Journal of Engineering and Applied Sciences
MEDWELL PUBLICATIONS

Enhanced Bio-Inspired Algorithm for Disease Diagnosis

J. Gitanjali and R. Subhashini
Vellore Institute of Technology, Vellore, India

Key words: Decision tree, feature extraction, firefly algorithm, hybridization, accuracy

Corresponding Author:
J. Gitanjali
Vellore Institute of Technology, Vellore, India

Page No.: 3024-3031
Volume: 15, Issue 16, 2020
ISSN: 1816-949x
Journal of Engineering and Applied Sciences

Abstract: Datasets are gathered for different diseases and then the feature is extracted using dimensionality reduction techniques. After the attributes are reduced, the attributes are used to test and train the data using decision tree classification algorithm techniques. The various decision tree algorithms are also used to find the accuracy for each diseases and then hybridization techniques are used to solve the problem which is then used to create upgraded yield. Metaheuristic is generally a search algorithm which solves the optimization problems that provides the best solution from the available solutions. It provides a better solution with less effort compared with other algorithms. One of the recent trend is hybrid optimization methods. Hybridization of metaheuristics are nothing but combining two bio-inspired algorithms. It improves algorithmic performance in a more efficient way to solve the problems. Hybridization techniques extracts the strengths from the combination of each algorithm.

INTRODUCTION

It is the extraction of patterns and knowledge from large datasets. In this project, the datasets used are Pima Indian Diabetes dataset and Cleveland Heart Disease Datasets. Diabetes is a disease which starts with a failure of pancreas to produce insulin or if the body will not be able to use the produced insulin. The condition in which the patient has high blood glucose level is called hyperglycemia. High cholesterol, high blood pressure and obesity are the main factors for diabetes. Heart disease have increased over the century and leads to death in many countries. Disease prediction is the most challenging task. Sometimes the doctors diagnose wrong cases of the diseases and specialists are shortage in number which leads to develop the fast and efficient detection system. The main aim of the project is to find the patterns or the features from the medical data using the classification model. Features are extracted using dimensionality reduction techniques. Those reduced features are given as input for the classifier models like decision tree. The attributes will help to predict whether the patient will get the disease or not in future. This will help the specialists to identify the depth or cause of the disease accurately. Bio-inspired techniques are used to enhance the performance evaluation[1, 2].

Disease has numerous features which will be difficult for doctors to predict it quickly. So, it is important to diagnose by using technologies, so that, the doctors can find the disease quicker and with high accuracy. There are many soft computing techniques to propose hybrid models. Hybrid models consists of two types: feature extraction and classification. Feature extraction is done by Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Classification model is done by decision tree which includes CART, ID3, C4.5 and C5.0 algorithms. Then, the performance result of these classification models are used as input to enhance

the performance evaluation by using bio-inspired algorithm. So, this will help the medical examiners to identify the accurate disease for the patients in future.

MATERIALS AND METHODS

System architecture

Proposed system: All techniques like support vector machine, ANN, Naive Bayes predicted the diseases with different accuracies and also some attributes have been reduced using feature selection method. But finding out the best minimal subset of features are needed. So, the feature extraction method is done which extracts the features by reducing the number of attributes. Classification techniques are done by decision tree algorithms by giving the input of extracted features to train and test the data. Various decision tree algorithms like ID3, CART, C4.5 and C5.0 are used to predict the accuracy (Fig. 1).

Metaheuristics algorithms are used to find the best minimal attributes. After testing the data, hybridization is done to enhance the accuracy by using bio-inspired algorithm which gives the accuracy of disease prediction. The datasets which are taken includes heart diseases, diabetes for pregnant women to predict the accuracy of diseases. Datasets are imported in the R language tool. So, enhancement of techniques are done in matlab or R language to predict the accuracy of diseases. After finding the new way, enhancement is done to predict the accuracy using bio-inspired algorithms[3, 4].

Data collection: The patient details are collected as a dataset with all features and attributes. Then, the dataset is imported into the R language tool. Pima Indian diabetes dataset is collected from the UCI repository which has two classes namely diabetic and non-diabetic. The dataset contains 9 attributes and one class with 768 instances (Table 1).

Cleveland heart diseases have 14 attributes and one class attributes with 303 instances. These datasets are also collected from UCI repository (Table 2).

Table 1: Dataset for Pima Indian diabetes

Attribute ID	Attribute name
V1	Patient ID
NPG	Number of pregnancy
PGL	Plasma glucose
DIA	Diastolic blood pressure
TSF	Triceps skin fold thickness
INS	Serum-Insulin
BMI	Body mass index
DPF	Diabetes pedigree function
AGE	Age
Diabet (class)	Diabetic or non-diabetic

Table 2: Dataset for Cleveland heart disease

Attribute name	Description
Age	Age in years
Sex	Male = 1, female = 0
CP	Chest pain type
RBp	Resting Blood pressure
Cholesterol	Serum cholesterol (mg dL^{-1})
Fasting blood	Fasting blood
Sugar	sugar>120 mg dL^{-1} true = 1 and false = 0
Resting ECG	Resting electrocardiographic results
Thalach	Maximum heart rate
Induced Angina	Does the patient experience angina as a result of exercise (value 1: yes, value 0: no)
Old peak	ST depression induced by exercise relative to rest
Slope	Slope of the peak exercise ST segment
Thal	Value 3: Normal, value 6: fixed defect, value 7: reversible defect
CA	Number of major vessels colored by fluoroscopy (value 0-3)
Concept class	Angiographic disease status

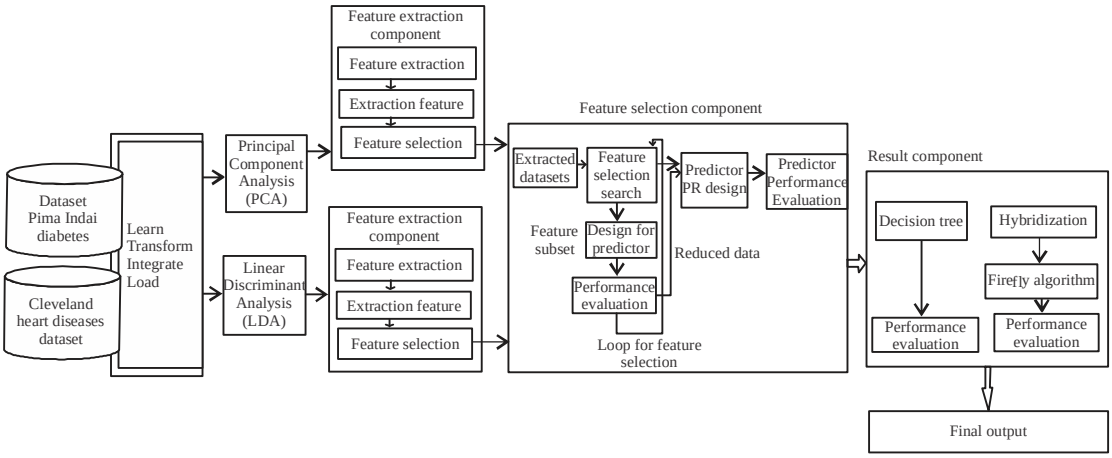

Fig. 1: System architecture

Feature extraction: The dataset are then extracted by principal component analysis and linear discriminant analysis which helps to reduce the attributes. Dimensionality reduction techniques are used to extract the features. Dimensionality reduction improves the accuracy besides saving memory and time consumption. It reduces the information content of the input data. A good dimensionality techniques keeps the high discriminative feature information and leaves the low discriminative feature information. So, the worst effect of the curse of the dimensionality are reduced after the dimensionality reduction process. Feature extraction is one of the most important methods of the dimensionality reduction process. It helps to find the linear transformations which links the original high dimensional sample space into a lower dimensional space that contains all discriminatory information (Algorithm 1).

Algorithm 1; PCA algorithm:

1. Subtract the mean
2. Calculate the covariance matrix
3. Calculate the eigenvectors and eigenvalues of the covariance matrix
4. Choose components and form a feature vector
5. Derive the new dataset

PCA algorithm: Subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. This produces a data set whose mean is zero. To find the covariance matrix, the formula used is:

$$C^{m*n} = (c_{i,j}, c_{i,j} = \text{cov}(\text{Dim}_i, \text{Dim}_j)) \quad (1)$$

It is a matrix where each entry is the result of calculating the covariance between two separate dimensions. After calculating the covariance matrix, we need to find the eigenvectors and eigenvalues, once the eigenvectors are found, then order them by eigenvalues in a descending order, i.e, highest to lowest. The number of eigenvectors that, we choose will be the number of the dimensions of the new dataset. This helps to construct feature vector. We have to form the matrix from the selected eigenvectors:

$$\text{FeatureVector} = (\text{eig}\_1, \text{eig}\_2, \ldots, \text{eig}\_n) \quad (2)$$

To derive the new dataset, take the transpose of the FeatureVector and multiply it on the left of the original dataset, transposed. So, the formula is:

$$\text{FinalData} = \text{RowFeaturevector} * \text{RowDataAdjusted} \quad (3)$$

We have a data matrix on n observations on p correlated value $x_1, x_2, \ldots, x_p$. PCA looks for a transformation of the x_i into p new variables y_i that are uncorrelated. For a transformation of the data matrix X(n×p) such that:

$$Y = \delta^T X = \delta_1 X_1 + \delta_2 X_2 +, \ldots, + \delta_p X_p \quad (4)$$

where, $\delta = (\delta_1, \delta_2, \ldots, \delta_p)^T$ is a column vector with:

$$\delta_1^2 + \delta_2^2 +, \ldots, + \delta_p^2 = 1 \quad (5)$$

Maximize the variance of the projection of the observations on the Y variables. Find δ, so that:

$$\text{Var}(\delta^T X) = \delta^T \text{Var}(X) \delta \text{ is maximum} \quad (6)$$

The matrix C = Var(X) is the covariance of the matrix X_i variables. The direction of δ is given by the eigenvector λ_1 corresponding to the largest eigenvalue of the matrix C. The second vector that is orthogonal or uncorrelated to the first is the one that has the second highest variance which comes to be the eigenvector corresponding to the second eigenvalue. New variables Y_i that are linear combination of the original variables (x_i):

$$Y_i = a_{i1} x_1 + a_{i2} x_2 +, \ldots, a_{ip} x_p; \; i = 1..p \quad (7)$$

The new variables Y_i are derived in decreasing order of importance; they are called principal components. The eigenvalues λ_i are found by using the formula:

$$\det(C - \lambda I) = 0 \quad (8)$$

Eigenvectors are columns of the matrix A (new variables in the dataset):

$$C = A \, D \, A^T \quad (9)$$

If we multiply one variable by a scalar you get different results. This is because it uses covariance matrix (and not correlation). PCA should be applied on data that have approximately the same scale in each variable. The new variables (PCs) have a variance equal to their corresponding eigenvalue:

$$\text{Var}(Y_i) = \lambda_i \text{ for all } i = 1, \ldots, p \quad (10)$$

Small $\lambda_i \Leftrightarrow$small variance$\Leftrightarrow$data change little in the direction of component Y_i. The relative variance explained by each PC is given by $\lambda_i / \Sigma\lambda_i$.

Enough principal components to have a cumulative variance explained by the PCs that is>50-70%. Kaiser criterion implies to keep principal components with eigenvalues>1. This prediction gives the new dataset with reduced variables of principal components (Fig. 2).

In R language, PCA implementation is done by choosing princomp method from the stats package.

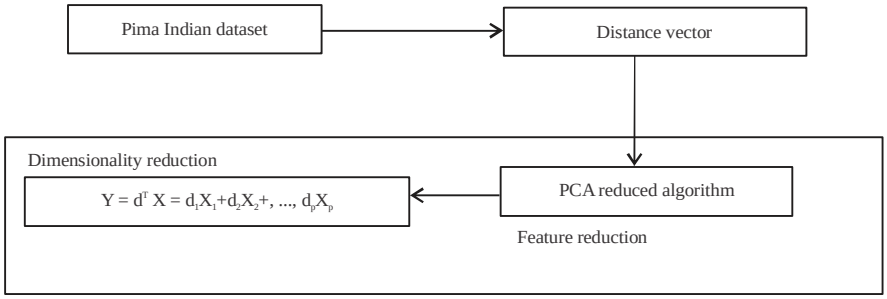

Fig. 2: Elaborated architecture of feature extraction

Prcomp is a generic method with formula and default methods from stats package which performs a PCA on the given numeric data matrix:

- prcomp(x, …)
- prcomp(formula, data = NULL, scale. = T, ….)
- predict(object, newdata, …)

Linear Discriminant Analysis (LDA): The best way to do pattern recognition is to first estimate Gaussian for all classes and construct the quadratic discriminant function by using the estimated density function to specify the decision boundaries. It has been proved that the training sample pattern is linearly related to the square of dimensionality of a feature space for a quadratic classifier. Since, the dimensionality of the sample space is large compared to the number of training sample pattern, it has been impossible to obtain a acceptable recognition rates by utilizing density estimation procedure. The easy way to solve this kind of problem is to assume that all classes have Gaussian distribution with identical covariance structure. Since, the discriminant functions used here is linear, the Linear discriminant analysis technique seeks projection that they maximize the between-class separability and minimize the within-class variability. By applying this approach, we find projection directions that one has maximize the distance between the samples of different class and on other minimize the distance between the samples of same class. However, the linear discriminant function can produce acceptable result even when the covariance structure are different (Algorithm 2).

Algorithm 2; LDA algorithm:

1. Compute the d-dimensional mean vectors for the different classes from the dataset
2. Compute the scatter matrices (in-between-class and within-class scatter matrix)
3. Compute the eigenvectors (e1, e2, ..., ed) and corresponding eigenvalues (λ1, λ2, ..., λd) for the scatter matrices
4. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d×k dimensional matrix W (where every column represents an eigenvector)
5. Use this d×k eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: Y = X×W (where, X is a n×d-dimensional matrix representing the n samples and y are the transformed n×k-dimensional samples in the new subspace)

LDA algorithm: The method used in the projection vectors satisfies the orthogonality constraints. After the feature extraction step, the data samples in the transformed space will be uncorrelated. Consider a set of observations x also called features, attributes, variables or measurements for each sample of an object or event with known class $y \in \{0, 1\}$. This set of samples is called the training set. The classification problem is then to find a good predictor for the class y of any sample of the same distribution (not necessarily from the training set) given only an observation x.

LDA approaches the problem by assuming that the conditional probability density functions p(x|y = 1) and p(x|y = 2) are both normally distributed with mean and covariance parameters (μ1, Σ1) and (μ2, S2), respectively. Under this assumption, the Bayes optimal solution is to predict points as being from the second class if the log of the likelihood ratios is below some threshold T, so that:

$$(x-\mu_1)^T \Sigma_1^{-}(x-\mu_1)+\ln|\Sigma_2|-(x-\mu_2)^T \Sigma_2^{-}(x-\mu_2)-\ln|\Sigma|<T \quad (11)$$

Without any further assumptions, the resulting classifier is referred to as QDA (Quadratic Discriminant Analysis). When Σ1 = Σ2 = Σ (homoscedasticity assumption) the discriminant functions it is:

$$(x-\mu_1)^T \Sigma_1^{-}(x-\mu_1)-(x-\mu_2)^T \Sigma^{-}(x-\mu_2)<T \quad (12)$$

this is the expression for the LDA (Linear Discriminant Analysis) function. In this case, several terms cancel and the above decision criterion it is linear:

$$w.x>c \quad (13)$$

for some threshold constant c:

$$\vec{w} = \Sigma^{-1}(\vec{\mu}_1-\vec{\mu}_0)$$
$$c = \frac{1}{2}(T-\vec{\mu}_0^{\,T} \Sigma_0^{-1}\vec{\mu}_0+\vec{\mu}_1^{\,T} \Sigma_1^{-1}\vec{\mu}_1) \quad (14)$$

Extracted classifier decision tree: The extracted features after reduction are used to test and train the data using

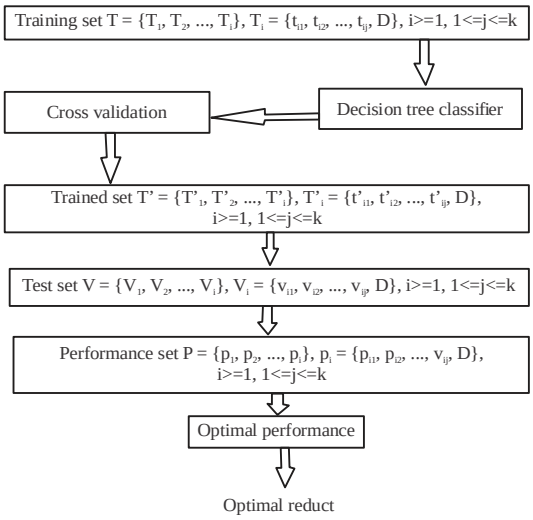

Fig. 3: Elaborated architecture of decision tree

decision tree classifiers. The decision tree classifiers used are CART, ID3, C4.5 and C5.0 algorithms (Fig. 3 and Algorithm 3).

Algorithm 3; Classification and regression algorithm (CART); CART algorithm:

1. For each non-terminal node
   For a variable
   - At all its split points, splits samples into two binary nodes
   - Select the best split in the variable in terms of the reduction in impurity (gini index)
- Rank all of the best splits and select the variable that achieves the highest purity at root
- Assign classes to the nodes according to a rule that minimizes misclassification costs
- Grow a very large tree T_{max} until all terminal nodes are either small or pure or contain identical measurement vectors

2. Prune and choose final tree using the cross validation

Recursive partitioning:

- Pick one of the predictor variables, x_i
- Pick a value of x_i, say s_i, that divides the training data into two (not necessarily equal) portions
- Measure how "pure" or homogeneous each of the resulting portions are
- "Pure" = containing records of mostly one class
- Algorithm tries different values of x_i and s_i to maximize purity in initial split

After you get a "maximum purity" split, repeat the process for a second split and so on.

In splitting rules: Select the variable value ($X = t_1$) that produces the greatest "separation" in the target variable. Regression tree for continuous target variable which most intuitively appropriate method for loss ratio analysis. It use sum of squared errors. It finds split that produces greatest separation in:

$$\Sigma[y-E(y)]^2 \quad (15)$$

i.e., find nodes with minimal within variance and therefore, greatest between variance like credibility theory. Every record in a node is assigned the same way. This model is a step function.

Statistical deviance generally encapsulates the difference between two probability measures. In CART, deviance measures the heterogeneity (e.g., misclassification, variability) at each node:

$$D_i = 2\sum_k n_{ik} \log(p_{ik}) \quad (16)$$

where, i is index for terminal nodes, j = 1, ..., n is index for data, k indexes the classes in each leaf. Classification Trees for discrete or categorical target variable. In contrast with regression trees, various measures of purity are used. Common measures of purity are Gini measure, entropy, "twoing" splitting criteria:

$$D_i = 2\sum_k n_{ik} \log(p_{ik}) \quad (17)$$

Gini purity of a node:

$$p(1-p) \quad (18)$$

where, p = relative frequency of defectors. Gini might produce small but pure nodes.

Entropy of a node:

$$-\Sigma p\log p-\left[p*\log(p)+(1-p)*\log(1-p)\right] \quad (19)$$

Max entropy/Gini when p = 0.5. Min entropy/Gini when p = 0 or 1. The "twoing" rule strikes a balance between purity and creating roughly equal-sized nodes. It uses Cross-Validation (CV) to select the optimal decision tree. To grow the tree as far as it can. CV tells when to stop pruning.

K-fold cross validation: A technique which assesses the predictive value of a model (tree) for new data:

- Split the data into k (say 10) parts
- Withhold one part (validation set), grow the tree using other 9 parts (training set)
- Assess predictive accuracy on the validation part using the tree
- Repeat, holding all 10 parts out in turn

Too big will overfit data. Too small might miss important structures. Generally cost-complexity pruning can be used:

- Cost-complexity pruning
- It makes a big tree
- Consider all subtrees which can be achieved by pruning the big tree

$$R_\alpha = MC+\alpha L \quad (20)$$

Where:

MC = Misclassification rate, Relative to misclassifications in root node

L = Leaves (terminal nodes)

Let T_0 be the biggest tree. Find sub-tree of T_α of T_0 that minimizes R_α. Let's sequentially collapse nodes that result in the smallest change in purity. This gives us a nested sequence of trees that are all sub-trees of T_0:

$$T_0 \text{»} T_1 \text{»} T_2 \text{»} T_3 \text{»}, \ldots, \text{»} T_k \text{»}, \ldots$$

The sub-tree T_α of T_0 that minimizes R_α is in this sequence. Gives us a simple strategy for finding best tree. Find the tree in the above sequence that minimizes CV misclassification rate. Note that a is a free parameter in $R_\alpha = MC+\alpha L$ 1:1 correspondence between α and size of tree:

- α = 0 implies maximum tree T_0 is best
- α = big implies you never get past the root node

Use cross-validation to select optimal α (size).

ID3 algorithm: ID3 is Iterative Dichotomizer 3 was invented by Ross Quinlan in 1979. ID3 is an algorithm used to generate a decision tree from a dataset. It is the precursor to the C4.5 algorithm. By using ID3 and other machine-learning algorithms from artificial intelligence, expert systems can engage in tasks usually done by human experts such as doctors diagnosing diseases by examining various symptoms (the attributes) of patients (the data instances) in a complex decision tree. The input data of ID3 is known as sets of "training" or "learning" data instances which will be used by the algorithm to generate the decision tree (Algorithm 4).

Algorithm 4; ID3 algorithm:

1) Establish classification attribute for table R
2) Compute classification entropy
3) For each attribute in R, calculate information gain using classification attribute
4) Select attribute with the highest gain to be the next node in the tree (starting from the root node)
5) Remove node attribute, creating reduced table RS
6) Repeat steps 3-5 until all attributes have been used or the same classification value remains for all rows in the reduced table

ID3 algorithm: This algorithm generates decision tree using shannon entropy. Entropy is a measure of how certain or uncertain the value of random variables is (or will be). Varying degrees of randomness, depending on the number of possible values and the total size of the set. Entropy quantifies randomness. Lower value refers to less uncertainty and higher value refers to more uncertainty. Thus, Shannon entropy is:

$$H(S) = -\sum p(x) \log_2 \left(1/p(x)\right) \quad (21)$$

Information gain uses Shannon entropy: IG calculates effective change in entropy after making a decision based on the value of an attribute. For decision trees, it's ideal to base decisions on the attribute that provides the largest change in entropy, the attribute with the highest gain. For Set S, attribute A where S is split into subsets based on values of A. The information gain is based on the decrease in entropy after a dataset is split on an attribute. First the entropy of the total dataset is calculated. The dataset is then split on the different attributes. The entropy for each branch is calculated. Then, it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the information gain or decrease in entropy. The attribute that yields the largest IG is chosen for the decision node. A branch set with entropy of 0 is a leaf node. Otherwise, the branch needs further splitting to classify its dataset. The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified:

$$IG(A, S) = H(S) - \sum p(t)\, H(t) \quad (22)$$

C4.5 algorithm: An improvement over ID3 algorithm which is designed to handle noisy data better, missing data, pre and post pruning of decision trees, attributes with continuous values and rule derivation. C4.5 builds decision trees from the set of training data like ID3 using information entropy. It chooses the attributes that mostly splits the samples into substes based on the normalized information gain which is difference in entropy. The higher gain which the attribute has the right to make the decision (Algorithm 5).

Algorithm 5; C4.5 algorithm:

- All the samples belongs to the same class, it simply creates the leaf node for the decision tree to ask to choose that class
- If no features provides information gain, C4.5 creates a node higher up the tree using the expected value of the class
- For each attribute a, find the normalized information gain ratio from splitting on a
- a_best be the attribute with highest gain
- Create a decision node that splits on a_best
- Recur on the sublists and those nodes as children of nodes

Hydridization: Hybridization of decision mining techniques are used to analyse the disease and then it is

used to invoke a new way of procedure. Hybridization in general means increasing the performance and decreasing the error rate. Hybridization is popular because:

- Hybridization is done which is not restricted to the use of different combinations of metaheuristics which allows to combine the use of hybrid algorithms like local search and metaheuristics
- Different combinations of metaheuristics and different research areas which leads to many new approaches which combines fuzzy logic and many optimization methods
- So, hybridization which takes advantage from each and every algorithm to improve the performance for more effective and efficient problem-solving
- Data mining methods are combined with metaheuristics in order to find the solutions which guide the heuristic search for better cost solutions and computational time required is less
- This hybridization method solves several optimization problems

Enhancement of hybridization using bio-inspired algorithm: After finding the new way, enhancement is done to predict the accuracy using bio-inspired algorithms. Enhancement is done by using firefly algorithm. Firefly algorithm was developed by Xin-She Yang in 2008. It is the global optimization algorithm that aims to improve the performance of firefly based on the flashing patterns and behaviour of fireflies. It is also based on the light intensity[5, 6]. The flash of the firefly act as a signal system to attract other fireflies. The firefly algorithm uses three rules:

- All fireflies are unisex, so that, an individual firefly will be attracted towards other fireflies
- The attractiveness is proportional to the brightness, so that, the less brighter fly will attract towards the brighter fly. However, both brightness of the flies decreases as the distance between the two increases
- It will move randomly when there are no brighter flies than the given firefly

The brightness of the firefly is determined by the landscape of the objective function. It makes to propose new optimization algorithms when the flashing light is associated with objective function to be optimized. In attribute reduction problem, binary number 0 and 1 are used to predict whether the attribute is selected or not selected. In the firefly algorithm, firefly i with lower intensity moving towards firefly j with higher light intensity (Algorithm 6).

Algorithm 6; Firefly algorithm:

1. begin
2. Objective function f(y), $y = (y_1, \ldots, y_d)T$
3. Generate initial population of fireflies y_i (i = 1, 2, …, n)
4. Light Intensity I at y_i is determined by $f(y_i)$
5. Define light absorption coefficient γ
6. While (t<MaxGeneration)
7. for i = 1: n all n fireflies
8. for j = 1: I all n fireflies
   a. if (I_j>I_i)
   b. Move firefly I towards j in d-dimension via. L'evy flights
   c. end if
   d. Attractiveness varies with distance r via. $\exp[-\gamma r^2]$
   e. Evaluate new solutions and update light intensity
9. end for j
10. end for i
11. Rank the firelies and find the current best
12. end while
13. Post-process results and visualization
14. end

RESULTS AND DISCUSSION

Different medical datasets like Pima Indian Diabetes and Cleveland Heart Disease Datasets are used for experimental work. This research is implemented in R language. Pima datasets includes 768 records with 9 attributes and one class attribute. Cleveland datasets includes 303 records with 13 attributes and 1 class attribute (Table 3-12).

Table 3: Description of datasets used for the experiment

Name of dataset	No. of instances	No. of attributes	No. of classes
Pima Indian diabetes	768	10	2
Cleveland heart diseases	303	15	5

Table 4: Results after using attribute reduction method

Name of dataset	No. of attributes without reduction	Reduced No. of attributes by PCA
Pima Indian diabetes	10	3
Cleveland heart diseases	15	4

Table 5: Confusion matrix

Actual vs. predicted	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Table 6: Confusion matrix and statistics for CART algorithm

Prediction	No	Yes
No	186	51
Yes	64	83

Table 7: Confusion matrix and statistics for ID3 algorithm

Prediction	No	Yes
No	181	71
Yes	67	65

Table 8: Confusion matrix and statistics for C4.5 algorithm

Prediction	No	Yes
No	190	13
Yes	66	39

Table 9: Confusion matrix and statistics for C5.0 algorithm

Prediction	No	Yes
No	181	43
Yes	70	90

Table 10: Results after using decision tree algorithms

Variables	CART	ID3	C4.5	C5.0
Accuracy	0.7214	0.7031	0.7143	0.7344
95% CI	0.652, 0.7459	0.5904, 0.6887	0.6909,0.7913	0.6574,0.7509
No information rate	0.651	0.6458	0.8312	0.6536
p-value (Acc>NIR)	0.02284	0.6070	1	0.01733
Kappa	0.3554	0.2092	0.35	0.3797
Mcnemar's test p-value	0.26314	0.7984	4.902e-09	0.01445
Sensitivity	0.7835	0.9150	0.7088	0.7804
Specificity	0.6000	0.3212	0.7347	0.6434
Pos pred value	0.7848	0.7183	0.9360	0.8080
Neg pred value	0.5646	0.4924	0.3714	0.5625
Prevalence	0.6510	0.6458	0.8312	0.6536

Table 11: Performance comparison of proposed model

Performance evaluation parameters	Pima Indian diabetes				Cleveland heart disease			
	CART	ID3	C4.5	C5.0	CART	ID3	C4.5	C5.0
Accuracy	72.14	70.31	71.43	73.44	66.45	61.84	61.54	64.47
Sensitivity	78.35	91.50	70.88	78.04	95.29	96.25	86.79	89.16
Specificity	60.00	32.12	73.47	64.34	70.15	77.78	76.32	81.16

Table 12: Comparison of predictive accuracies

References	Approach	Predictive accuracy (%)
Statlog	Logdisc	77.70
Weka	Logistic	77.08
Ster and Dobnikar	QDA	59.50
Weka	Naive Bayes	76.04
Polar, etc.	LSSVM	78.21
Statlog	BP	75.20

Performance measures: Three evaluation measures are used to provide the comparison among different decision tree algorithms. They are accuracy, sensitivity and specificity. These measures are calculated using True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Formula to calculate the performance measures using this confusion matrix are as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \times 100\%$$

$$\text{Specificity} = \frac{TN}{TN+FP} \times 100\%$$

- Confusion matrix
- Performance features of decision tree algorithms for Pima Indian diabetes dataset
- Performance comparison of proposed model
- Comparison of predictive accuracies of other classifiers for Pima Indian dataset and heart diseases

CONCLUSION

In this project, we are using firefly algorithm for enhanced accuracy of disease detection. Using firefly algorithm, we are using to enhance the accuracy based on the inputs of the decision tree accuracy.

REFERENCES

01. Dangare, C.S. and S.S. Apte, 2012. Improved study of heart disease prediction system using data mining classification techniques. Intl. J. Comput. Appl., 47: 44-48.
02. Chitra, S. and G. Balakrishnan, 2012. A survey of face recognition on feature extraction process of dimensionality reduction techniques. J. Theor. Appl. Inf. Technol., 36: 92-100.
03. Quinlan J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Burlington, Massachusetts, USA., ISBN:1-55860-238-0, Pages: 267.
04. Valdes, J.J., 2002. Similarity-based heterogeneous neurons in the context of general observational models. Neural Netw. World, 5: 499-508.
05. Wang, Z., G. Yu, Y. Kang, Y. Zhao and Q. Qu, 2014. Breast tumor detection in digital mammography based on extreme learning machine. Neurocomputing, 128: 175-184.
06. Obenshain, M.K., 2004. Application of data mining techniques to healthcare data. Infect. Control Hosp. Epidemiol., 25: 690-695.