

Research Article

A Holistic Performance Comparison for Lung Cancer Classification Using Swarm Intelligence Techniques

Sunil Kumar Prabhakar ¹, Harikumar Rajaguru,² and Dong-Ok Won ³

¹Department of Artificial Intelligence, Korea University, Anam-dong Seongbuk-gu, Seoul 02841, Republic of Korea

²Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamangalam 638401, India

³Department of Artificial Intelligence Convergence, Hallym University, Chuncheon, Gangwon 24252, Republic of Korea

Correspondence should be addressed to Dong-Ok Won; dongok.won@hallym.ac.kr

Received 14 December 2020; Accepted 17 July 2021; Published 31 July 2021

Academic Editor: Saverio Maietta

Copyright © 2021 Sunil Kumar Prabhakar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the field of bioinformatics, feature selection in classification of cancer is a primary area of research and utilized to select the most informative genes from thousands of genes in the microarray. Microarray data is generally noisy, is highly redundant, and has an extremely asymmetric dimensionality, as the majority of the genes present here are believed to be uninformative. The paper adopts a methodology of classification of high dimensional lung cancer microarray data utilizing feature selection and optimization techniques. The methodology is divided into two stages; firstly, the ranking of each gene is done based on the standard gene selection techniques like Information Gain, Relief-F test, Chi-square statistic, and T -statistic test. As a result, the gathering of top scored genes is assimilated, and a new feature subset is obtained. In the second stage, the new feature subset is further optimized by using swarm intelligence techniques like Grasshopper Optimization (GO), Moth Flame Optimization (MFO), Bacterial Foraging Optimization (BFO), Krill Herd Optimization (KHO), and Artificial Fish Swarm Optimization (AFSO), and finally, an optimized subset is utilized. The selected genes are used for classification, and the classifiers used here are Naïve Bayesian Classifier (NBC), Decision Trees (DT), Support Vector Machines (SVM), and K -Nearest Neighbour (KNN). The best results are shown when Relief-F test is computed with AFSO and classified with Decision Trees classifier for hundred genes, and the highest classification accuracy of 99.10% is obtained.

1. Introduction

The number of patients who are diagnosed with cancer is steadily increasing in a rapid manner [1]. With the help of biopsies, image processing techniques, and blood analysis, the diagnosis of cancer is done presently. When damaged cells are excessively accumulated in human body, it leads to cancer [2]. For every patient, the behavior of cancer differs, and by examining deeply into the origin of it, it can be well understood. The cancer originates in the cells and to every individual, the structure of the cell is quite unique. Therefore, to cure cancer permanently, there is not a single specific vaccine available [3]. Understanding the relation between the gene and its products

is a contribution to the genetic approach to cancer diagnosis, so that the identification of biomarker genes for targeting drug therapies can be understood well [4]. With this approach, the effects of genes on some cell signaling pathways can be well understood [5]. The information about active levels of a gene is provided by the gene expression. For gene expression, one of the widely used measurement technique is microarray [6]. In the cancer diagnosis and cancer classification types, the gene expression values obtained by microarrays can be utilized. In many studies, the microarray datasets are employed for these purposes. For the selection of biomarker gene subsets, various feature selection algorithms are employed [7]. To this microarray dataset, statistic machine learning techniques are

implemented with or without feature selection [8]. Biomarker genes are to classify cancer types, with a highest classification accuracy being identified by the biomarker genes.

In recent years, a new dimension to cancer research has been encompassed by the advent of microarray technology. For the classification, analysis, diagnosis, and treatment of cancer, a proficient method has been emerged by the microarray-based gene expression data [9]. Thousands of features termed as genes are found in the microarray gene expression dataset. Such data has records or instances from a few patients only and due to this limited availability of samples in comparison to larger number of genes, it is termed as curse of dimensionality problem [10]. Due to this, (a) the training time during the classification process is increased, and (b) there is a reduction in classification accuracy [11]. Therefore, the extraction of useful information from the dataset is hindered due to these challenging issues. So, the number of genes has to be reduced, and then, the highly informative genes should be selected, so that classification accuracy is increased, and it is a significant step in the microarray data analysis [12]. Feature selection/gene selection in the microarray data classification aim is to select a small subset of features from the original huge feature space [13]. By removing redundant and irrelevant features, feature selection can be done, so that the classification accuracy is increased, and the classification time is reduced. The feature selection technique proposed in the literature includes hybrid method, embedded technique, filter, and wrapper methods [14]. In this study, the primary aim is to classify and select the optimal gene subsets for lung cancer. Then, feature selection is implemented along with optimization techniques and finally classified.

Some of the prominent works in the lung cancer classification using microarray gene analysis are explained as follows. For the molecular classification of lung cancer, a cross study comparison of gene expression study was done by Parmigiani et al. [15]. Using the significance analysis of Microarray-Gene set reduction algorithm, the classification of non-small cell lung cancer was performed by Zhang et al. [16]. For multiclass classification of lung cancer, an adaptive multinomial regression with overlapping groups is performed by Li et al. [17]. The lung cancer prediction from microarray data by gene expression programming was done by Azzawi et al. [18]. A support vector machine-based classification method for lung cancer gene expression data base analysis was done by Guan et al. [19]. Some progresses in the techniques and integrated analysis related to the image processing techniques and the development of advanced devices for tissue engineering approach as a potential solution to treat lung diseases too have been discussed in the literature [20, 21].

As far as the microarray gene selection techniques using optimization and classification are concerned, self-organizing maps [22], ensemble classification techniques [23], Taguchi chaotic binary Particle Swarm Optimization (PSO) [24], adaptive wrapper approach combined with SVM [25], kernel based methods [26], pattern classification methods [27], Convolution Neural Networks (CNN) [28], fuzzy approaches [29], Analysis of Variance (ANOVA), and K -

Nearest Neighbour (KNN) [30] were proposed in the literature. Using ant colony optimization, a hybrid gene selection approach was proposed by Sharbaf et al. [31]. For the cancer classification data on gene expression data, PSO and DT classifiers were implemented by Chen et al. [32]. For gene selections, the various techniques reported in literature are utilizing multiobjective algorithms [33], a hybrid binary Imperialist Competition Algorithm (ICA), and tabu search approach [34], a binary differential evolution algorithm [35], a simplified swarm optimization using a Social Spider Optimization (SSO) algorithm [36], Artificial Bee Colony (ABC) [37], Binary PSO [38], novel rule-based algorithm [39], and Shuffled Leap Frog Algorithm (SLFA) [40], and it has been well explored. However, in this paper, other suitable swarm intelligence techniques have been explored and analyzed comprehensively. The organization of the paper is as follows. In Section 2, the materials and methods followed by the gene selection techniques are explained. In Section 3, the optimization techniques for gene selection are explained, and in Section 4, the classification techniques are explained followed by the results and discussion in Section 5 and conclusion in Section 6.

2. Materials and Methods

For the lung cancer classification, a lung Harvard 2 dataset was utilized, which is publicly available online [41]. The dataset has 181 samples with 150 Adenocarcinoma (ADCA) and 31 Malignant Pleural Mesothelioma (MPM). The dataset is tabulated in Table 1.

The pictorial representation of the work is shown in Figure 1.

2.1. Gene Selection Techniques. The gene selection techniques utilized in this paper are Information Gain, Relief-F, Chi-square statistic, and T-statistic. The discretization of the attribute values is done before using chi-square, information gain, and other feature selection methods. The main intention of utilizing the gene selection techniques is to select the most important genes from 12,533 genes. Here, in our work, we have selected 1000 important genes after the gene selection process through the following techniques.

2.1.1. Information Gain. It is used generally as an attribute selection criteria while dealing with decision trees; hence, it is used as a gene selection technique too [7]. Assume the class set $S = \{S_x\}$, where $x = 1, 2, \dots, l$. For every feature Y_j , the Information Gain is expressed as

$$\text{InfoGain}(Y_j) = H(S) - H\left(\frac{S}{Y_j}\right), \quad (1)$$

where $H(S) = -\sum_{s \in S} p(s) \log_2 p(s)$ and

$$H\left(\frac{S}{Y_j}\right) = -\sum_{y \in Y_j} p(y) \sum_{s \in S} p\left(\frac{s}{y}\right) \log_2 p\left(\frac{s}{y}\right). \quad (2)$$

Only for discrete features, Information Gain is used widely, and therefore prior to computing Information Gain,

TABLE 1: Dataset details.

Dataset	Number of genes	Class 1 (ADCA)	Class 2 (MPM)	Total samples
Lung Harvard 2	12533	150	31	181

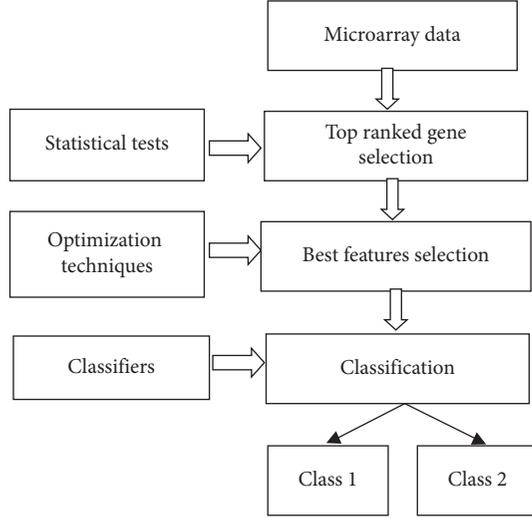


FIGURE 1: Pictorial representation of the work.

the discretization of numeric features should be done. Depending on the large values of information gain, the selection of features are done.

2.1.2. Relief-F. For dealing with multiclass, noisy, and incomplete datasets, Relief-F is introduced, and it is an extension of Relief algorithm [7]. To each feature, a relevance weight is assigned. The selection of a random sample instance I is done from n sample instances. Based on the basic differences between the selected instance I and its neighboring instance represented as Q and termed as hit and different class termed as nearest miss represented by $N(S)$, the updating of the relevance features is done. The features that discriminate the instance from various neighbors of the surrounding classes are given more weight. By analyzing the average contribution of neighboring nearest misses $N(S)$, the updating of the weights is done. The prior probability of each class is considered by the average contribution. The updating of the weight of j^{th} feature Y_j is as follows:

$$w_j = w_j - \frac{\Psi(Y_j, I, Q)}{n} + \sum_{S=S_i} \frac{P(S) * \Psi(Y_j, I, N(s))}{n}, \quad (3)$$

where the distance between sample instances (I) and the nearest hit (Q) or nearest miss $N(S)$ is calculated by the function $\Psi(Y_j, I, Q)$.

2.1.3. Chi-Square Statistic. With respect to the classes, for each feature, the value of χ^2 statistic is computed [7]. Before computing χ^2 statistic, the discretization of the numeric attributes is done. For every feature, Y_j , χ^2 statistic is computed as

$$\chi^2 = \sum_{y \in Y_j} \sum_{s \in S} \frac{(n_{(y \in Y_j, s \in S)} - e_{(y \in Y_j, s \in S)})^2}{e_{(y \in Y_j, s \in S)}}, \quad (4)$$

where $n_{(y \in Y_j, s \in S)}$ represent the number of samples in Y_j for class s whose value is y . The definition of expected frequency is expressed as

$$e_{(y \in Y_j, s \in S)} = \frac{n_{y \in Y_j} \times n_{s \in S}}{n}, \quad (5)$$

where the number of samples in Y_j with value y is denoted by $n_{y \in Y_j}$; $n_{s \in S}$ indicates the number of samples of class s . The total number of samples is expressed by n . Based on the sorted value of χ^2 statistic, the selection of features is done.

2.1.4. T-Statistic. This is a famous gene selection technique and quite popular in two-class problems [7]. Every sample can be classified into either class S_1 or class S_2 . For every feature Y_j , the computation of t-statistic is expressed as

$$t(Y_j) = \frac{|\mu_{j1} - \mu_{j2}|}{\sqrt{(\sigma_{j1}^2/n_1) + (\sigma_{j2}^2/n_2)}}, \quad (6)$$

where μ_{jk} indicates the mean of the j^{th} feature for class S_k . The “ k ” indicates the class index, i.e., $k = 1$ or $k = 2$.

Once the t-statistic value for each feature is computed, then it is sorted out in a descending order, so that the important features can be selected.

3. Optimization Techniques

The shortlisted 1000 genes will undergo again a secondary feature selection methodology to select the best 50 genes, 100 genes, and 200 genes by means of utilization of optimization techniques. The second level feature selection is done using the five optimization algorithms as follows.

3.1. Grasshopper Optimization Algorithm. In many engineering optimization problems, this algorithm is widely used. Based on the biggest swarms of all creatures, one of the recently proposed naturally inspired algorithms is GO [42]. Severe damage to the crops is caused by the herbivores grasshopper. The grasshopper has a swarming behavior, and it depends on both adults and nymphs. Soft plants and succulents are fed by the nymph, which rolls on the ground continuously. In search of food, the adult grasshopper can jump to a very high extent, and so, it will have a very large area to explore. The observation of both types of movement such as slow movement and abrupt movement has been achieved, which indicates that exploitation and exploration are possible. For the grasshopper, the swarming behavior is represented mathematically as

$$Q_j = A_j + F_j + B_j, \quad (7)$$

where Q_j represents the position of the j^{th} grasshopper, A_j represents the social interaction, F_j represents the gravity force in the j^{th} grasshopper, and B_j represents the wind advection. The representation of social interaction A_j is given as

$$A_j = \sum_{k=1, k \neq j}^N a(d_{jk}) \hat{d}_{jk}, \quad (8)$$

where $d_{jk} = |q_k - q_j|$ represents the distance between the j^{th} and k^{th} grasshopper and $\hat{d}_{jk} = (q_k - q_j) / (d_{jk})$ represents a unit vector from the j^{th} grasshopper to the k^{th} grasshopper. The social forces are expressed by the function “ a ” and are expressed as

$$a(s) = ge^{-(s/l)} - e^{-s}, \quad (9)$$

where the intensity of the attraction is represented as g and the attractive length scale is expressed by l . In terms of social interaction, three types of regions are created by the grasshoppers in search of food, that is, attraction zone, comfort zone, and repulsion region. Strong forces cannot be applied by the function “ a ” when the distance is large between grasshoppers. To resolve this, the F component in (7) is expressed as

$$F_j = -f\hat{e}_f, \quad (10)$$

where the f represents the gravitational constant and \hat{e}_f indicates a unity vector progressing towards the Earth center. The computation of B component is as follows:

$$B_j = v\hat{e}_u, \quad (11)$$

where v represents the constant drift and \hat{e}_u represents a unity vector in the wind direction. If the values of a , F and B are substituted in (7), then

$$Q_j = \sum_{K=1, K \neq j}^N a(|q_k - q_j|) \frac{q_k - q_j}{d_{jk}} - f\hat{e}_f + v\hat{e}_u, \quad (12)$$

where $a(s)$ is given by (9) and the number of grasshoppers is represented by N . To solve optimization problems, a revised version of this formula is used as

$$Q_j^d = c \left(\sum_{k=1, k \neq j}^N c \frac{v_{bd} - l_{bd}}{2} a(|q_k^d - q_j^d|) \frac{q_k - q_j}{d_{jk}} \right) + \hat{T}_d, \quad (13)$$

where v_{bd} represents the upper bound and l_{bd} represents the lower bound in the D^{th} dimension. In the target, the value of the D^{th} dimension is represented by \hat{T}_d . To shrink the three worms, the decreasing coefficient is “ c .” Only towards a target the wind direction is progressed always. While the food is a searched form, adults start jumping in the air, and nymphs move on rolling in the ground creating both cases of exploration and exploitation. By reducing the parameter c in the below equation, one can balance both these two in

proportion to the total number of iterations. Its computation is done as

$$c = c_{\max} - i \left[\frac{c_{\max} - c_{\min}}{I} \right], \quad (14)$$

where the maximum value is represented as c_{\max} , minimum value is represented as c_{\min} , i denotes the current iteration, and I represents the maximum number of iterations.

3.2. Moth Flame Optimization Algorithm. Based on the simulation of moth behavior for this special movement method during nighttime, Moth flame optimization algorithm was developed [43]. For the purpose of navigation or movement, a mechanism termed as transverse orientation is utilized. By maintaining a standard angle with reference to the moon, moth flies, which is a very effective methodology for travelling by distances in a straight path, as the distance between the moon and the moth is very far away. This kind of methodology is adopted, so that moth flies along a very straight path at nighttime. It is a general observation that the moths fly around the lights in a spiral manner. The artificial lights can easily trick the moths to exhibit such behavior. As the light lies with close proximity to the moon, a spiral fly of moths is caused due to the maintenance of a similar angle to the light source. In this algorithm, the representation of the set of moths is done as a matrix A . For the storage of all the corresponding fitness values, there is an array OA for all the moths. In this algorithm, the second key component is the flames. Now, again, a matrix B similar to the moth matrix is considered. For the storage of all the corresponding fitness values, there is an array OB for all the flames. The global optimal of the optimization problem is approximated by the MFO algorithm by a three-tuple process as follows:

$$\text{MFO} = (C, D, K). \quad (15)$$

A random population of moths with its corresponding fitness value is denoted by a function C . In this function, the methodical model is expressed as

$$C: \phi \longrightarrow \{A, OA\}. \quad (16)$$

The movement of the moths around the search space is determined by the function D which is the primary function. The matrix of A is received by this function and eventually returns the updated one as

$$D: A \longrightarrow A. \quad (17)$$

The K function remains true if termination criterion is satisfied and false if the termination criterion is not satisfied.

$$K: A \longrightarrow \{\text{true}, \text{false}\}, \quad (18)$$

With C , D , and K , the general framework of the MFO is expressed as follows:

```

A = C ();
while K (A) is equal to false
A = D (A);
end

```

Until the K function returns true, the D function is run iteratively after the initialization. To simulate the moth behavior mathematically, the updating of the position of every moth is updated with respect to a flame using the following equation:

$$A_c = F(A_c, B_g), \quad (19)$$

where A_c indicates the c^{th} moth, B_g specifies the g^{th} flame, and F represents the spiral function.

Subject to the following conditions, the utilization of any type of spiral can be done using the three conditions as follows:

- (1) The initial point of the spiral should begin from the moth
- (2) The final point of the spiral should be the flame position
- (3) The range of spiral fluctuation should not exceed the search space

For the MFO algorithm, the logarithmic spiral is defined as

$$F(A_c, B_g) = J_c \cdot e^{hk} \cdot \cos(2\pi k) + B_g, \quad (20)$$

where J_c specifies the distance of the c^{th} moth for the g^{th} flame, h denotes a constant for defining the shape of a logarithmic spiral, and k is a random number in $[-1, 1]$.

The computation of J is done as follows:

$$J_c = |B_g - A_c|, \quad (21)$$

where A_c indicates the c^{th} moth, B_g specifies the g^{th} flame, and J_c specifies the distance of the c^{th} moth for the g^{th} flame.

The spiral flying path of the moth is expressed by (20). From this equation, with respect to a flame, the next position of a moth is explained. In the spiral equation, the k parameters defer the next position of the moth with reference to its proximity or closeness to the flame. While the position is updated, it only regains a moth to progress towards a flame; thereby, it may be trapped in local optima fastly. Each moth is obliged to update its position using only one of the flames to prevent such situations. The position updating of moths with respect to “ n ” various locations in the search space may sometimes denote the exploitation of most promising solutions.

$$\text{flame number} = \text{round}\left[N - I * \frac{N - 1}{K}\right], \quad (22)$$

where I denotes the current number of iterations, N denotes the maximum number of flames, and K specifies the maximum number of iterations. To balance the exploration and exploitation of the search space, there is a gradual decrease in the number of flames. The general steps of the D function are described in Algorithm 1.

As projected in the algorithm, unless the K function returns true, the D function is executed. Once the D function is terminated, the best moth returns, as it is shown as the best attained optimum approximation value.

3.3. Bacterial Foraging Optimization Algorithm. The three main mechanisms are present in the classical BFO, that is, chemotaxis process, reproduction process, and elimination-dispersal process [44].

3.3.1. Chemotaxis Process. Here, a tumble indicates a unit walk with random direction, and a run indicates a unit walk with the similar direction in the last step. Assuming $\theta^a(b, c, d)$ indicates the bacterium at b^{th} chemotactic, c^{th} reproductive, and d^{th} elimination-dispersal method. $R(a)$ is considered as the run-length unit parameter is the chemotactic step size during every tumble or run. The movement of the a^{th} bacterium in every computational chemotactic step is expressed as

$$\theta^a(b + 1, c, d) = \theta^a(b, c, d) + R(a) \frac{\Delta(a)}{\sqrt{\Delta^T(a)\Delta(a)}}, \quad (23)$$

where $\Delta(a)$ represents the direction vector of the b^{th} chemotactic step. $\Delta(a)$ is the same as the final chemotactic step if the bacterial movement is run; or else $\Delta(a)$ becomes a random vector, where specific elements lie in the range of $[-1, 1]$. A step fitness indicated as $B(a, b, c, d)$ is evaluated with the activity of both run or tumble assumed and considered at each step during the chemotaxis process.

3.3.2. Reproduction Process. During its lifetime, the sum of the step fitness is calculated as the health status of each bacterium as $\sum_{b=1}^{N_r} B(a, b, c, d)$, where N_r represents the maximum step in a chemotaxis process. Based on the health status, the sorting of the bacteria is done in a reverse order. Only the first half of population lives/survives in the reproductive step. The living bacterium divides into two identical ones, and they are kept in the same places, and so the population of bacteria keeps constant.

3.3.3. Elimination and Dispersal Process. A basis for local search is provided by the chemotaxis, and the convergence is sped up by the reproduction process. Using the classical BFO, this situation has been simulated to a large extent. For searching of global optima, only chemotaxis and reproduction are not enough. Around the local optima, the bacteria may get stuck and to eliminate the accidents of being trapped into local optima easily and gradually, the diversity of the BFO changes. Only after a certain number of reproductive processes, the dispersion event happens. Then, based on a probability P_{pr} , some bacteria are chosen to be killed and shifted to another position within a particular environment. The step by step procedure is explained in Algorithm 2.

3.4. Krill Herd Optimization Algorithm. Based on the simulation of the herding of krill swarms, a famous meta-heuristic algorithm for solving optimal problems is KH optimization algorithm [45]. The herding of the skill swarms is usually in response to a certain environmental and

biological process. In a 2D space, the time-dependent position of an individual krill is decided by 3 primary actions, that is,

- (i) Movement which influences or influenced by other krill individuals.
- (ii) Foraging actions
- (iii) Random diffusion

In a d-dimensional decision space, the following Lagrangian model is adopted by the KH algorithm as

$$\frac{dZ_j}{dt} = M_j + G_j + D_j, \quad (24)$$

where M_j is the motion led by other krill individuals, G_j is the foraging motion, and D_j is the physical diffusion of the j^{th} krill individual.

The krill individuals affecting the other movement are represented by the direction of motion α_j and it is computed by the target swarm density, a local swarm density, and a repulsive swarm density. The movement for a krill individual is defined as follows:

$$M_j^{\text{new}} = M^{\text{max}} \alpha_j + v_m M_j^{\text{old}}. \quad (25)$$

The maximum induced speed is represented by M^{max} , the inertia weight of the motion induced in $[0, 1]$ is represented as v_m and the latest motion induced is represented by M_j^{old} .

With the help of two main components, the estimation of foraging motion is done. The first one is the food location, and the second one is the basic knowledge about the food location. The motion is approximately formulated for the j^{th} krill individual as follows:

$$G_j = W_g \beta_j + v_g G_j^{\text{old}}, \quad (26)$$

where

$$\beta_j = \beta_j^{\text{food}} + \beta_j^{\text{best}}, \quad (27)$$

where the foraging speed is represented by W_g . The inertia weight of the foraging motion between 0 and 1 is represented as v_g and G_j^{old} is the last foraging action.

A random process is modelled to the random diffusion of the krill individuals. In terms of both a random directional vector and maximum diffusion speed, the description of the motion can be done. It is represented as follows:

$$D_j = D^{\text{max}} \delta, \quad (28)$$

where the maximum diffusion speed is D^{max} , the random directional vector is δ and its arrays are random values in the range of $[-1, 1]$. Utilizing various motion parameters during the time and based on the above-mentioned movements, using the following equation, the position vector of a krill individual from the time interval t to $t + \Delta t$ is given as

$$Z_j(t + \Delta t) = Z_j(t) + \Delta t \frac{dZ_j}{dt}, \quad (29)$$

Δt is regarded as the most important term and based on the specific type of optimization, the parameters can be fine-

tuned. The scalar factor of the speed vector is assumed because of Δt parameter.

3.5. Artificial Fish Swarm Optimization Algorithm. It is a famous Swarm Intelligence technique, which is helping to solve the optimization problem by utilizing the behavior of artificial fishes like imitating swarming process, chasing process, and preying behaviors [46]. Assume A_p is the current position of one artificial fish and A_w is the viewpoint of artificial fish at one specific moment. The visual scope of every individual is expressed as Vis ; therefore, within Vis of A_p be the fishes A_y and A_z . The largest step of artificial fish is assumed as step and the congestion factor of the fish swarm is expressed as δ . The food concentration factor is highly proportional to the fitness function $f(A)$. In the fish swarm, the behavior patterns are expressed as follows:

3.5.1. Swarming Behavior. If $f(A_s) > f(A_p)$, then A_s is the central point inside the Vis of the point A_p and so the execution of swarming behavior is done easily. Assume A_s as A_w and so, the fish at A_p will progress towards the point A_s .

3.5.2. Chasing Behavior. The point (expressed by A_{max}), which has the best objective function value, is present inside the visual satisfying the criterion $f(A_{\text{max}}) > f(A_p)$ and if there is less crowd in the visual of A_p , then the execution of chasing behavior is done. Consider A_{max} as A_w and so, the fish at A_p will progress towards the point A_{max} .

3.5.3. Preying Behavior. Under the following situations, preying behavior is tried.

- (a) $f(A_q) > f(A_p)$, $f(A_{\text{max}}) < f(A_p)$ and there is less or no crowd in the Vis
- (b) Alternatively, if the visual is crowded, then the random selection of a point A_q inside the visual of A_p is done.

The preying behavior is executed if $f(A_q) > f(A_p)$. Assume A_q as A_w and so, the fish at A_p will progress towards the point A_q . Otherwise, with its visual limit, it will move a step in a random fashion.

In each iteration, the best solution obtained is termed as "board." The search process can be terminated after the specified iterations, and the result present on the board is considered as the final solution. The position updating for the artificial preying fishes is formulated as

$$A_{\text{next}} = A_p + \text{rand} \cdot \frac{\text{step} \times (A_q - A_p)}{\text{norm}(A_q - A_p)}, \quad (30)$$

where the next position of artificial fish is termed as A_{next} . The current position of the artificial fish is expressed as A_p and the position having a better objective function value is A_q . The random number is expressed as rand and it is in the range of $[-1, 1]$. Between the two position vectors, the

distance is expressed as $\text{norm}(A_q - A_p)$. The position updating for the artificial swarm fishes can be done as

$$A_{\text{next}} = A_p + \text{rand} \cdot \frac{\text{step} \times (A_s - A_p)}{\text{norm}(A_s - A_p)} \quad (31)$$

The position updating for the artificial chasing fishes is formulated as

$$A_{\text{next}} = A_p + \text{rand} \cdot \frac{\text{step} \times (A_{\text{max}} - A_p)}{\text{norm}(A_{\text{max}} - A_p)} \quad (32)$$

4. Classification Procedures

The optimized values or the best gene values obtained after the second level optimization techniques are classified using NBC, Decision trees, SVM, and KNN algorithms. The Performance Analysis of Classifiers in terms of Classification accuracies with GO, MHO, BFO, KHO, and AFSO for different gene selection techniques using 50–200 selected genes is done here.

4.1. Naïve Bayesian Classifier. It is a famous probabilistic algorithm, where, given the class, the feature values based on Bayes rule are conditionally independent [47]. If a new sample observation is given, the assignment of the classifier to the class having the maximum conditional probability estimate is done.

4.2. Decision Tree. A famous rule-based classifier is DT, where leaf nodes represent classification outcomes and non-leaf nodes represent selected attributes [48]. A classification rule is reflected by the path from the root to a leaf node. The J4.8 algorithm is used here.

4.3. Support Vector Machine. The SVM analyzes the input data as two unique sets of vector in a p-dimensional space initially [49]. Then, in that space, a separate hyperplane is constructed, so that the margin is maximized between the two data sets. The SVM is utilized with SVM Polynomial kernels for training purposes.

4.4. KNN Algorithm. One of the famous instance-based classifiers is KNN [50]. The class label of a new testing sample is decided by the classifier. It is done by considering the majority of classes of its K closest neighbors dependent on their Euclidean distance. Here, the value of K is assigned to be 4.

5. Results and Discussion

It is classified with a 10-fold cross-validation method, and its performance is shown in the tables. The mathematical formulae for computing the Performance Index (PI), Sensitivity, Specificity, and Accuracy are mentioned in literature and using the same, the values are computed and exhibited.

PC is Perfect Classification; MC is Missed Classification; and FA is False Alarm in the following expressions.

The sensitivity is expressed as

$$\text{Sensitivity} = \frac{\text{PC}}{\text{PC} + \text{FA}} \times 100. \quad (33)$$

Specificity is expressed as

$$\text{Specificity} = \frac{\text{PC}}{\text{PC} + \text{MC}} \times 100. \quad (34)$$

Accuracy is expressed as

$$\text{Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2}. \quad (35)$$

Performance Index (PI) is expressed as

$$\text{PI} = \left(\frac{\text{PC} - \text{MC} - \text{FA}}{\text{PC}} \right) \times 100. \quad (36)$$

Table 2 shows the performance analysis of classifiers for classification accuracy parameter with GO method for different gene selection techniques. As indicated in Table 2 that SVM classifier with 100 selected genes in Relief F test method and NBC with information gain method for 100 genes attained higher accuracy of 98.96%. The lower accuracy of 76% is thrown out by KNN classifier in all three statistical methods.

Table 3 indicates the performance analysis of classifiers for classification accuracy with MFO method for different gene selection techniques. As shown in Table 3, DT classifier with 50 selected genes in Relief F test method reached higher accuracy of 98.012%. The lower accuracy of 78.125% is depicted by SVM classifier with 100 genes selected in relief F test method. The lower accuracy of SVM is due to the presence of outlier in the gene samples.

Table 4 demonstrates the performance analysis of classifiers for classification accuracy with BFO method for different gene selection techniques. From Table 4, it is identified that DT classifier with 50 selected genes in Chi-square test method reached higher accuracy of 98.56%. The lower accuracy of 82.24% is shown by SVM classifier with 100 genes selected in information gain method. Across the gene samples, all the classifiers performed well in this BFO method.

Table 5 reveals the performance analysis of classifiers for classification accuracy with KHO method for different gene selection techniques. Table 5 shows that SVM classifier with 50 selected genes in Relief F test method reached higher accuracy of 98.38%, as the number of selected genes increased gradually and given to SVM classifiers, which reported lower accuracy of 77.47% with 200 genes selected in Relief F test method.

Table 6 reports the performance analysis of classifiers for classification accuracy with AFSO method for different gene selection techniques. As indicated in Table 6, DT classifier with 100 selected genes in Relief F test method reached the highest accuracy of 99.10%. The NBC classifier is settled at the lower accuracy of 77.08% with 200 selected genes for Relief F test method.

```

Initialize the position of moths
While (Iterations ≤ Max_Iterations)
Update flame number using (22)
OA = Fitness function (A);
if iteration == 1
B = sort (A)
OB = sort(OA);
else
B = sort (Ak - 1, Ak)
OB = sort (Ak-1, Ak);
end
for c = 1: n
for g = 1: j
Update r and k
Calculate J using (21) with respect to the corresponding moth
Update A(c, g) using (19) and (20) with respect to the corresponding moth
end
end
end

```

ALGORITHM 1: D function execution and termination.

Step 1: Parameter Initialization

s : dimension of the search space

N : number of bacteria

S_r : chemotactic steps

S_n : swim steps

S_{re} : reproductive steps

S_{pr} : elimination and dispersal steps

P_{pr} : probability of elimination

$R(a)$: run-length unit

Step 2: The elimination-dispersal loop is expressed as $d = d + 1$

Step 3: Reproducing loop: $c = c + 1$

Step 4: Chemotaxis loop: $b = b + 1$

(a) For $a = 1, 2, \dots, N$, a chemostatic step for bacteria “ a ” is considered.

(b) Fitness function computation, $F(a, b, c, d)$

(c) Assume $F_{\text{last}} = F(a, b, c, d)$ to save this value so that a better value is found out via run process

(d) A random vector $\Delta(a) \in R^s$ is generated in the tumble process. Here each element is represented as $\Delta_m(a)$, $m = 1, 2, \dots, N$ with a random number $[-1, 1]$

(e) For bacteria “ a ”, in the direction of the tumble, the movement is progressed so that it results in a step of size $R(a)$

(f) Calculate $F(a, b + 1, c, d)$ with $\theta^a(b + 1, c, d)$

(g) Swim phase:

(i) Let $q = 0$ (swim length counter)

(ii) While $q < S_n$;

(a) Assume $q = q + 1$

(b) If $F(a, b + 1, c, d) < F_{\text{last}}$, let $F_{\text{last}} = F(a, b + 1, c, d)$. Another step of size $R(a)$ in this similar direction is considered and then the newly generated $\theta^a(b + 1, c, d)$ is utilized to compute the new $F(a, b + 1, c, d)$

(c) Else let $q = S_n$

(h) Proceed to next bacterium ($a + 1$): if $a \neq N$, go to (b) to process the next bacteria

Step 5: If $b < S_r$, proceed to step 3. In such a case, the chemotaxis is continued as the bacteria life is not over.

Step 6: Reproduction

(a) For the given “ c ” and “ d ” and for every $a = 1, 2, \dots, N$ assume $B_{\text{health}}^a = \sum_{b=1}^{S_r+1} F(a, b, c, d)$ be bacteria health. In an ascending order, the bacterium is sorted out (B_{health})

(b) The N_z bacteria with the highest B_{health} values expire and the other N_z bacteria comprising of best values split.

Step 7: If $c < S_{re}$, go to step 2. The next generation in the chemotactic loop is initiated as the number of specified reproduction steps is not reached.

Step 8: Elimination dispersal: For $a = 1, 2, \dots, N$, with a probability P_{pr} , the bacteria are eliminated and dispersed which results in keeping the total number of bacteria in the entire population as a constant. On the optimization domain, simply disperse it to a random location, so that the bacterium is eliminated. If $d < S_{pr}$, then proceed to step 2 of this procedure or else end the procedure.

ALGORITHM 2: BFO.

TABLE 2: Performance analysis of classifiers in terms of classification accuracies (%) with Grasshopper optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	82.29	98.96	76	78.12	83.59	83.59	76	83.59	77.08	76	76	77.08
Relief-F test	85.93	91.67	83.59	83.59	78.12	89.6	78.12	98.96	95.83	87.5	78.12	77.08
Chi-square test	97.91	77.08	83.59	95.83	93.75	77.08	82.29	83.59	91.67	76	77.08	83.59
T statistic test	77.08	82.29	82.29	97.91	95.83	78.12	83.59	93.75	95.83	77.08	76	85.93

TABLE 3: Performance analysis of classifiers in terms of classification accuracies (%) with Moth flame optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	96.34	89.6	93.75	85.93	83.59	93.26	91.67	97.91	89.6	89.6	83.59	95.83
Relief-F test	93.75	82.29	91.67	98.01	97.91	86.19	85.93	78.12	95.83	82.29	97.91	89.6
Chi-square test	93.75	93.75	90.74	91.67	85.93	84.10	97.91	86.40	85.93	85.93	97.91	91.67
T statistic test	85.93	91.67	95.83	85.93	93.75	84.75	96.15	85.93	91.67	85.93	85.93	97.91

TABLE 4: Performance analysis of classifiers in terms of classification accuracies (%) with Bacterial foraging optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	97.91	96.85	91.64	95.83	85.93	85.93	85.02	82.24	89.92	85.93	87.12	93.75
Relief-F test	97.91	93.75	89.6	89.6	91.67	83.59	97.91	83.81	87.30	95.83	97.91	97.91
Chi-square test	93.75	89.6	91.67	98.56	90.53	97.33	97.72	85.93	97.91	97.91	95.83	95.83
T statistic test	91.67	93.75	85.93	84.41	95.83	98.04	86.71	97.91	84.95	95.83	93.75	89.6

TABLE 5: Performance analysis of classifiers in terms of classification accuracies (%) with Krill Herd optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	97.91	91.67	81.57	86.58	89.6	96.76	91.89	93.75	97.91	92.05	91.40	96.92
Relief-F test	89.6	97.91	83.59	82.29	85.93	87.37	98.38	79.21	77.47	83.46	90.97	84.25
Chi-square test	93.75	93.27	81.72	97.12	82.59	92.12	88.55	78.99	98.69	87.88	78.61	88.91
T statistic test	95.83	97.91	91.67	95.83	96.93	95.83	94.72	85.06	96.67	86.73	88.03	98.96

Table 7 signifies the performance analysis of classifiers for classification PI parameter with GO method for different gene selection techniques. As shown in Table 7, SVM classifier with 100 selected genes in Relief F test method and NBC with information gain for 100 genes attained higher PI of 97.87%. The lower PI of 7.69% is indicated by KNN classifier in all three statistical methods.

Table 8 demonstrates the performance analysis of classifiers for classification PI with MFO method for different gene selection techniques. As shown in Table 8, DT classifier with 50 selected genes in Relief F test method reached higher PI of 95.85%. The lower PI of 22.22% is indicated by SVM classifier with 100 genes selected in Relief F test method. The lower PI of SVM is due to the presence of outlier genes in the samples.

Table 9 represents the performance analysis of classifiers PI with BFO method for different gene selection techniques. From Table 9, it is known that DT classifier with 50 selected genes in Chi-square test method reached higher PI of

97.009%. The lower PI of 45.09% is indicated by SVM classifier with 100 genes selected in information gain method. Across the gene samples, all the classifiers performed well in this BFO.

Table 10 exposes the performance analysis of classifiers for classification PI with KHO method for different gene selection techniques. Table 10 reported that SVM classifier with 50 selected genes in Relief F test method reached higher PI of 96.68%, as the number of selected genes increased gradually and given to SVM classifiers, which reported lower PI of 17.93% with 200 genes selected in Relief-F test method.

Table 11 details the performance analysis of classifiers for classification PI with AFSO method for different gene selection techniques. As indicated in Table 11, DT classifier with 100 selected genes in Relief F test method reached the highest PI of 98.16%. The NBC classifier depicts lower PI of 15.36% with 200 selected genes for the same Relief F test method.

TABLE 6: Performance analysis of classifiers in terms of classification accuracies (%) with Artificial fish swarm optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	90.24	91.67	89.05	88.41	86.82	81.03	94.90	93.75	93.75	93.75	89.6	93.75
Relief-F test	95.96	85.93	77.08	89.79	99.10	86.42	95.83	97.91	97.91	89.6	78.59	80.56
Chi-square test	82.59	83.48	78.12	84.10	95.34	94.61	95.83	98.63	93.75	94.49	97.91	84.39
T statistic test	93.75	89.75	87.5	86.78	92.29	89.33	89.6	97.91	97.91	95.83	94.79	97.91

TABLE 7: Performance analysis of classifiers in terms of PI (%) with Grasshopper optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	45.16	97.87	7.69	22.22	51.16	51.16	7.69	51.16	15.36	7.69	7.69	15.36
Relief-F test	60.87	80.01	51.163	51.163	22.22	78.93	22.22	97.87	91.58	66.66	22.22	15.36
Chi-square test	95.65	15.36	51.163	91.58	85.7	15.36	45.16	51.16	80.01	7.69	15.36	51.16
T statistic test	15.36	45.16	45.16	95.65	91.58	22.22	51.16	85.7	91.58	15.36	7.69	60.87

TABLE 8: Performance analysis of classifiers in terms of PI (%) with Moth flame optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	92.45	78.93	85.7	60.87	51.16	84.31	80.01	95.65	78.93	78.93	51.16	91.58
Relief-F test	85.7	45.16	80.01	95.85	95.65	61.14	60.87	22.22	91.58	45.16	95.65	78.93
Chi-square test	85.7	85.7	77.62	80.01	60.87	53.35	95.65	62.50	60.87	60.87	95.65	80.01
T statistic test	60.87	80.01	91.58	60.87	85.7	56.08	92.17	60.87	80.01	60.87	60.87	95.65

TABLE 9: Performance analysis of classifiers in terms of PI (%) with Bacterial foraging optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	95.65	93.17	79.93	91.58	60.87	60.87	57.22	45.09	78.31	60.87	64.03	85.7
Relief-F test	95.65	85.7	78.93	78.93	80.01	51.16	95.65	52.11	63.85	91.58	95.65	95.65
Chi-square test	85.7	78.93	80.01	97.00	77.11	94.45	94.96	60.87	95.65	95.65	91.58	91.58
T statistic test	80.01	85.7	60.87	52.98	91.58	95.92	63.76	95.65	56.00	91.58	85.7	78.93

TABLE 10: Performance analysis of classifiers in terms of PI (%) with Krill Herd optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	95.65	80.01	39.87	62.47	78.93	93.44	80.36	85.7	95.65	81.07	79.21	93.28
Relief-F test	78.93	95.65	51.16	45.16	60.87	64.68	96.68	28.94	17.93	50.06	78.04	53.72
Chi-square test	85.7	84.41	42.31	93.60	46.56	81.00	72.79	27.54	97.28	68.96	24.01	74.91
T statistic test	91.58	95.65	80.01	91.58	93.22	91.58	88.45	57.42	92.69	63.08	62.83	97.87

TABLE 11: Performance analysis of classifiers in terms of PI (%) with Artificial fish swarm optimization for different gene selection techniques using 50–200 selected genes.

Method	NBC			DT			SVM			KNN		
Genes selected	50	100	200	50	100	200	50	100	200	50	100	200
Information gain	77.67	80.01	75.76	72.02	64.12	37.59	88.01	85.7	85.7	85.7	78.93	85.7
Relief-F test	91.26	60.87	15.36	78.55	98.16	62.67	91.58	95.65	95.65	78.93	25.11	35.42
Chi-square test	46.56	50.67	22.22	53.37	90.08	88.09	91.58	97.14	85.7	87.6	95.65	54.61
T statistic test	85.7	78.67	66.66	65.91	82.65	77.2	78.93	95.65	95.65	91.58	88.64	95.65

6. Conclusion and Future Work

One of the most prominent lethal factors for human beings nowadays is cancer. The best chances of suitable treatment can sometimes be missed due to mistaken diagnosis. The accuracy of cancer diagnosis with machine learning along with clinical tests is very helpful in the treatment of cancer. Microarray expression data is highly redundant and with respect to most number of classes, the genes present are uninformative. Therefore, it is a critical necessity to select the best feature genes for the analysis of cancer. Out of a large dataset, the techniques should be capable of identifying a subset of most informative genes in a robust manner. In this work, a comprehensive analysis of lung cancer classification with the help of feature selection and optimization techniques is done. The best results are obtained when Relief-F test is computed with AFSSO and classified with Decision Trees classifier for hundred genes, and a highest classification accuracy of 99.10% is obtained. Future works aim to work with other feature selection techniques and a variety of optimization techniques classified with deep learning techniques for effective classification of lung cancer.

Data Availability

The data along with the program codes will be available to genuine researchers upon request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health and Welfare, Republic of Korea (grant number: HR21C0198), partly supported by the Korea Medical Device Development Fund grant funded by the Korea government (the Ministry of Science and ICT (MSIT), the Ministry of Trade, Industry and Energy, the Ministry of Health and Welfare, the Ministry of Food and Drug Safety) (Nos. 1711139109 and KMDF_PR_20210527_0005), and partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1F1A1061814).

References

- [1] S. K. Prabhakar and S.-W. Lee, "Transformation based tri-level feature selection approach using wavelets and swarm computing for prostate cancer classification," *IEEE Access*, vol. 8, Article ID 127462, 2020.
- [2] S. K. Prabhakar and S.-W. Lee, "An integrated approach for ovarian cancer classification with the application of stochastic optimization," *IEEE Access*, vol. 8, Article ID 127866, 2020.
- [3] S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.
- [4] J. Liu, W. Cai, and X. Shao, "Cancer classification based on microarray gene expression data using a principal component accumulation method," *Science China Chemistry*, vol. 54, no. 5, pp. 802–811, 2011.
- [5] S. K. Prabhakar, H. Rajaguru, and S.-H. Kim, "An amalgamated approach to Bi-level feature selection techniques utilizing soft computing methods for classifying colon cancer," *BioMed Research International*, vol. 2020, Article ID 8427574, 13 pages, 2020.
- [6] G. V. S. George and V. C. Raj, "Review on feature selection techniques and the impact of SVM for cancer classification using gene expression profile," *International Journal of Computer Sciences and Engineering Systems*, vol. 2, no. 3, pp. 1–12, 2011.
- [7] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
- [8] F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois, "Improving feature subset selection using a genetic algorithm for microarray gene expression data," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 2529–2534, Vancouver, Canada, July 2006.
- [9] M. Sardana and R. K. Agrawal, "A comparative study of clustering methods for relevant gene selection in microarray data," in *Advances in Computer Science, Engineering & Applications*, D. C. Wyld, J. Zizka, and D. Nagamalai, Eds., Springer, Berlin, Germany, 2012.
- [10] F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois, "A genetic algorithm-based method for feature subset selection," *Soft Computing*, vol. 12, no. 2, pp. 111–120, 2007.
- [11] R. Nakkeeran and T. A. A. Victoire, "Hybrid approach of data mining techniques, PCA, EDM and SVM for cancer gene feature selection and classification," *European Journal of Scientific Research*, vol. 79, no. 4, pp. 638–652, 2012.
- [12] G. Getz, E. Levine, and E. Domany, "Coupled two-way clustering analysis of gene microarray data," *Proceedings of the National Academy of Sciences*, vol. 97, no. 22, Article ID 12079, 2000.
- [13] X. Li, S. Peng, J. Chen, B. Lü, H. Zhang, and M. Lai, "SVM-T-RFE: a novel gene selection algorithm for identifying metastasis-related genes in colorectal cancer using gene expression profiles," *Biochemical and Biophysical Research Communications*, vol. 419, no. 2, pp. 148–153, 2012.
- [14] H. Zhang, C.-Y. Yu, B. Singer, and M. Xiong, "Recursive partitioning for tumor classification with gene expression microarray data," *Proceedings of the National Academy of Sciences*, vol. 98, no. 12, pp. 6730–6735, 2001.
- [15] G. Parmigiani, E. S. Garrett-Mayer, R. Anbazhagan, and E. Gabrielson, "A cross-study comparison of gene expression studies for the molecular classification of lung cancer," *Clinical Cancer Research*, vol. 10, no. 9, pp. 2922–2927, 2004.
- [16] L. Zhang, L. Wang, B. Du et al., "Classification of non-small cell lung cancer using significance analysis of microarray-gene set reduction algorithm," *BioMed Research International*, vol. 2016, Article ID 2491671, 8 pages, 2016.
- [17] J. Li, Y. Wang, X. Song, and H. Xiao, "Adaptive multinomial regression with overlapping groups for multi-class classification of lung cancer," *Computers in Biology and Medicine*, vol. 100, pp. 1–9, 2018.
- [18] H. Azzawi, J. Hou, Y. Xiang, and R. Alanni, "Lung Cancer prediction from microarray data by gene expression

- programming,” *IET Systems Biology*, vol. 10, no. 5, pp. 168–178, 2016.
- [19] P. Guan, D. Huang, M. He, and B. Zhou, “Lung cancer gene expression database analysis incorporating prior knowledge with support vector machine-based classification method,” *Journal of Experimental and Clinical Research*, vol. 278, no. 103, pp. 1–7, 2009.
- [20] R. De Santis, A. Gloria, S. Viglione et al., “3D laser scanning in conjunction with surface texturing to evaluate shift and reduction of the tibiofemoral contact area after meniscectomy,” *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 88, pp. 41–47, 2018.
- [21] P. Fucile, I. Papallo, G. Improta et al., “Reverse Engineering and Additive Manufacturing towards the design of 3D advanced scaffolds for hard tissue regeneration,” in *Proceedings of the 2019 2nd Workshop on Metrology for Industry 4.0 and IOT (MetroInd4.0 and IOT)*, Naples, Italy, June 2019.
- [22] S. Vanichayabon, W. Siriphan, and W. Wiphada, “Microarray gene selection using self-organizing map,” in *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*, Beijing, China, September 2007.
- [23] H. Liu, L. Liu, and H. Zhang, “Ensemble gene selection for cancer classification,” *Pattern Recognition*, vol. 43, no. 8, pp. 2763–2772, 2010.
- [24] L.-Y. Chuang, C.-S. Yang, K.-C. Wu, and C.-H. Yang, “Gene selection and classification using Taguchi chaotic binary particle swarm optimization,” *Expert Systems with Applications*, vol. 38, no. 10, Article ID 13367, 2011.
- [25] M. S. Mohamed, D. Safaai, and R. O. Muhammad, “Genetic Algorithms wrapper approach to select informative genes for gene expression microarray classification using support vector machines,” in *Proceedings of 3rd International Conference on Bioinformatics*, Auckland, New Zealand, May 2004.
- [26] S. A. Medjahed, T. A. Saadi, A. Benyettou, and M. Ouali, “Kernel-based learning and feature selection analysis for cancer diagnosis,” *Applied Soft Computing*, vol. 51, pp. 39–48, 2017.
- [27] T.-C. Lin, R.-S. Liu, C.-Y. Chen, Y.-T. Chao, and S.-Y. Chen, “Pattern classification in DNA microarray data of multiple tumor types,” *Pattern Recognition*, vol. 39, no. 12, pp. 2426–2438, 2006.
- [28] D. Q. Zeebaree, H. Haron, and A. M. Abdulazeez, “Gene selection and classification of microarray data using convolutional neural network,” in *Proceedings of the International Conference on Advanced Science and Engineering (ICOASE)*, pp. 145–150, Duhok Technical University, Duhok, Iraq, October 2018.
- [29] M. Khashei, A. Zeinal Hamadani, and M. Bijari, “A fuzzy intelligent approach to the classification problem in gene expression data analysis,” *Knowledge-Based Systems*, vol. 27, pp. 465–474, 2012.
- [30] M. Kumar, N. K. Rath, A. Swain, and S. K. Rath, “Feature selection and classification of microarray data using Map-Reduce based ANOVA and K-nearest neighbor,” *Procedia Computer Science*, vol. 54, pp. 301–310, 2015.
- [31] F. Vafaee Sharbaf, S. Mosafer, and M. H. Moattar, “A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization,” *Genomics*, vol. 107, no. 6, pp. 231–238, 2016.
- [32] K.-H. Chen, K.-J. Wang, K.-M. Wang, and M.-A. Angelia, “Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data,” *Applied Soft Computing*, vol. 24, pp. 773–780, 2014.
- [33] S. K. Pati, A. K. Das, and A. Ghosh, “Gene selection using multi-objective genetic algorithm integrating cellular automata and rough set theory,” in *Proceedings of the International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 144–155, Chennai, India, December 2013.
- [34] S. Wang, W. Kong, W. Zeng, and X. Hong, “Hybrid binary imperialist competition algorithm and tabu search approach for feature selection using gene expression data,” *BioMed Research International*, vol. 2016, Article ID 9721713, 12 pages, 2016.
- [35] J. Apolloni, G. Leguizamón, and E. Alba, “Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments,” *Applied Soft Computing*, vol. 38, pp. 922–932, 2016.
- [36] H. M. Zawbaa, E. Emary, A. E. Hassanien, and B. Parv, “A wrapper approach for feature selection based on swarm optimization algorithm inspired from the behavior of social spiders,” in *Proceedings of the 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 25–30, Fukuoka, Japan, November 2015.
- [37] D. Karaboga and B. Basturk, “Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems,” in *International Fuzzy Systems Association World congress*, pp. 789–798, Springer, Berlin, Germany, 2007.
- [38] I. Jain, V. K. Jain, and R. Jain, “Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification,” *Applied Soft Computing*, vol. 62, pp. 203–215, 2018.
- [39] A. Pino Angulo, “Gene selection for microarray cancer data classification by a novel rule-based algorithm,” *Information*, vol. 9, no. 1, p. 6, 2018.
- [40] C. Gunavathi and K. Premalatha, “A comparative analysis of swarm intelligence techniques for feature selection in cancer classification,” *The Scientific World Journal*, vol. 2014, Article ID 693831, 12 pages, 2014.
- [41] G. J. Gordon, R. V. Jensen, L. L. Hsiao et al., “Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma,” *Cancer Research*, vol. 62, no. 17, pp. 4963–7, 2002.
- [42] J. Wu, H. Wang, N. Li et al., “Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by adaptive grasshopper optimization algorithm,” *Aerospace Science and Technology*, vol. 70, pp. 497–510, 2017.
- [43] S. Mirjalili, “Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm,” *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [44] H. Chen, Y. Zhu, and K. Hu, “Self-adaptation in bacterial foraging optimization algorithm,” in *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering (ISKE 08)*, pp. 1026–1031, Xiamen, China, November 2008.
- [45] A. H. Gandomi and A. H. Alavi, “Krill Herd: a new bio-inspired optimization algorithm,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
- [46] Y. Wu, X. Z. Gao, Z. Kai et al., “Knowledge-based artificial fish-swarm algorithm,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 188–200, 2011.
- [47] S. K. Prabhakar, H. Rajaguru, and S.-W. Lee, “Metaheuristic-based dimensionality reduction and classification analysis of PPG signals for interpreting cardiovascular disease,” *IEEE Access*, vol. 7, Article ID 165181, 2019.
- [48] S. K. Prabhakar, H. Rajaguru, and S.-W. Lee, “Eigen vector method with swarm and non swarm intelligence techniques for epileptic seizure classification,” in *Proceedings of the 8th International Winter Conference on Brain-Computer*

Interface, IEEE Xplore, Gangwon, Republic of Korea, February 2020.

- [49] H.-D. Yang and S.-W. Lee, "Robust sign language recognition by combining manual and non-manual features based on conditional random field and support vector machine," *Pattern Recognition Letters*, vol. 34, no. 16, pp. 2051–2056, 2013.
- [50] H. Rajaguru and S. K. Prabhakar, "Power spectral density and KNN based adaboost classifier for epilepsy classification," in *Proceedings of the IEEE International Conference on Electronics, Communication and Aerospace Technology (ICECA 2017)*, pp. 441–445, IEEE Xplore, Coimbatore, India, April 2017.