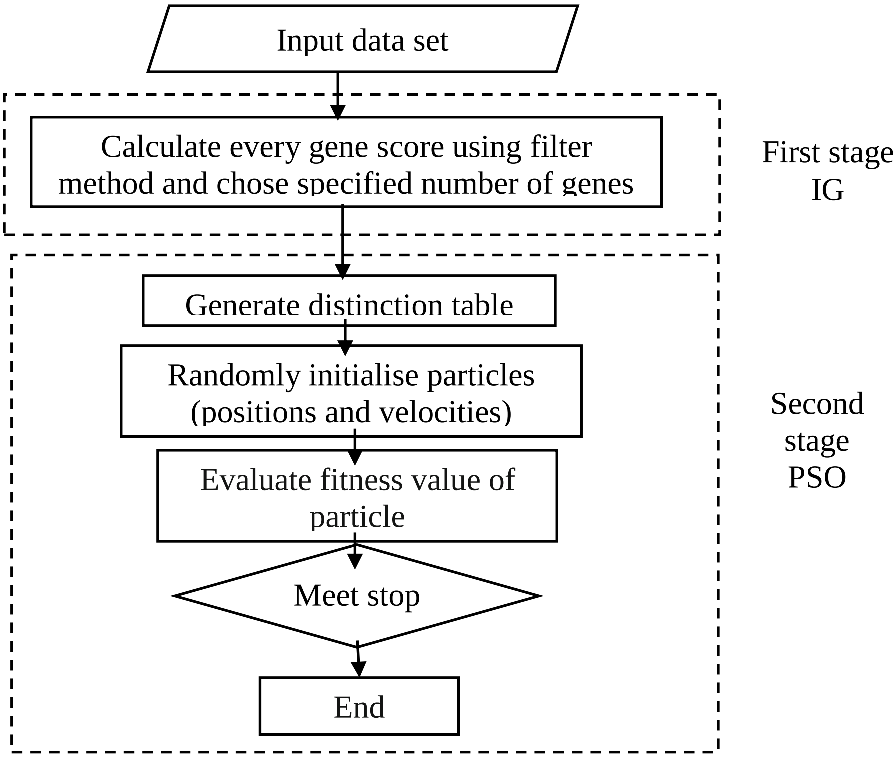

Fig.1. Hybrid filter and wrapper feature selection approach

3.2.1 Stage 1: Filter Method

This stage's major aim is to use EBPSO to speed up the feature selection process and perform an initial reduction for gene expression data. This paper used information gain (IG) that had been selected for its effectiveness and efficiency. The relevance of that feature X is determined by IG [41] of an attribute *X* with respect to the class attribute *Y*. This is achieved when its information gain is evaluated with respect to the class y. The following formula denotes this:

$$IG\ (Y\backslash X) = H(Y) - H(Y\backslash X) \tag{10}$$

$$H(Y) = -\sum_{i=1}^{k} P\ (y_i)\log_2(P(y_i)) \tag{11}$$

$$H(Y\backslash X) = -\sum_{j=1}^{n} P(x_j)\sum_{i=1}^{k} P(y_i\backslash x_j)\log_2(P(y_i\backslash x_j)) \tag{12}$$

In this stage, the feature's importance is used to reduce the original genes from N to M. The filter method is used to calculate the feature's importance. In other words, based on their importance, the original genes are ranked in decreasing order. The top genes or the most important genes are used in the next stage while the less important genes are ignored. The most important genes are the ones that use EBPSO to search for a better solution

3.2.2 Stage 2: Wrapper Method

In this stage, EBPSO is applied to the feature subsets obtained in the first stage. Dimension reduction is performed as well. Application of the EBPSO is carried out to the most important, high-ranking features. Consequently, there is an improvement of the classification performance and a reduction of the computational cost when EBPSO is used. Section 3.1 provides the details of the EBPSO.

4 Experiments

4.1 Data Sets and Experimental Setup

Java language was used to implement the proposed EBPSO. All the experiments were performed on a PC with the Windows operating system and an Intel(R) Core^{TM} i3 processor, 1.70 GHz, 4GB of RAM. The quality of the computational results could be affected by the parameter setting of the algorithm. Two accelerator coefficient parameters $c1= c2= 2$ were set up. Generally, the velocity should be limited in order to achieve a good convergence rate. Therefore, for this study's BPSO, the maximum velocity is set to 4. Inertia weight (w) is one of the PSO's most important parameters. This can control the exploitation and exploration ability of the algorithm. The inertia weight (w) was set at a value of 0.9. The swarm size is set to a value of 35. Different population sizes like 10, 30, 50 and 100 were also tried out. The maximum number of generations is at 50. It was observed that no further improvement is achieved when the number of iterations goes past 50.

In this study, the proposed EBPSO and hybrid feature selection approach were evaluated using six binary and multiclass microarray datasets in all the experiments conducted. The datasets include lymphoma, colon cancer, leukaemia, leukaemia2, MLL, and prostate tumour. There was a random partitioning of every dataset into training and testing sets of equal sizes (50% of the data are utilised for training while the other 50% is used for testing). The selected features subsets' quality is tested with the use of two classifiers SVM and KNN due to their promising classification performance and simplicity, respectively. Descriptions of all the datasets are presented in Table 1.

Table 1: Description of the six cancer datasets

Dataset	#features	#classes	#samples	source
Colon	2000	2	62	http://eps.upo.es/bigs/datasets.html
Lymphoma	4026	9	96	http://eps.upo.es/bigs/datasets.html
Leukemia	7129	2	72	http://eps.upo.es/bigs/datasets.html
Prostate	10509	2	102	http://www.gems-system.org/
Leukemia2	11225	3	72	http://www.gems-system.org/
MLL	12582	3	72	http://csse.szu.edu.cn/staff/zhuzx/

4.2 Experimental Results

The results are reported based on a comparison between BPSO and EBPSO. This is done in order to determine if the proposed EBPSO algorithm performs better than BPSO based on classification performance. First, different subsets of features are chosen by using the application of the EBPSO to the six datasets. Next, 10-fold cross validation accuracy on features subsets generated from EBPSO is measured using SVM. 10-fold cross validation needs to partition the training set into k smaller sets of (approximately) equal sizes. Model training uses $k-1$ of the folds to serve as the training data. The rest of the data is utilised as test set for the calculation of the classification accuracy. This study used 10-fold cross validation (i.e. K=10) for every experimental run. 9 folds are used for training (90%) while one-fold is used for testing (10%). Here, validation of the 10-fold cross is conducted on the six datasets with feature subsets that are produced using our proposed algorithm and the SVM classifier. For every dataset, experiments are independently performed 10 times. Afterwards, the results from the 10 independent runs are averaged. Table 2 shows that except for the colon and prostate datasets, all runs have achieved over 95% classification accuracies on all the datasets. This signifies that EBPSO was able to use high-dimensional data to efficiently select and produce a near-optimal gene subset.

Table 2: Results for every run using EBPSO on colon, leukaemia, lymphoma, prostate tumour, leukaemia2 and MLL

Run#	Colon	Lymphoma	Leukaemia	prostate	leukaemia2	MLL
	ACC	ACC	ACC	ACC	ACC	ACC
1	79.03	95.83	97.22	91.18	98.61	97.22
2	83.87	95.83	98.61	92.16	98.61	97.22
3	80.65	95.83	98.61	90.2	97.22	95.83
4	85.48	96.88	98.61	91.18	97.22	97.22
5	82.26	95.83	98.61	92.16	94.44	97.22
6	82.26	96.88	98.61	90.2	95.83	97.22
7	82.26	95.83	98.61	92.16	95.83	97.22
8	82.26	95.83	98.61	91.18	97.22	97.22
9	83.87	95.83	98.61	92.16	95.83	97.22
10	79.03	95.83	98.61	92.16	97.22	97.22
Avg.	82.10	96.04	98.47	91.47	96.80	97.08
S.D.	2.0748	0.4427	0.4396	0.8068	1.3187	0.4396

Note: ACC: Accuracy rate (%), S.D.: Standard deviation, Avg.: Average accuracy rate, R#: A run number

Comparisons

Table 3 shows the comparison between the performance of EBPSO and BPSO [12] using support vector machine (SVM) classifier and 10-fold cross validation. The table shows that in most cases, the proposed EBPSO produced better results than BPSO in terms of the variance, the classification accuracy, and the standard deviation. Compared to the BPSO algorithm, the proposed EBPSO algorithm produces significantly larger improvements in five of the six benchmark datasets. This could be attributed to the utilisation of v-shaped transfer and how its mechanism for updating position could speed up the convergence rate and improve the capacity of BPSO to avoid local minima. In BPSO, Eq. (4) refers to the mechanism used for updating position when the s-shaped function is used where the particle is assigned a value of 0 or 1. This may prove to be ineffective because if a variable of particle is given the value of 1 and the velocity is high, there should be a high chance for the variable to become 0 instead. However, Eq. (4) requires this variable to stay in 1 despite the very high probability. As a result, exploration of PSO algorithm may deteriorate. Contrary to the existing methods that employed the above mechanism based on s-shaped function, our proposed approach's v-shaped function Eq. (9) uses a different mechanism for updating position and is therefore more superior compared to the s-shaped function. The first condition was built using the complement of variable as the basis. This guarantees and reinforces changing the particle position in a manner that is proportional to its velocity. Exploration of the search space is therefore promoted. Furthermore, the second condition requires the particle to stay in its current location and conduct the search nearby. In other words, the particle variables tend to remain in their positions more often. This in turn promotes exploitation. Therefore, this function is beneficial mainly because it does not force the particles to assume values of 0 or 1. The results from table 3 signify that the performance of BPSO is enhanced by the v-shaped transfer function in terms of variance, classification accuracy, and standard deviation. Thus, it suggests that using the v-shaped transfer function for the proposed EBPSO allows it to find a global solution possessing a good convergence rate for high dimensional data. In summary, the EBPSO results prove that the method for updating position of the v-shaped transfer function is useful for binary PSO in terms of convergence speed, avoiding local minima, and accuracy results.

Moreover, a quartile-based heuristic technique is used in BPSO to pre-process the initial datasets. This is done to reduce the dataset's dimensionality before the PSO is applied. Our proposed method has been applied to full-featured datasets. However, table 3 shows that EBPSO produces better results. This result can be attributed to the fact that EBPSO is able to obtain some important features that may have been lost when applying the pre-processing technique in BPSO.

Furthermore, a T-test at the 95% confidence level is performed in order to see if there is a significant difference between the EBPSO and BPSO, where the significance level was set as 0.05. In other words, it is not enough to compare algorithms based on the variance, classification accuracy, and standard deviation values. A statistical test is necessary to prove that a proposed algorithm gives a significant improvement over the existing one. The results showed that the EBPSO achieved better results than BPSO in four out of six datasets. These datasets include colon, Lymphoma, Leukemia and Leukemia2. Hence it can be concluded that the use of a V-shaped function and its mechanism for updating position in the EBPSO leads to superior performance when compared to the BPSO. This is due to the ability of the EBPSO to produce good-quality solutions, where it encourages particles to stay in their current positions when their velocity values are low or switches to their complements when the velocity values are high.

Table 3: Comparison of average classification accuracy, standard deviation and variance on EBPSO and BPSO

Dataset	BPSO			EBPSO		
	ACC	S.D.	Variance	ACC	S.D.	Variance
Colon	90.28	0.3871	0.1498	82.10	2.0748	4.3048
Lymphoma	93.25	0.4369	0.1909	96.04	0.4427	0.1959
Leukemia	91.25	0.514	0.2645	98.47	0.4396	0.1932
Prostate	95.55	0.8791	0.7728	96.80	1.3187	1.7390
Leukemia2	91.38	1.0121	1.0243	91.47	0.8068	0.6509
MLL	96.66	0.9719	0.9446	97.08	0.4396	0.1932

Note: ACC: Average accuracy rate (%), S.D. : Standard deviation

Using a powerful k-nearest neighbour (k-NN) classifier [42] given different values of k (1,3 and 5) on test data, a comparative study between the proposed hybrid approach IG- EBPSO and BPSO [12] was conducted. Table 4 shows that in terms of the number of selected genes and classification accuracy, the proposed hybrid IG-EBPSO showed better performance than BPSO. In BPSO, the selected feature subsets ranged from 10 to 20, while our proposed approach's feature subsets were in the range of 2 to 17. These values were indeed quite smaller. Furthermore, in comparison to the BPSO, our proposed approach was able to obtain higher classification accuracy in most cases.

In the case of the datasets for colon and lymphoma, the proposed algorithm was able to recognise correctly more than 96% with a feature subset size 17 or less. However, in BPSO, classification is 100% correct for the same two datasets, when the feature subset size 20 or less. BPSO using feature subsets of size 12, 16 and 15 could achieve a classification score of 91.67%, 82.35% and 72.22% for leukaemia2, prostate tumour, and MLL, the respectively. On the other hand, with feature subsets of size 8, 9 and 3, the proposed algorithm IG- EBPSO could achieve a classification score of 100%, 94.12% and 100%, respectively which are better than those obtained by BPSO. For the leukaemia data, IG-EBPSO and BPSO had the same accuracy. However, when the attribute size is lowered, it comes down to two by IG-EBPSO. This finding may be attributed to the fact that IG-EBPSO is capable of effectively eliminating irrelevant and redundant genes because of the combination of the IG filter and the BPSO wrapper. As a result, a small set of reliable genes possessing high classification accuracy is identified. Moreover, it can be contended that as a pre-processing technique for gene evaluation, the IG is more efficient compared to the quartile based fast heuristic technique utilised in BPSO. This is due to the fact that IG aims to reduce uncertainty or gain a feature's information for classification. More information obtained means higher classification accuracy because the new instances' predicated classes are more likely to match with their true classes.

Table 4: Comparative study of k-NN classifier using hybrid IG-EBPSO and BPSO for colon, leukaemia, lymphoma, prostate tumour, leukaemia2 and MLL

Dataset	Feature subset size	Used method	*k*-NN classification (%) on test set		
			K=1	*K=2*	*K=3*
Colon	≤2	**Proposed**	80.65	**96.77**	**93.55**
	≤10	BPSO	100	93.55	90.33
Lymphoma	≤17	**Proposed**	97.92	**95.83**	**97.92**
	≤20	BPSO	100	93.75	95.80
Leukemia	≤8	**Proposed**	97.22	**100**	**100**
	≤10	BPSO	100	94.74	94.74
Leukaemia2	≤8	**Proposed**	**100**	**100**	**100**
	≤12	BPSO	91.67	91.67	75
Prostate	≤9	**Proposed**	**90.2**	**94.12**	**94.12**
	≤16	BPSO	78.43	82.35	78.43
MLL	≤3	**Proposed**	**91.67**	**100**	**100**
	≤15	BPSO	61.11	55.56	72.22

Table 5 shows another comparison among previous related works that utilised the GA-based methods in their proposed methods [12]. A powerful *k*-nearest neighbour (*k*-NN) classifier [48] was utilised for the test data, given different values of *k* (1, 3, and 5). The table shows that compared to the previous works, the classification accuracies of our work were higher than others for all the data sets. Moreover, our study was also able to achieve a smaller number of selected features on most cases. It was observed that in five out of six datasets, the IG-EBPSO obtained less than 10 features. Moreover, all of the solutions provided by IG-EBPSO possess a classification rate that is larger than 94%.

In GA, classification scores of 71.0%, 93.76%, and 73.53% were obtained for colon, lymphoma and leukaemia data using feature subsets of size 15, 18 and 19, respectively. Using the same three datasets, the proposed IG-EBPSO achieved improved accuracies of 96.77%, 97.92%, and 100% using 2, 17 and 8 feature subsets. The same authors proposed a feature selection method that is based on the multi-objective (NSGA-II). A K-NN classifier was used and they reported scores of 90.3%, 95.80% and 94.1% using 10, 2 and 5 feature subsets for the respective data sets. On the other hand, the proposed approach achieved classification accuracies of 96.77%, 97.92% and 100% when feature subsets of 2, 17, and 8 were used. It was also observed that for lymphoma and leukaemia datasets, the NSG-II was able to produce subsets of 2 and 5 features with approximate accuracies of 93.8% and 94.1%. However, for the same two datasets, the proposed approach achieved accuracies of 97.92% and 97.22%.

For colon cancer, our proposed algorithm exhibited better performance in terms of both the number of selected features and the classification accuracy. This is true except for K =1, where the NSGA-II achieved good classification accuracy. This finding can be attributed to the v-shaped transfer function’s significant role and mechanism for updating position. This encourages the particles to either switch to their complement under high velocity so that they get to the promising regions of the search space on its way to the best particle (exploration) or stay in current location and try to search nearby (exploitation) when the velocity is low. Moreover, it is believed that the IG-EBPSO based on IG aims to gain information for a feature and reduce the level of uncertainty for classification. More information corresponds to higher classification accuracy of the model because the new instances’ predicated classes have a higher likelihood of corresponding to their true classes.

All the results presented above signify that our proposed IG-EBPSO algorithm had a better and comparative performance compared to the other approaches based on the six bench mark high dimensional datasets.

Table 5: Comparative study of k-NN classifier using hybrid IG-EBPSO, NSGA-II and GA for colon, leukaemia, lymphoma, prostate tumour, leukaemia2 and MLL

Dataset	Feature subset size	Used method	*k*-NN classification (%) on test set		
			K=1	*K=2*	*K=3*
Colon	≤2	**Proposed**	80.65	**96.77**	**93.55**
	≤10	NSGA-II [12]	**90.3**	90.3	87.1
	≤15	GA [12]	71.0	58.10	48.40
Lymphoma	≤17	**Proposed**	**97.92**	**95.83**	**97.92**
	≤2	NSGA-II [12]	93.8	95.80	95.80
	≤18	GA [12]	89.59	89.59	93.76
Leukemia	≤8	**Proposed**	**97.22**	**100**	**100**
	≤5	NSGA-II [12]	94.1	91.2	91.2
	≤19	GA [12]	73.50	73.53	60.77
Leukaemia2	≤8	**Proposed**	**100**	**100**	**100**
	-	NSGA-II [12]	-	-	-
	-	GA [12]	-	-	-
Prostate	≤9	**Proposed**	**90.2**	**94.12**	**94.12**
	-	NSGA-II [12]	-	-	-
	-	GA [12]	-	-	-
MLL	≤3	**Proposed**	**91.67**	**100**	**100**
	-	NSGA-II [12]	-	-	-
	-	GA [12]	-	-	-

Note**:** ‘- ‘means that a result is not reported in the related previous work.

5 Conclusion

In this paper, we have offered an improved BPSO algorithm for feature selection in high dimensional gene expression data. The experimental outcomes of EBPSO were better than BPSO and earlier GA-based methods with regards to classification precision and the number of chosen genes. The outcomes have been tested utilising six benchmark cancer datasets: lymphoma, colon, leukaemia, leukaemia2, MLL and prostate tumour. The v-shaped transfer function with its own approach of updating positions vector was utilised. The outcomes indicate that the introduced v-shaped transfer function can considerably enhance the original BPSO's performance with regards to evading local minima, convergence rate and precision of outcomes. Furthermore, a hybrid FS approach which included a filter method with EBPSO was recommended in this paper. This approach was introduced to decrease the data dimensionality and create a feature subset which would offer improved classification performance. The attained outcomes of the hybrid FS approach exhibited better performance in decreasing the dimensionality and gaining higher classification precision in the majority of the cases, in comparison to other methods. Also, the selection of the attributes which occur most frequently among the reducts might turn out to be significant for biologists as the features in the core (the reducts intersection) can be the interesting genes accountable for a specific medical condition.

Going forward, it would be interesting to examine the effect of the v-shaped transfer functions on other binary algorithms, including the binary grey wolf algorithm and binary cuckoo search algorithm. Furthermore, other mathematical functions can be explored, which could be utilised as new transfer functions for BPSO.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Higher Education, Malaysia (FRGS/1/2016/ICT02/UKM/01/2).

References

[1] Kennedy, J., & Eberhart. R. (1995). Particle Swarm Optimization. *In: IEEE International Conference on Neural Networks* (vol. 4, pp.1942–1948).IEEE

[2] Kennedy, J., & Eberhart. R. (1997). A discrete binary version of the particle swarm algorithm. *In: IEEE International Conference on Systems, Man, and Cybernetics* (pp. 4104–4108).IEEE

[3] Kennedy, J., & Eberhart. R. (2001). Swarm Intelligence. *Morgan Kaufmann*, San Francisco, USA.

[4] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research* (vol.3, pp.1157– 1182)

[5] Tan, F., Fu, X., Zhang, Y., & Bourgeois, A. G. (2008). A genetic algorithm based method for feature subset selection. *Soft Computing* (vol.12, pp.111–120)

[6] Hunt, R., Neshatian, A., & Zhang, M. (2012). A genetic programming approach to hyper-heuristic feature selection. *In: Simulated Evolution and Learning* (vol 7673, pp.320– 330).

[7] Kashef, S., & Nezamabadi-pour, H. (2013) .A new feature selection algorithm based on binary ant colony optimization. *In Information and Knowledge Technology* (pp. 50–54).

[8] Tang, E.K., Suganthan, P., & Yao X. (2005). Feature selection for microarray data using least squares SVM and particle swarm optimization. *In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology* (CIBCB) (pp.1–8,).

[9] Kamala F. R., & Thangaiah, P. (2016). A Proposed Two Phase Hybrid Feature Selection Method using Backward Elimination and PSO. *International Journal of Applied Engineering Research* (pp.77-83).

[10] Yasodha, P., & Ananthanarayanan, N. R. (2015). Analysing Big Data to Build Knowledge Based System for Early Detection of Ovarian Cancer. *Indian Journal of Science and Technology* (pp. 1-7).

[11] Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis* (vol. 1, pp.131– 156).

[12] Banka, H., & Dara, S. (2015). A Hamming Distance based Binary Particle Swarm Optimization (HDBPSO) Algorithm for High Dimensional Feature Selection, Classification and Validation. *Pattern Recognition Letters* (pp. 94-100)

[13] Liu, Y., Tang, F., & Zeng, Z. (2015). Feature selection based on dependency margin. *In: IEEE Transactions on Cybernetics* (vol.45, pp.1209– 1221).

[14] Liu, H., Motoda, H., Setiono, R., & Zhao, Z. (2010). Feature selection: An ever evolving frontier in data mining. *In Feature Selection for Data Mining* (*FSDM*) (vol. 10, pp.4–13)

[15] Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* (vol. 17, pp.491–502)

[16] Unler, A., & Murat, A. (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research* (vol. 206, 528–539).

[17] Liu, Y., Wang, G., Chen, H., & Dong, H. (2011). An improved particle swarm optimization for feature selection, *Journal of Bionic Engineering* (vol. 8, pp.191–200)

[18] Chuang, L.Y., Li, J.C., & Yang, C.H. (2008). Chaotic maps in binary particle swarm optimization for feature selection. *In: IEEE Conference on Soft Computing in Industrial Applications*. (pp. 107–112).

[19] Tran, B., Xue, B., & Zhang, M. (2014). Improved PSO for Feature Selection on High-Dimensional Datasets. *Proceedings of the 10th International Conference on Simulated Evolution and Learning. Lecture Notes in Computer Science* (pp. 503-515).

[20] Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation* (vol. 9, pp. 1-14).

[21] Xue, B., Zhang, M., Browne, W., and Yao, X. (2014). A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transaction on Evolutionary Computation (*pp.606-626).

[22] Banerjee, M., Mitra, S., & Banka, H. (2007). Evolutionary rough feature selection in gene expression data. *IEEE Transactions on Systems, Man, and Cybernetics* (pp. 622–632).

[23] Dang, V.Q., Lam, C.P., & Lee, C.S. (2001). Incorporating genetic algorithm into rough feature selection for high dimensional biomedical data. *Paper presented at the 3rd International Symposium on IT in Medicine and Education (ITME)* (pp. 283-287).

[24] Luque-Baena, R., Urda, D., Subirats, J., Franco, L., & Jerez, J. (2013). Analysis of Cancer Microarray Data using Constructive Neural Networks and Genetic Algorithms. *Proceedings of the IWBBIO, International Work-Conference on Bioinformatics and Biomedical Engineering* (pp. 55-64).

[25] Moon, S. N. & Bawane, N. (2015). Optimal Feature Selection by Genetic Algorithm for Classification Using Neural Network. *International Research Journal of Engineering and Technology (IRJET)* (pp. 1561-1566).

[26] Li, Y., Wang, G., Chen, H., Shi, L., & Qin, L. (2013). An ant colony optimization based dimension reduction method for high-dimensional datasets. *Journal of Bionic Engineering*, *10*(2), 231-241.

[27] Tabakhi, S., & Moradi, P. (2015). Relevance–redundancy feature selection based on ant colony optimization. *Pattern recognition*, *48*(9), 2798-2811. J. Quinlan, Induction of decision trees, Machine Learning, pp. 81–106, 1986.

[28] Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, *18*, 261-276.

[29] Chen, K. H., Wang, K. J., Tsai, M. L., Wang, K. M., Adrian, A. M., Cheng, W. C., & Chang, K. S. (2014). Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm. *BMC bioinformatics*, *15*(1), 49.

[30] Mohamad, M. S., Omatu, S., Deris, S., & Yoshioka, M. (2011). A modified binary particle swarm optimization for selecting the small subset of informative genes from gene expression data. *IEEE Transactions on Information Technology in Biomedicine*, *15*(6), 813-822.

[31] Sahu, B., & Mishra, D. (2012). A novel feature selection algorithm using particle swarm optimization for cancer microarray data. *Procedia Engineering*, *38*, 27-31..

[32] Cervante, L., Xue, B., Zhang, M., & Shang, L. (2012, June). Binary particle swarm optimisation for feature selection: A filter based approach. In *Evolutionary Computation (CEC), 2012 IEEE Congress on* (pp. 1-8). IEEE.

[33] Esseghir, M. A., Goncalves, G., & Slimani, Y. (2010, September). Adaptive Particle Swarm Optimizer for Feature Selection. In *IDEAL* (Vol. 6283, pp. 226-233).

[34] Wei, S., Zhang, J., & Sun, T. (2015). Nodes selection mechanism based on modified binary particle swarm optimization algorithm.

[35] Yang, J., Zhang, H., Ling, Y., Pan, C., & Sun, W. (2014). Task allocation for wireless sensor network using modified binary particle swarm optimization. *IEEE Sensors Journal*, *14*(3), 882-892.

[36] Saremi, S., Mirjalili, S., & Lewis, A. (2015). How important is a transfer function in discrete heuristic algorithms. *Neural Computing and Applications*, *26*(3), 625-640.

[37] Xue, B., Zhang, M., & Browne, W. N. (2012, July). Multi-objective particle swarm optimisation (PSO) for feature selection. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation* (pp. 81-88). ACM.

[38] Sousa, T., Silva, A., & Neves, A. (2004). Particle swarm based data mining algorithms for classification tasks. *Parallel Computing*, *30*(5), 767-783.

[39] Parsopoulos, K. E., & Vrahatis, M. N. (2002). Particle swarm optimization method for constrained optimization problems. *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies*, *76*(1), 214-220.

[40] T. Mitchell, Machine Learning. McGraw-Hill, 1997

[41] Tou, J. T., & Gonzalez, R. C. (1974). Pattern recognition principles. Reading, MA: Addison Weley Publishing Company (pp. 75–109).