International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020

Integrating Numerical and Quantitative Techniques for Analysis of Large-Scale Biological data using Hyper-heuristic Algorithm

Prachi Vijayeeta, M. N. Das, B. S. P. Mishra.

Abstract - A major thrust in the field of computational intelligence is the ability to comprehend and interpret intelligence in combination with several research disciplines of computer science, biology, statistics, and cognitive science. The emergence of bio informatics is a two-fold manifestation of advance biology along with data mining and statistical learning that attempts to analyse and interpret a large collection of medical data. It aims at exploring facts about unknown patterns. This paper aims at providing a model that numerically analyses the existence of carcinomas in the genomic data sequence. Attempts have been taken to optimise the highly contributing factors using recent nature inspired algorithms like Emperor Penguin Optimization Algorithm (EPOA) and Chaotic Artificial Algae Optimization(CAAO). These algorithms are gaining popularity as they are capable of exploring global optimum instead of local optimum. The performance of these algorithms are judged by implementing it in seven benchmark datasets. To ensure a good potential of these new algorithms, we have made a comparative study with the capability of other indigenous optimization algorithms like PSO etc.

Keywords: Optimization, microarray datasets, support vector machines, k-fold validations.

I.INTRODUCTION

The dynamic evaluations of carcinomic and immune cells with a view of finding the most significant way to control the abnormal growth of tumours is a major challenge in the recent research area. Accurate diagnosis and proper treatment seems to be carried out hand-in-hand with the advent of time factor. The stochastic behaviour of cancer cells makes the treatment and the chances of cure extremely complex for the medical practitioners. However, the major focal point of our study is to set-up an efficient strategy for predicting the cancer-causing factors at the genome level and its possible chemotherapic treatments [1]. In this work, a two-fold manifestation has been carried out with an objective of simplifying our simulated task and secondly by optimizing our hyper-parameters. Analysis of large-scale topological data has been made possible with the help of decision support tools that takes into consideration right from the atomic and cellular levels.

Manuscript received on April 02, 2020.
Revised Manuscript received on April 21, 2020.
Manuscript published on May 30, 2020.
* Correspondence Author
Prachi Vijayeeta*, School of Computer Engineering, Kalinga Institute of Industrial Technology, [Deemed To Be University], Bhubaneswar.
M. N. Das, School of Computer Engineering, Kalinga Institute of Industrial Technology, [Deemed To Be University], Bhubaneswar.
B. S. P. Mishra. School of Computer Engineering, Kalinga Institute of Industrial Technology, [Deemed To Be University], Bhubaneswar.

According to oncologists, the growth of the tumours at the step cellular, cellular or at the tissue levels can be controlled at various times scales. From genetic mutational point of view, the primary cause of tumour formulation due to improper gene-pairing during genetic mutation phase, currently, many researchers have put forth multi-scale models to simulate both numerical as well as therapeutic optimization process at various time scales. The advantage of these models are to lead towards faster global optimal solution, reduces interference of irrelevant factors, allows diagnosis from the grass-root level and ensures treatment of the disease confined to that particular spatial locations and non-locations [1], our work is precisely based on identification of biomarkers from the high-dimension and microarray gene dataset. But this high-dimensional mature of the dataset is more vulnerable to the computational overhead. It is therefore advisable to choose only the most contributing genes and eliminate the irrelevant as well as redundant genes [2]. Researches do believe that selection of relevant genes locals to the reduction in the size of the gene expression thus improving the computational accuracy. Kernel PCA and Kernel LDA are used for selecting the relevant features that are highly responsible for the occurrence of malignancy. These selected features are further subjected to the classifier for classifying benign and malignant tissues. Our research work lays emphasis on deciding the appropriate weights and biases that are to be fed into the classifier model for generating significant outputs. Recently developed swarm-based hyper heuristic algorithms are implemented to optimize the parameters. A Emperor penguin optimization (EPO) algorithm and a Chaotic Artificial Microalgae Optimization (CAAO) algorithms are applied on bench mark data sets. The classification accuracy and the performance measures are compared. Moreover, we have also tried to incorporate traditional swarm intelligence algorithm like PSO [19] which seemingly has resulted in less accurate results in comparison to the recent ones. The computational time complexity is estimated based upon the input size and the number of iterations carried out. The paper is organized as follows: section 1 introduces the problem statement with its objectives, section 2 describes the state of art of the works carried out so far, section 3 demonstrates the working process model along with the workflow diagram. Section 4 suggests the significance of the applied technologies, feature reduction methodologies and parameter settings, section 5 briefly explains the experimental set up with valid dataset descriptions, section 6 graphs are plotted and the behaviour of the variables are thoroughly studied. Finally, section 7 concludes the paper with an authentic suggestion for future work.

II.RELATED WORK

A broad spectrum of formalisms and comparison between filter and wrapper techniques over both discrete and continuous datasets Inza et al. [3]. Liu et al.[4] had built up a three dimensional integrated intelligent system for feature selection that could automatically recommend the most suitable algorithm to the user both for labelled as well as unlabelled data . Lee [5] used Chi-Square test along with SVM (Support Vector Machine) and dynamic parameter setting GA for identifying top-ranked homogeneous genes. Sushmita Mitra [6] published in their article a consortium of Rough set theory for efficient selection of genes from microarray gene expression patterns. Supervised principal component analysis was suggested by Hastie at al. [7] to generate a subset of predictors based on their association with the outcome. They had worked on survival analysis with the help of these components and extended their contribution for generalised regression problems. Peng [8] had employed a kernel based method on five multi-class cancer datasets to describe a non-linear correlationship among multi-variates. Chuang et al.[9] had applied Binary PSO by improving the running time and the complexity over gene expression datasets. Benjamin [10] had designed a numerical based model to study the dynamic evolutions of cancer and immune cells. They had also formulated a therapeutic optimization mechanism to maximise the immune cells. An integration of particle swarm optimization and support vector machine (PSO-SVM) along with genetic algorithm (GA-SVM) was experimented by Alba [11] that generated a better accuracy than other traditional methods. Many other learning algorithms like k-Nearest Neighbour (KNN) [12], Naïve Bayes (NB) [13], Extreme Learning Machine (ELM)[14] , Support Vector Machine(SVM) [15], Decision Tree [16], Random Forest [17].

Apart from this, numerous metaheuristic and hyper heuristic methodologies have been developed to solve a wide range of optimization problem. Some of the nature inspired evolutionary learning algorithms includes genetic algorithm [18], differential evolution, simulated annealing, particle swarm optimization, ant colony optimization, artificial bee colony optimization, cat swarm optimization [21], elephant herd optimization, gravitational search optimization, salp swarm optimization, emperor penguin optimization, firefly optimization, dragonfly algorithm, crow search optimization, cuttle fish optimization, social engineering optimization etc. have a unique property of exploring a huge search space to generate an optimal solution in an efficient manner. Many hybridised algorithms such as genetic bee colony optimization, ant bee colony optimization, Emperor penguin social engineering algorithm etc. is employed to auto-learn the hyper-parameters for performance improvement [26].

III.WORKING PROCESS MODEL

Let us consider a continuous optimal problem:-

$$Min\ J = \int_{T_i}^{T_f} F\left[u(t), \overrightarrow{w_j}(t)\right] + f_f\left(u_f, \vec{w}_f . T_f\right) . dt \quad (1)$$

Subject to the constraint:-

$$\frac{d\overrightarrow{w_j}}{dt} = f_{ct}\left(\overrightarrow{w_j}, u, t\right), t > 0 \quad (2)$$

$$\frac{du}{dt} = f_{ct}(u, t) \quad (3)$$

$$u_{min} < u < u_{max} \quad (4)$$

$$\vec{w}_{min} < \vec{w}_j < \vec{w}_{max} \quad (5)$$

Where $\vec{w}_j$ represents the weighted sum of different types of cells (cancerous/ normal/immune cells)

u is the treatment criteria

f is the final value function.

This mathematical model builds up the gap between theoretical and clinical applications. Optimal control theory has been used in cancer literature to determine the control and state trajectories for a dynamic system over a period of time to minimise the final value of a single variable J. However, we can reduce equation (1) to a linear function to minimise the number of tumour cells at the end of the treatment period.

$$Min\ J = \int_{T_i}^{T_f} [c(t) + u(t)] . dt \quad (6)$$

Where $c(t)$the number of is cells sensitive to treatment and $u(t)$ is the treatment concentration at the site of action. $u = (u_1, u_2, \ldots . u_m)^T$is a control vector. In our work we have mapped $c(t)$ to population size at a given instance of time t. Computational biology scores better performance and accurate result when local search is coupled with global search mechanism. Recent approaches termed as hyper heuristic algorithms are devised to ensure local search optimization using evolutionary algorithms followed by global search mechanism adopting another nature inspired algorithm. In our work we have employed Chaotic Artificial Microalgae Optimization (CAAO)a metaheuristic as a local search mechanism to generate the most contributing features and Emperor Penguin optimization algorithm as a global search mechanism to optimize the hyper parameters of the learning model.

IV.METHODS APPLIED

In a high dimension dataset, it is always wise to segregate only the informative genes that acts as a vital mechanism in performing accurate classification. In the present work we have applied Kernel based PCA (Principal component analysis) and Kernel-LDA (Linear discriminant analysis) which seemingly has yielded better result than other dimensionality reduction techniques like RFE(Recursive feature elimination) [25], mRMR(minimum Redundancy Maximum Relevance) [26].

Various statistical methods like T-score, Z-score, and Fisher Score have also
been applied for feature selections where the inter-class and intra-class distance are the vital parameters for deducing the class type. These filter techniques have enabled the researchers to review the statistical behaviour of every gene within the class, thereby proving it to be one of the most discriminatory factors [22].

4.1 Selection of Gene Subsets using kernel-PCA

Kernel-PCA extends conventional PCA to a high dimensional feature space using kernel trick. It has the potential to extract nearly 'n' non-linear principal components with minimal complexity [27]. A kernel function (Eq 7) is used to project a dataset into a high dimensional feature space with linear separability. Among the popular kernels such as Gaussian Kernel, Polynomial kernel and Hyperbolic Tangent kernel, we have employed Gaussian kernel[28]. The primary goal of kernel is to compute the eigenvalue problem with λ representing the eigen values and $\vec{\alpha}$ being the eigenvectors (Eq 8). Eigen vectors are expressed as linear combination of features (Eq 9).

$$k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \quad (7)$$

$$k\vec{\alpha} = \lambda\vec{\alpha} \quad (8)$$

$$Cv = \lambda v \quad (9)$$

Where C is the covariance and v is the eigen vector with α_i being the coefficient that needs to be computed prior to the implementation of kernel functions.

$$C = \frac{1}{n}\sum_{i=1}^{n} \varphi(x_i)^T \varphi(x_i) \quad (10)$$

And $v = \sum_{i=1}^{n} \alpha_i \varphi(x_i)$ (11)

Substituting equation (10) and (11) in equation (9) we will get

$$\left(\frac{1}{n}\sum_{i=1}^{n} \varphi(x_i)^T \varphi(x_i)\right) v = \lambda v$$

$$\Longrightarrow v = \frac{1}{\lambda}\left(\frac{1}{n}\sum_{i=1}^{n} \varphi(x_i)^T \varphi(x_i)\right) v \quad (12)$$

Now, upon plugging the kernel and normalizing the feature space we will get the transformed features as:

$$\tilde{\varphi}(x_k) = \varphi(x_i) - \frac{1}{n}\sum_{i=1}^{n} \varphi(x_k) \quad (13)$$

$$\tilde{K}(x_i, x_j) = \tilde{\varphi}(x_k)^T \tilde{\varphi}(x_k) \quad (14)$$

Substituting equation (13) in equation (14), we will get:

$$\tilde{K}(x_i, x_j) = \left(\varphi(x_i) - \frac{1}{n}\sum_{i=1}^{n} \varphi(x_k)\right)^T - \left(\varphi(x_i) - \frac{1}{n}\sum_{i=1}^{n} \varphi(x_k)\right) \quad (15)$$

$$\Longrightarrow K(x_i, x_j) - \frac{1}{n}\sum_{k=1}^{n} K(x_i, x_k) - \frac{1}{n}\sum_{k=1}^{n} K(x_j, x_k) - \frac{1}{n^2}\sum_{l,k=1}^{n} K(x_l, x_k) \quad (16)$$

Algorithm for k-PCA

Step-1: Input a M- dimensional data matrix (V) in the feature space of (M_F) dimension.

Step-2: Select an appropriate kernel function $k(x_m, x_n)$ for mapping from input space to feature space.

Step-3: Generate kernel matrix based on the training data $\{x_n, for\ n = 1\ to\ N\}$

Step-4: Compute the eigenvalue problem of the kernel matrix to obtain $\lambda_i\ and\ \alpha_i$.

Step-5: Determine the correlation score of the vector $corr(V)$.

Step-6: Obtain the principal components $PC_1, PC_2, \ldots . PC_n$ using Equation (11).

Step-7: Sort the components in the descending order of their correlation score.

Step-8: Save the current vector with the highest correlation score.

Step-9: Repeat this process for all possible combinations of PCs that can increase the correlation score.

Step-10: Finish.

4.2 Selection of Feature Subsets using kernel-LDA

Kernel discriminant analysis also called as generalised discriminant analysis (GDA) is employed to map the data points into a new feature space in a non-linear manner. The primary aim of GDA is to provide a large separation of class means along with a minimum in-class variance. Researchers mostly prefer kernel methods since it maps the data points explicitly without having any prior knowledge about non-linearity [29]. If we consider equation (7) and apply Gaussian Kernel function as:

$$k(x, y) = \frac{exp(-\|x-y\|)^2}{\sigma} \quad (17)$$

where $\sigma \in \mathbb{R}$ and x, y are the data points.

Algorithm for k-LDA

Step-1: Input a M- dimensional data matrix (V) in the feature space of (M_F) dimension.

Step-2: Select an appropriate kernel function $k(x_m, x_n)$ for mapping from input space to feature space using Equation (17).

Step-3: Generate kernel matrix based on the training data $\{x_n, for\ n = 1\ to\ N\}$.

Step-4: Compute the eigen vectors and eigen values for the expression:

$$K_b {K_b}^T \alpha = \lambda K_w {K_w}^T \alpha \qquad (18)$$

where K_b, K_w are the between class and within class squared matrix respectively.

Step-5: Normalise the eigen vectors α and eigen values λ.

Step-6: Compute the projections of the test points onto the eigen vectors.

Step-7: Finish.

Subsequently the selected gene subset are subjected to the suggested optimization techniques to obtain the best subset for further classifying the elements.

4.3 Kernel-Support vector machine (SVM)

Support vector machine is one of the stronger and powerful tool for building classification as well as regression models. It was first proposed by Vapnik in the year 1995 with an idea of locating each data point (n features) on a n-dimensional space. The plotted co-ordinates thus represents the values of each features. Support vector machine estimates a multivariate function $\emptyset(x)$ to associate an input vector $x \epsilon R^n$ onto a large dimensional space. In Fig. 1 a hyperplane is estimated for maximization of margin, i.e the distance between the margin and the data points of the class within the space S [30]. The classification boundary for all values of x, such that $f(x) = 0$ is a hyperplane defined by $wx + b = 0$ where:-

- $w$is a normal vector.
- $\frac{b}{||w||}$is the perpendicular distance from the hyperplane to origin.

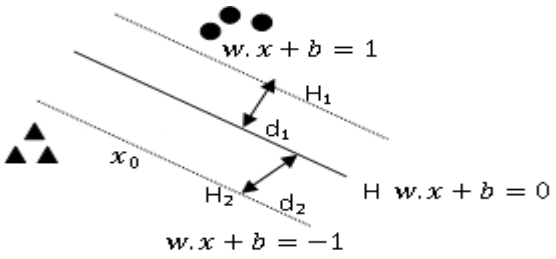

Fig 1. Geometrical interpretation of SVM

M.kumar[31], had experimented Polynomial kernel, RBF kernel and sigmoid kernel on ALL and AML having 72 samples. Hamidreza [32] had proposed a modified support vector machine (υ-SVM) algorithm in which a parallel training is used in sub-classifiers. From the literature survey we have come across the three popular kernel function $K(x_i, x_j)$ with γ, c, d as the kernel parameters.

1. Polynomial Kernel:- $k(x_i, x_j) = (\gamma {x_i}^T . x_j + c)^d; \gamma > 0, c \geq 0.$
2. RBF Kernel:-$k(x_i, x_j) = exp(-\gamma \parallel x_i - x_j \parallel^2);\ \gamma > 0$
3. Sigmoid:- $k(x_i, x_j) = tanh(\gamma {x_i}^T . x_j + c)\ \gamma > 0, c \geq 0.$

In our work we have applied kernel-SVM that enables non-linear mapping of data points to a high dimensional space. The advantage of kernel function is to formulate a linear decision plane with the help of non-linear transformations. Eventually, the selection of appropriate kernel function, proper parameter tuning and selection of optimal decision boundary renders a major contribution to the optimization problem. If $X = \{x_1, x_2 \dots \dots x_n\}$ be the data input and $Y = \{y_1, y_2 \dots \dots y_n\}$ be the output as the class labels then, the purpose of employing SVM is to explore an optimal hyper plane H that could separate the data samples into two binary class [20]. Mathematically, we can represent H in the form:

$$W^T x + b = 0 \qquad (19)$$

Where W represents the weight vector and b represents the bias.

It is now required to fit this problem into a constrained optimization problem where W, b needs to be optimised.

$$Min\ z(w) = \frac{1}{2} \parallel w \parallel + C \sum_{i=1}^{l} \xi_i \qquad (20)$$

Subject to the constraint:-

$$y_i((w.x) + b) \geq 1 - \xi\ ; for\ i = 1,2..n\ ;\ \xi \geq 0 \qquad (21)$$

where C is the penalty parameter for no. of misclassified data and ξ_i is the distance measures of the points crossing the margin. This regularization parameter C controls the trade-off between the complexity and accuracy of the applied classification model. Ultimately, the

$$(w) = \begin{cases} [w^T x_i + b] \geq 1 - \xi,\ for\ y_i = +1\ , \xi_i \geq 0 \\ [w^T x_i + b] \leq -1 + \xi,\ for\ y_i = -1, \xi_i \geq 0 \end{cases}$$

(22)

4.4 Emperor Penguin Optimization

In our work, a recently developed meta heuristic algorithm named as Emperor Penguin Colony optimization algorithm is employed. Dhiman [33] had initially proposed this bio-inspired algorithm by simulating the huddling behaviour of penguins during the crowding process. A mathematical model has been designed by taking into consideration the parameters like the huddle boundary, the encircling temperature, their distance and the most active mover.

The emperor penguins are a particular species of flightless penguins mostly found in the open ice of Antarctic belt. They breed during the winter season by moving ashore in large colonies [24]. The female penguins lay a single egg and moves toward the ocean surface for hunting.

A. Phases of huddling behaviour of Penguins

There are basically four phases of the huddling behaviour of penguins:

Phase-I: Creating and estimating huddle boundary of emperor penguins. The huddle boundary is randomly generated and hypothetically it is placed on a two-dimensional L-shaped polygon plane.

Phase-II: Evaluate the body heat radiation. The emperor penguins gather themselves to conserve energy and maximise the temperature factor T. They keep on exploring different locations by enhancing the temperature parameter $T = T'$ which is computed below:

$$T' = \left(T - \frac{Iteration_{max}}{x - Iteration_{max}}\right) \quad (24)$$

$$T = \begin{cases} 0, & if\ R > 0.5 \\ 1, & if\ R < 0.5 \end{cases}$$

Where R is the radius of the polygon, T is the best optimal solution in the search space. $Iteration_{max}$ is the maximum number of iterations and x defines the current iteration.

Phase-III: Calculate the distance and the velocity of spiral movement around the huddle. After the creation of huddle boundary, the distance between the emperor penguin and the optimal best solution is calculated. The remaining penguins shall update their positions according to the current best optimal solution which is computed as follows:

$$\overrightarrow{Dist_{ep}} = Abs\left(S(\vec{A}).\overrightarrow{P(x)} - \vec{C}.\overrightarrow{P_{ep}(x)}\right) \quad (25)$$

Where $\overrightarrow{Dist_{ep}}$ represents the distance between the emperor penguin and the best optimal fitness penguin, $\vec{A}, \vec{C}$ are the parameters to control the random search and to avoid collision among the penguins. $\overrightarrow{P(x)}$ represents the best fit penguin where as $\overrightarrow{P_{ep}(x)}$ denotes the position vector of penguin. $S(\vec{A})$ refers to the social forces that pulls down the penguins towards the optimal best value. If M is the movement parameter responsible to maintain gap among the penguins to avoid collision then, the values of the parameters $\vec{A}, \vec{C}$ can be estimated as follows:

$$\vec{A} = \left(M * \left(T' + P(Accuracy)\right) * Rand()\right) - T' \quad (26)$$

$$P(Accuracy) = Abs(\vec{P} - \overrightarrow{P_{ep}}) \quad (27)$$

$$\vec{C} = Rand() \quad (28)$$

$Rand()$ is a random function whose value lies in the range [0,1] and M is set to 2.

$$S(\vec{A}) = \left(\sqrt{f.e^{\frac{-t}{l}} - e^{-t}}\right)^2 \quad (29)$$

Where e denotes an expression function, t is the iteration count. l, f are the control parameters for maintaining exploration and exploitation of penguins in such a manner that the lower and upper bounds of l, f are set to as follows: $l \in [1.5,2]$ and $f \in [2,3]$.

Phase-IV: Re-compute and update to the best position of the penguin. subsequently, other positions are also updated proportionately in the following manner:

$$\overrightarrow{P_{ep}}(t+1) = \vec{P}(t) - \vec{A}.\overrightarrow{Dist_{ep}} \quad (30)$$

4.5 Chaotic Artificial Microalgae Optimization (CAAO)

Chaotic Artificial Microalgae Optimization is a nature inspired meta heuristic optimization algorithm that mimics the lifestyle of a photosynthetic microalgae (also known as phytoplankton). The algorithm is designed based upon three parameters such as algal reproduction by mitotic division, adaptation and direction of movement towards the light source [34]. The objective function has a global optimum at the point where it receives the maximum light rays for photosynthesis with adequate nutrients.

Algorithm of Chaotic Artificial Microalgae Optimization

Step-1: Determine the parameters: Problem Dimension (N), population size (N+1), maximum number of iterations (M) and energy loss (E).

Step-2: Initialise the population of algal colonies

$$P_{AC} = \begin{bmatrix} x_1^1 & \dots & x_1^D \\ \vdots & \dots & \vdots \\ x_N^1 & \dots & x_N^D \end{bmatrix} \text{ and}$$

$i_{AC}^{th} = [x_i^1, x_i^2 \dots x_i^D]$ where x_i^j is the algal cell in the j[th] dimension of the i[th] algal colony.

Compute $x_i^j = x_i^{min} + \left(x_i^{max} - x_i^{min}\right) * rand(0,1)\ \ for\ i \in 1\ to\ N\ and\ j \in 1\ to\ N+1$ (31)

Step-3: Define the initial size of every algal colony $S_j = 1\ (for\ j = 1,2 \dots N+1)$ and initial hunger level of $H_j = 0\ (for\ j = 1,2 \dots N+1)$

Step-4: Set the iteration counter $t = 0$.

Step-5: Update $t = t + 1$

Step-6: Compute the objective function (Z_j) and find the best of objective function (Z_{best}) corresponding to the best algal colony $x_i^{best}, i = 1,2 \dots N$.

Step-7: Evolutionary Phase: Calculate the size, friction surface and energy of each algal colony.

$S_j = S_j + \mu_j S_j \ (for\ j = 1,2..N+1)$ where μ_j is a Monod function for algal growth and is computed as follows:

$$\mu_j = \frac{2Z_j}{S_j + 2Z_j} \ (for\ j = 1,2..N+1) \quad (32)$$

$$Max\ Z'_j = \frac{Z_j - Z_{min}}{Z_{max} - Z_{min}} \quad (33)$$

And The Energy of each algal colony is:-

$$E_j = \frac{S_j^2 - S_{min}^2}{S_{max}^2 - S_{min}^2} \ (for\ j = 1,2..N+1) \quad (34)$$

Step-8: Set $j = 0$.

Update $j = j + 1$

Step-9: While $(E_j > 0)$, perform helical movement using $\tau_j = 2\pi \left(\sqrt{3}\frac{3S_j}{4\pi}\right)^2 \ (for\ j = 1,2..N+1)$ (35)

Where τ_j is the friction surface and decrease the energy using

$$(E_j = E_j - \frac{\epsilon}{2}) \quad (36)$$

Step-10: If there is no improvement in the objective function then increase the hunger level

$$H_j = H_j + 1$$

Step-11: If $(j < N + 1)$ then go to step 8 else go to step 12

Step-12: Update the size of the algal colony using equation (31) to find the optimal colony. Introduce logical map a popular chaotic function to ensure a faster rate of convergence. Random variable **rand** is replaced by:

$\alpha_{t+1} = 4.\alpha_t(1 - \alpha_{t+1}) for\ t \in 1,2 \dots N$ is the iteration counter, α_t is a logistic chaotic variable with the t[th] chaotic iteration. The initial value at $t = 0$ is generated randomly between [0,1].

Step-13: Perform the adaptation as follows:

$$x_i^t = \begin{cases} x_j^t + \left[Biggest\ algal\ colony(x_i^{max}) - x_i^{min}\right) * \alpha_t, \\ if\ \alpha_t \geq 0.5 \\ \\ x_j^t - \left[Biggest\ algal\ colony(x_i^{max}) - x_i^{min}\right) * \alpha_t, \\ if\ \alpha_t < 0.5 \end{cases}$$

Step-14: If $(t < M)$ then go to step (5)

Else

For each updated candidate **do**

Compute steps 6 to 9 to obtain the new fitness function using equation (33) and the energy of the new algal colony using equation (34).

End for

If the $current\ fitness > previous\ fitness$

Then update the algal colony to the new fitness value

Else

Store the previous value.

Step-15: Finish.

V.EXPERIMENTAL SET-UP

5.1 Dataset Description

The potentiality of the hyper heuristic methods applied in our work has been tested on seven microarray benchmark datasets, listed in the Table- below. These datasets belong to Kent Ridge Biomedical Dataset repository. Leukaemia, colon tumour, ovarian cancers are of binary class and the remaining dataset are of multi-class type [23]. The datasets are normalised using mi-max normalization to confine the values within the range of [0, 1]. This leads to the generation of improved accuracy of the classifier as well as overcomes the flaw associated with higher and lower numeric ranges of values.

$$Y = \frac{X - \min}{max - min} \quad (37)$$

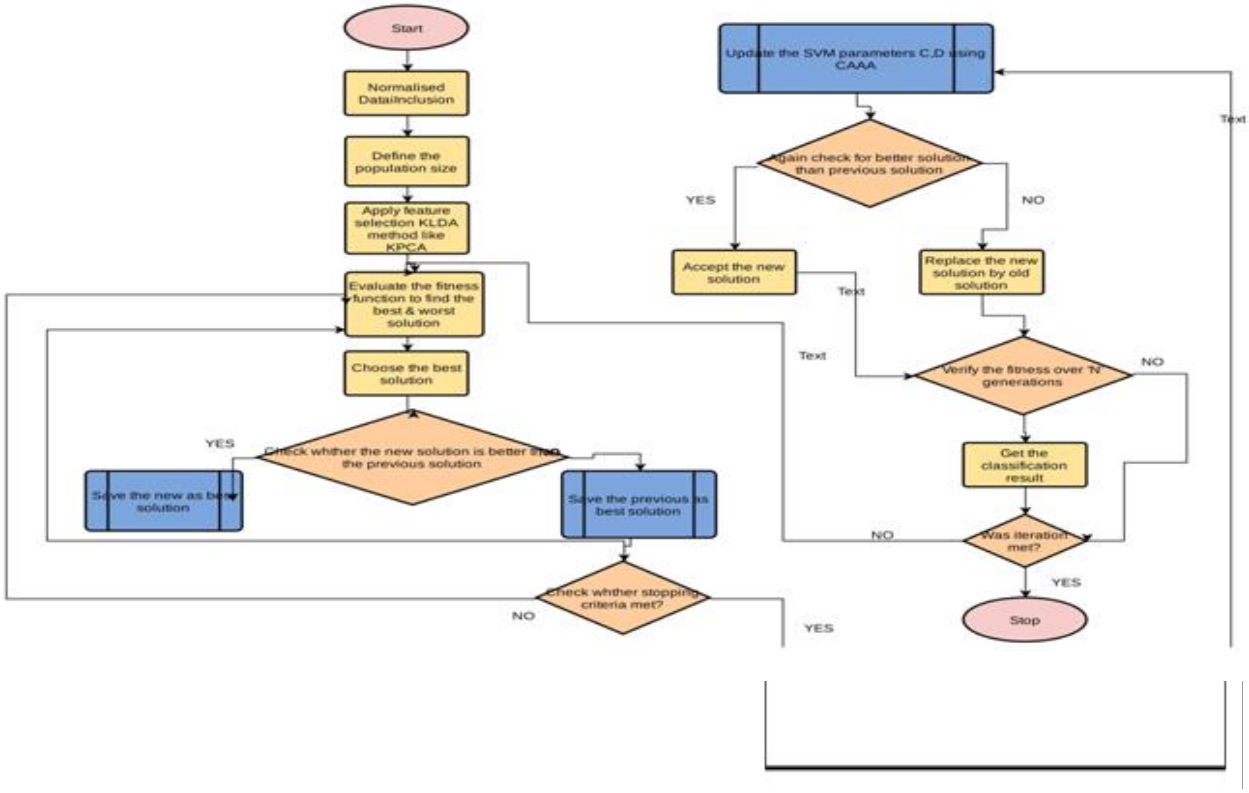

Fig.2 Workflow Diagram of the Hyper heuristic learning model

Table 1- Dataset Description with class types

Sl_No.	Dataset_Name	No. of Attributes	No. of Instances	Class Type
1	Colon	2000	62	2
2	Ovarian	15154	253	2
3	Leukaemia	7129	72	2
4	Lung cancer	12,600	203	5
5	ALL-AML	7129	72	3
6	SRBCT	2308	83	4
7	Lymphoma	4026	62	3

X_1	X_2	X_3	X_4	X_5	....	.....	X_n
Parameter Representation (C,γ)			Feature representations				

5.2 Experimental set-up

The experiment has been conducted using scikit learn in python environment on a PC with Intel core i5 CPU(GHz) and 16 GB of RAM. A 10-fold cross validation has been employed to obtain impartial classification accuracy.

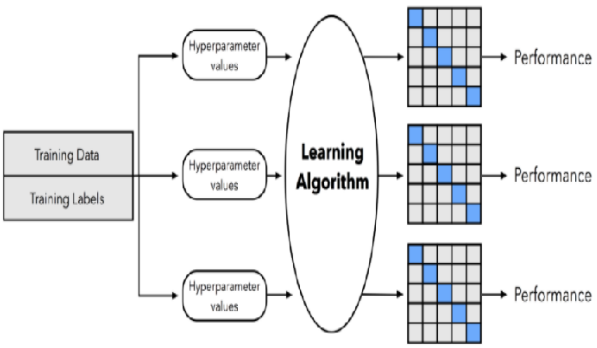

Fig.3 Representation of solution space and schematic view of learning model

Fig. 4 Representation of 10-fold cross validation

5.3 Performance Metrics

To judge the proficiency of applied model we need to evaluate various performance Metrics like classification accuracy, sensitivity, specificity, confusion matrix, F-measures, Kappa [25].

- **Confusion Matrix:** It is a 2-dimensional representation for evaluating the accuracy, sensitivity, specificity, confusion matrix, F-measures of a classification model. The details are demonstrated in Table-3.

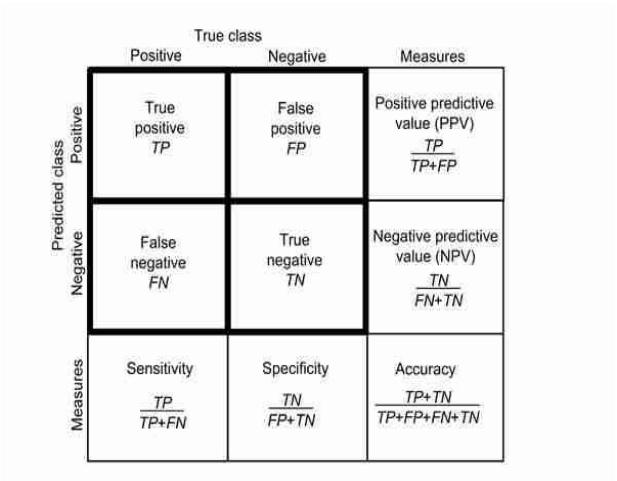

Fig.5 Confusion Matrix

- **Classification Accuracy:** It is the ratio of number of correct predictions to the total number of predictions made.
- **Kappa**: Kappa statistics takes into account the accuracy that would have happened through random predictions. It measures the extent to which the data raters actually guess on some variables due to uncertainty.

$$K = \frac{P_{agree} - P_{chance}}{1 - P_{chance}}$$

P_{agree} = Proportion of trials in which judges agree

P_{chance} = Proportion of trials in which agreement would be expected due to chance

Fig.6 Kappa statistics formula

- **Mathews correlation coefficient (MCC)**

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TN+FN)(TN+FP)(TP+FN)(TP+FP}}$$

Fig.7 MCC evaluation

- **F-Score:** It is a statistical analysis tool to measure the test accuracy by creating an optimal balance between recall and precision. The computational formula is mentioned in Fig.8.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Fig.8 F-score estimation

VI.SIMULATION AND RESULTS

A. Results of Feature Reduction

For improving the performance and efficiency of the model we need to minimize the dimension of the dataset. Kernel PCA and kernel LDA are employed to remove the redundant features as well noise. Even though a 2-dimensional microarray stores very large-scale biological data in a structured manner but still it needs to be scaled down to lower dimensions. Table-2 depicts the original number of attributes and the number of attributes after KPCA and KLDA reduction process. Fig.9 represents the graph plotting of the attributes and the samples collected from seven cancer datasets. Fig.10 resembles the graph for attributes 'vs' reduction process. In the Fig the percentage of accuracy scored applying KPCA and KLDA are plotted.

Table-2 Effect of Feature Reduction on Dataset

Sl_No.	Dataset_Name	No. of Attributes	After KPCA	After KLDA
1	Colon	2000	41	2
2	Ovarian	15154	72	2
3	Leukaemia	7129	55	3
4	Lung cancer	12,600	61	2
5	ALL-AML	7129	68	2
6	SRBCT	2308	83	3
7	Lymphoma	4026	62	2

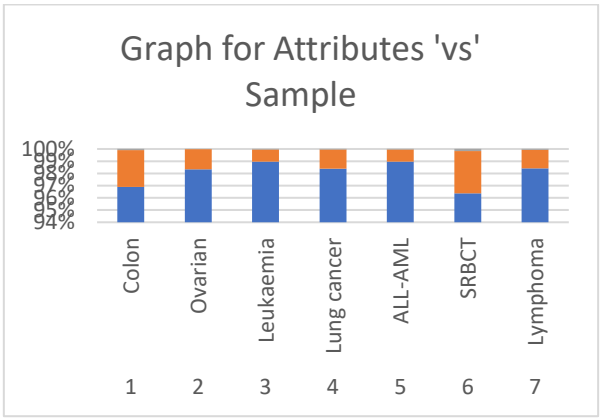

Fig. 9 Plotting of Attributes and samples

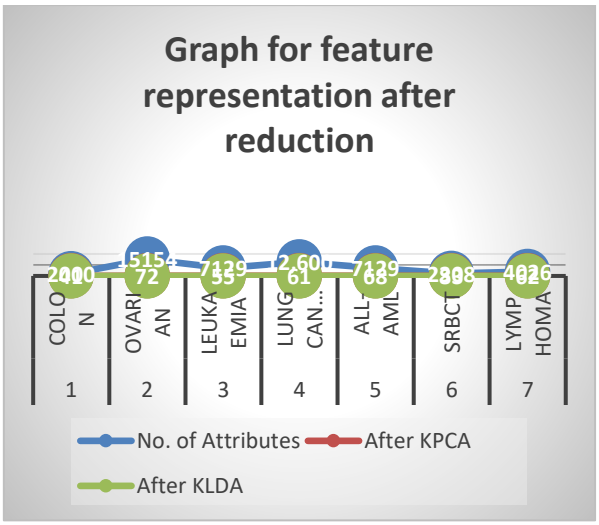

Fig.10 Representation of Attributes after feature reduction

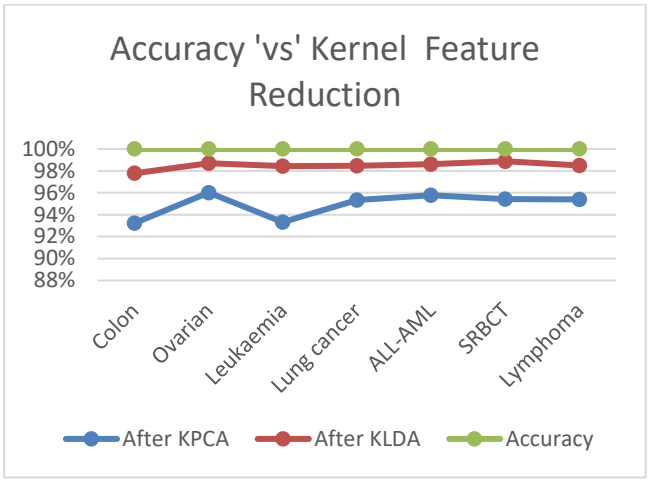

Fig.11 Graph plotted for Accuracy obtained from feature reduction.

B. Results of Classification

To analyse the effect of the suggested hyper heuristic model we have taken the help of the publicly available seven microarray datasets such as Colon, Ovarian, Leukaemia, Lung cancer, ALL-AML, SRBCT, Lymphoma. Table 3 represents the six performance measures such as accuracy, precision, recall, F-score and Mathews correlation coefficient. Graph for accuracy of each of the seven dataset is shown in Fig.11 The two parameters C,γ are represented in the search space and the diagrammatic view of the functionality of the learning model is also clearly mentioned. Fig represents the 10-fold cross validation over 10 independent runs. Table-4 represents the Parameter settings and the possible range of values for computing the performance metrics after 10 runs. According to our simulation Lung cancer has an accuracy of 98.7% followed by Lymphoma and ALL-AML having 98.2% accurate classification. Table-5 provides an observation of the execution time in seconds consumed by three meta heuristics algorithms. Upon keenly analysing we can conclude that even though lung cancer has the best accuracy but it takes a lot of time for execution as compared to others.

Table-3 Performance Metrics Evaluation values

Sl_No.	Dataset Name	Accuracy	Recall	Precision	F-Score	Kappa measures	MCC
1	Colon	0.976	0.981	0.972	0.975	0.953	0.911
2	Ovarian	0.981	0.971	0.983	0.982	0.974	0.951
3	Leukaemia	0.927	0.946	0.882	0.905	0.943	0.923
4	Lung cancer	0.987	0.983	0.988	0.981	0.962	1
5	ALL-AML	0.982	0.981	0.972	0.973	0.981	0.983
6	SRBCT	0.971	0.977	0.9773	0.977	0.964	0.952
7	Lymphoma	0.982	0.971	0.985	0.983	0.973	0.962

VII.CONCLUSION AND FUTURE WORK

An attempt for integrating numerical computation with the help of a Hyper heuristic algorithm is designed and simulated using Scikitlearn. In comparison to previous research works, we could realise some better promising results from kernel based (especially Gaussian kernel) feature reduction techniques instead of simple ones. After certain repetitions of generations, it is observed that the rate of convergence and the stability of the results is uniformly maintained. The learning model optimises the control parameters (C,γ) of SVM using two optimization algorithm such as CAAO and EPO for local search and global search respectively. The future work is directed towards employing of Deep learning, deep convolution network and some Quantum computing mechanism to improve the parameter efficiency. Instead of optimising the control parameters, many other parameters like population size, maximum iterations, movement velocity, energy loss, radiations of light source etc. can also be tried for optimization.

Table-4 Parameter setting

Sl-No.	Parameter Setting	Range of values
1	Exploration control parameter in EPO(l)	[1.5,2]
2	Exploitation control parameter in EPO(f)	[2,3]
3	Spiral movement parameter M	2
4	Population size(both in EPO and CAAO)	100
5	Maximum Iterations (both in EPO and CAAO)	100
6	SVM control parameters(C,γ)	[0.5,1]
7	Energy loss algae	[0.1]

Table-5 Comparison of Running time(in seconds) of algorithms on seven datasets

		Run time(in seconds)		
		EPO+SVM	PSO+SVM	SVM+CAAO +EPO
1	Colon	176.45	180.22	168.453
2	Ovarian	72.32	77.32	65.587
3	Leukaemia	186.11	184.56	173.84
4	Lung cancer	250.47	255.32	242.11
5	ALL-AML	188.45	192.56	185.35
6	SRBCT	178.87	186.34	172.32
7	Lymphoma	76.23	86.45	72.57

REFERENCES

1. Pal, Sankar K., Sanghamitra Bandyopadhyay, and Shubhra Sankar Ray. "Evolutionary Computation in bioinformatics: A review." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 36, no. 5 (2006): 601-615.
2. Tom M. Mitchell, McGraw-Hill, 1997, ISBN 0071154671, 9780071154673.
3. Inza, Iñaki, Pedro Larrañaga, Rosa Blanco, and Antonio J. Cerrolaza. "Filter versus wrapper gene selection approaches in DNA microarray domains." Artificial intelligence in medicine 31, no. 2 (2004): 91-103.
4. Liu, Huan, and Lei Yu. "Toward integrating feature selection algorithms for classification and clustering." IEEE Transactions on Knowledge & Data Engineering 4 (2005): 491-502.
5. Lee, Chien-Pang, and Yungho Leu. "A novel hybrid feature selection method for microarray data analysis." Applied Soft Computing 11, no. 1 (2011): 208-213.
6. Banerjee, Mohua, Sushmita Mitra, and Haider Banka. "Evolutionary rough feature selection in gene expression data." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37, no. 4 (2007): 622-632.
7. Bair, Eric, Trevor Hastie, Debashis Paul, and Robert Tibshirani. "Prediction by Supervised Principal Components. "Journal of the American Statistical Association 101, no. 473(2006): 119-137.
8. Sun, Shiquan, Qinke Peng, and Adnan Shakoor. "A kernel-based multivariate feature selection method for microarray data classification." PloS one 9, no. 7 (2014): e102541.
9. Chuang, Li-Yeh, Hsueh-Wei Chang, Chung-Jui Tu, and Cheng-Hong Yang. "Improved Binary PSO for feature selection using gene expression data." *Computational Biology and Chemistry* 32, no. 1 (2008): 29-38.
10. Oyelami, Benjamin Oyediran. "Mathematical Models and Numerical Simulation for Dynamic Evolutions of Cancer and Immune Cells." *Applied Mathematics* 9, no. 06 (2018): 561.
11. Alba, Enrique, Jose Garcia-Nieto, Laetitia Jourdan, and El-Ghazali Talbi. "Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms." In 2007 IEEE Congress on Evolutionary Computation, pp. 284-290. IEEE.
12. Pradhan, Manaswini, and R. K. Sahu. "An extensive survey on gene prediction methodologies." International Journal of Computer Science and Information Security 8, no. 7 (2010): 88-104.
13. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, Jerome Friedman Springer Science & Business Media, 2013,ISBN 0387216065, 9780387216065.
14. Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: a new learning scheme of feedforward neural networks." Neural networks 2 (2004): 985-990.
15. Cao, L., H. P. Lee, C. K. Seng, and Q. Gu. "Saliency analysis of support vector machines for gene selection in tissue classification." Neural Computing & Applications 11, no. 3-4 (2003): 244-249.
16. Katuwal, Rakesh, Ponnuthurai N. Suganthan, and Le Zhang. "An ensemble of decision trees with random vector functional link networks for multi-class classification." *Applied Soft Computing* 70 (2018): 1146-1153.

17. Okun, Oleg, and Helen Priisalu. "Random forest for gene expression-based cancer classification: overlooked issues." In *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 483-490. Springer, Berlin, Heidelberg, 2007.
18. Pattern Classification. Richard O. Duda, Peter E. Hart, David G. Stork, Edition 2, John Wiley & Sons, 2012, ISBN 111858600X, 9781118586006.
19. Ramyachitra, D., M. Sofia, and P. Manikandan. "Interval-value Based Particle Swarm Optimization algorithm for cancer-type specific gene selection and sample classification." *Genomics data* 5 (2015): 46-50.
20. Deb, Kalyanmoy, and A. Raji Reddy. "Reliable classification of two-class cancer data using evolutionary algorithms." BioSystems 72, no. 1-2 (2003): 111-129.
21. Pradhan, Pyari Mohan, and Ganapati Panda. "Solving multiobjective problems using cat swarm optimization." Expert Systems with Applications 39, no. 3 (2012): 2956-2964.
22. Alshamlan, Hala M., Ghada H. Badr, and Yousef A. Alohali. "Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification." Computational biology and chemistry 56 (2015): 49-60.
23. Review of Optimization Methods for Cancer Chemotherapy Treatment Planning Hoda Sbeity and Rafic Younes L.I.S.V, Université de Versailles, Vélizy 78140, France, 2015.
24. Baliarsingh, Santos Kumar, Weiping Ding, Swati Vipsita, and Sambit Bakshi. "A memetic algorithm using emperor penguin and social engineering optimization for medical data classification." *Applied Soft Computing* 85 (2019): 105773.
25. Vijayeeta P. Das M.N. "An Efficient Approch to Optimize the Learning Rate of Radial Basis Function Neural Network for Prediction of Metastatic Carcinoma". Computational Intelligence in Pattern Recognition pp 935-946.
26. Baliarsingh, Santos Kumar, Swati Vipsita, Khan Muhammad, and Sambit Bakshi. "Analysis of high-dimensional biomedical data using an evolutionary multi-objective emperor penguin optimizer." *Swarm and Evolutionary Computation* 48 (2019): 262-273.
27. Kingravi, Hassan A., Patricio A. Vela, and Alexandar Gray. "Reduced set KPCA for improving the training and execution speed of kernel machines." In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 441-449. Society for Industrial and Applied Mathematics, 2013.
28. Yang, Ruixin, John Tan, and Menas Kafatos. "A Pattern Selection Algorithm in Kernel PCA Applications." In *International Conference on Software and Data Technologies*, pp.374-387. Springer, Berlin, Heidelberg, 2006.
29. Baudat, Gaston, and Fatiha Anouar. "Generalized discriminant analysis using a kernel approach." *Neural computation* 12, no. 10 (2000): 2385-2404.
30. Zhu, Zexuan, Yew-Soon Ong, and Jacek M. Zurada. "Identification of full and partial class relevant genes." *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 7, no. 2 (2008): 263-277.
31. Kumar, Mukesh, and Santanu Kumar Rath. "Classification of microarray data using kernel fuzzy inference system." International scholarly research notices 2014.
32. Saberkari, Hamidreza, MousaShamsi, MahsaJoroughi, FaeghehGolabi, and Mohammad HosseinSedaaghi. "Cancer classification in microarray data using a hybrid selective independent component analysis and υ-support vector machine algorithm." Journal of medical signals and sensors 4, no. 4 (2014): 291.
33. Dhiman, Gaurav, and Vijay Kumar. "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems." *Knowledge-Based Systems* 159 (2018): 20-50.
34. Uymaz, Sait Ali, Gulay Tezel, and Esra Yel. "Artificial algae algorithm (AAA) for nonlinear global optimization." *Applied Soft Computing* 31 (2015): 153-171