INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Accurate liver disease prediction system using convolutional neural network

K Jeyalakshmi[1]*, **R Rangaraj**[2]

1 Associate Professor, Department of Computer Science, Hindusthan College of Arts and Science, Coimbatore, Tamil Nadu, India
2 Professor and Head, Department of Computer Science, Hindusthan College of Arts and Science, Coimbatore, Tamil Nadu, India

OPEN ACCESS

Received: 15.03.2021
Accepted: 28.04.2021
Published: 17.05.2021

Citation: Jeyalakshmi K, Rangaraj R (2021) Accurate liver disease prediction system using convolutional neural network. Indian Journal of Science and Technology 14(17): 1406-1421. https://doi.org/10.17485/IJST/v14i17.451

* **Corresponding author**.

jeyas1201@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To introduce the technique which can ensure the accurate and reliable prediction of liver disease by adapting the deep learning technique. **Methods:** In this work Modified Convolutional Neural Network based Liver Disease Prediction System (MCNN-LDPS) is introduced for the accurate liver disease prediction outcome. In the proposed research work, Dimensionality reduction is carried out using Modified Principal Component Analysis. Optimal feature selection is carried out using Score based Artificial Fish Swarm Algorithm (SAFSA). In SAFSA algorithm, information gain and entropy values are taken as input values which proved accurate outcome. This research method has been analysed over Indian Liver patient dataset. **Findings:** The analysis of the research work proves that the proposed method MCNN-LDPS obtains better outcome in terms of increased accuracy, precision. Here comparison analysis proved that MCNN-LDPS obtains 4.05% increased accuracy, 21.23% F-measure, 4.22% precision and 34.26% recall. This research method has been compared with the existing Multi layer Perceptron Neural Network (MLPNN) for the performance analysis. **Novelty:** The major limitation of CNN is its inability to encode Orientational and relative spatial relationships, view angle. CNN do not encode the position and orientation of data. Lack of ability to be spatially invariant to the input data sample. This is resolved in this research work by combining the genetic algorithm with the CNN method.

Keywords: Liver Disease Prediction; Feature Selection; Information Gain; Entropy; Convolutional neural network; Dimensionality Reduction

1 Introduction

Liver disease prediction is the most concentrated research issue in various medical organization and industries. Hepatic disorder needs to be predicted immediately to ensure the timely treatment. However automated and faster prediction of liver disease presence is more difficult task, especially with the incomplete patient data. In[(1)] proposed the data classification is based on liver disorder. The training dataset is developed by collecting data from UCI repository consists of 345 instances with 7 different attributes. This paper deals with results in the field of data classification

obtained with Naïve Bayes algorithms. FT tree algorithms, and KStar algorithms and on the whole performance made know FT Tree algorithm when tested on liver disease datasets, time taken to run the data for result is fast when compare to other algorithm with accuracy of 97.10%. Based on the experimental results the classification accuracy is found to be better using FT Tree algorithm compare to other algorithms. However this algorithm doesn't perform well on high scale data with more noisy features.

In [(2)] proposed to identify if the patients have the liver disease based on the 10 important attributes of liver disease using a Decision Tree, Naive Bayes, and NB Tree algorithms. The result shows NB Tree algorithm has the highest accuracy; however, the Naïve Bayes algorithm gives the fastest computation time. For future study, the performance of NB Tree algorithm will be the target of improvement of the accuracy by finding the most significant factor in identifying liver disease patients. This work only utilizes the standard algorithms for the liver disease prediction which cannot perform well on high dimensional data.

In [(3)] compared Naïve Bayes and FT tree algorithm and concluded that the accuracy of Naïve Bayes algorithm is much better than the other algorithms. However, this research methodology tends to have more computational overhead and didn't focus on risk factors.

In [(4)] applied the data mining techniques, such as KNN, SVM, MLP or decision trees over a unique dataset, which is collected from 16,380 analysis results for a year. This study can be useful for reducing the number of analysis, since the prediction can be correlated and furthermore the correlation can be utilized for detecting the anomaly on the analysis. However, this research methodology tends to have lesser accuracy value while processing incomplete patient information.

In [(5)] described the categorization of liver disorder through feature selection and fuzzy K-means classification. Accordingly, various liver disorders also share same attribute values and it needs more effort to classify liver disorder type correctly with basic attributes. So Fuzzy based classification gives better performance in these confusing classes and achieved above 94 percentage accuracy for each type of liver disorder. Their methodology tends to consume more processing cycles and computational time for accurate liver disease prediction.

In [(6)] analysed the data of liver diseases using particle swarm optimization algorithm (PSO) with K Star classification. The proposed algorithm enhanced the performance of accuracy when compared to existing classification algorithms. Accordingly, the PSO-KStar algorithm is considered good data mining algorithm with respect to understandability, transformability and with accuracy of 100%. This research methodology structure is more complex to understand and doesn't support real time dataset with more number of attributes.

In [(7)] predicted liver diseases using classification algorithms such as Naïve Bayes and support vector machine. Comparison was done based on the performance factors classification accuracy and execution time. They concluded that the SVM classifier is considered as the best classification algorithm because of its highest classification accuracy values. On the other hand, while comparing the execution time, the Naïve Bayes classifier needs minimum execution time; The computation complexity of the proposed research methodology is higher and also this methodology prone to error fitting.

In [(8)] proposed back propagation neural network and radial basis function neural network are designed to diagnose these diseases. The algorithms were compared with the c4.5, CART, Naïve Bayes, Support Vector Machine (SVM) and concluded that the radial basis function neural network is the optimal model because it has a recognition rate of 70% which has proved more accurate and efficient than the other algorithms. However, this research technique cannot handle the continuous values and missing values in the dataset, efficiently.

In [(9)] focused on the aspect of Medical diagnosis by learning pattern through the collected data of Liver disorder to develop intelligent medical decision support systems . They employed several classification (J.48, SVM, Random Forest, etc) algorithms. The predictive performances of popular classifiers were compared quantitatively. By analyzing the results, Multilayer perceptron gives the overall best classification result with the accuracy 71.59% than other classifiers. This research technique requires specifying the decision function to ensure the increased accuracy rate.

In [(9)] made hybrid model construction and comparative analysis for improving prediction accuracy of liver patients in three phases. In first phase, classification algorithms are applied on the original liver patient datasets collected from UCI repository. In second phase, by the use of feature selection, a subset (data) of liver patient from whole liver patient datasets is obtained which comprises only significant attributes and then applying selected classification algorithms on obtained, significant subset of attributes. SVM algorithm is considered as the better performance algorithm, because it gives higher accuracy in respective to other classification algorithms before applying feature selection. But, Random Forest algorithm is considered as the better performance algorithm after applying feature selection. In third phase, the results of classification algorithms with and without feature selection are compared with each other. The results obtained from our experiments indicate that Random Forest algorithm outperformed all other techniques with the help of feature selection with an accuracy of 71.8696%. This research technique failed to perform well on large volume of data with more noises.

The main goal of our research work is to introduce the automated system which can predict the liver disease in the accurate way. This is done with the concern of the noises and missing terms present in the collected dataset. This study attempts to predict the liver disease present in the patient by analysing the given input dataset. This is done by introducing the Modified Convolutional neural network based on liver disease prediction system which can automatically detect the liver disease from the given input dataset. This research work also attempts to handle the large volume of dataset with more irrelevant terms by adapting the Dimensionality reduction technique. This research work intends to reduce the computation overhead of the classification process by selecting the more optimal features from the input dataset.

2 Automated liver disease prediction system

In the proposed research work, Dimensionality reduction is carried out using Modified Principal component analysis as a preprocessing step. After the pre-processing step, the optimal features are selected using Score based Artificial Fish Swarm Algorithm (SAFSA). Finally, Modified Convolutional neural network is used for classification of the dataset. The overall flow of the proposed research work is shown in Figure 1.

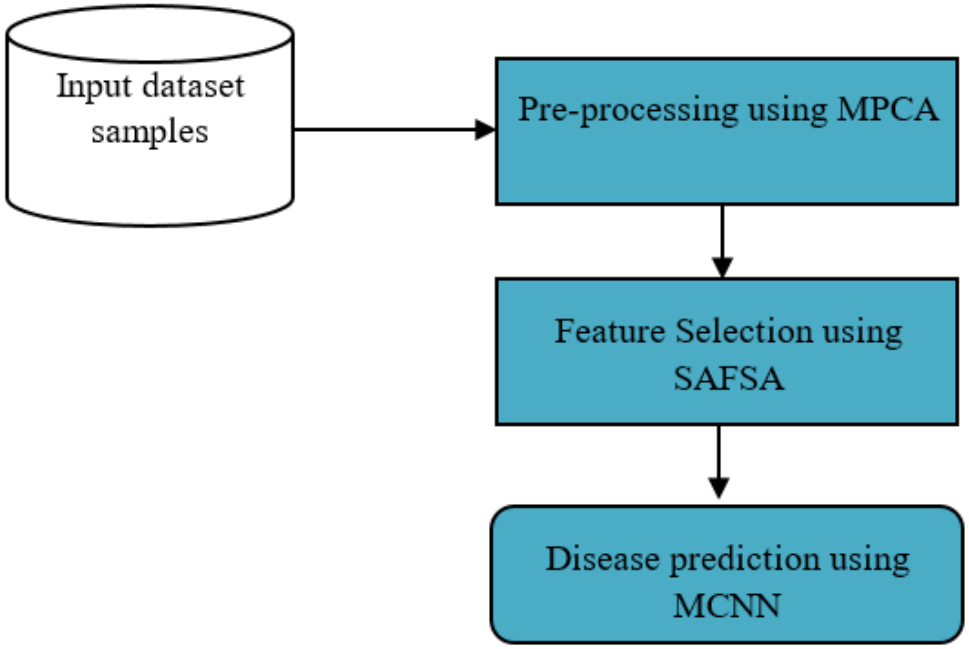

Fig 1. Architecture view representation of the contribution

The analysis of the research work is carried out on the Indian patient liver dataset which is described in the following sub section. From the Figure 1, it is learned that, initially preprocessing is carried out on Indian patient liver dataset. After preprocessing feature selection is performed which will then be classified using the MCNN algorithm. Here training is done based on 10 cross validation procedure where input data will be divided into 10 equal parts. Here 9 parts of input data will be considered for the training process and the remaining 1 part will be used for the testing process. The detailed discussion of the proposed research technologies are given in the following sub sections.

2.1 Dataset description

Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. This dataset was used to evaluate prediction algorithms in an effort to reduce burden on doctors. This data set contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The ”Dataset” column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age ”90”. The size of the dataset is 22.8 KB.

2.2 Dimensionality reduction using modified principal component analysis

The liver disease dataset might consist of most noisy features and the more irrelevant features. This might increase the computation overhead of the classifier. This can be avoided by the pre-processing step over the input dataset. In this work, Dimensionality reduction is carried out by using Modified Principal component analysis.

PCA is a classical multivariate data analysis method that is useful in linear feature extraction and data compression[10] [11]. The approach has been applied in many fields of information processing to extract useful important features for data compressing and classification due to its error minimizing and de-correlating properties. Indicating the spectral data (original image) as the matrix: X= $[x_{ik}]_{m\times n}$, where m is the number of the origional spectral bands and n is the number of pixels in whole scene. So each line in this matrix stands for one band of the original bands. In general, the linearly transform (PCA) can be expressed as following equation (1):

$$Y = TX \tag{1}$$

where T is the transform matrix, X is the original vectors and Y is the transformed vectors. In order to solve the transform matrix T, the following equation (2):

$$(\lambda I - S)U = 0 \tag{2}$$

where the matrices I , S, U and λ are the square matrix with unity along its diagonal, the covariance matrix of original images, the eigenvectors and the eigen values. U_j and λ_j(j =1,2,...,m) can be computed through the equation (2), with the eigen values ordered as $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_m$. The eigenvectors U can be expressed as U = $[U_1, U_2,....,U_m] = [u_{ij}]_{mxn}$, where U satisfies with the equation: $U^TU = UU^T$=I. The matrix T can be determined by inversing the matrix U.

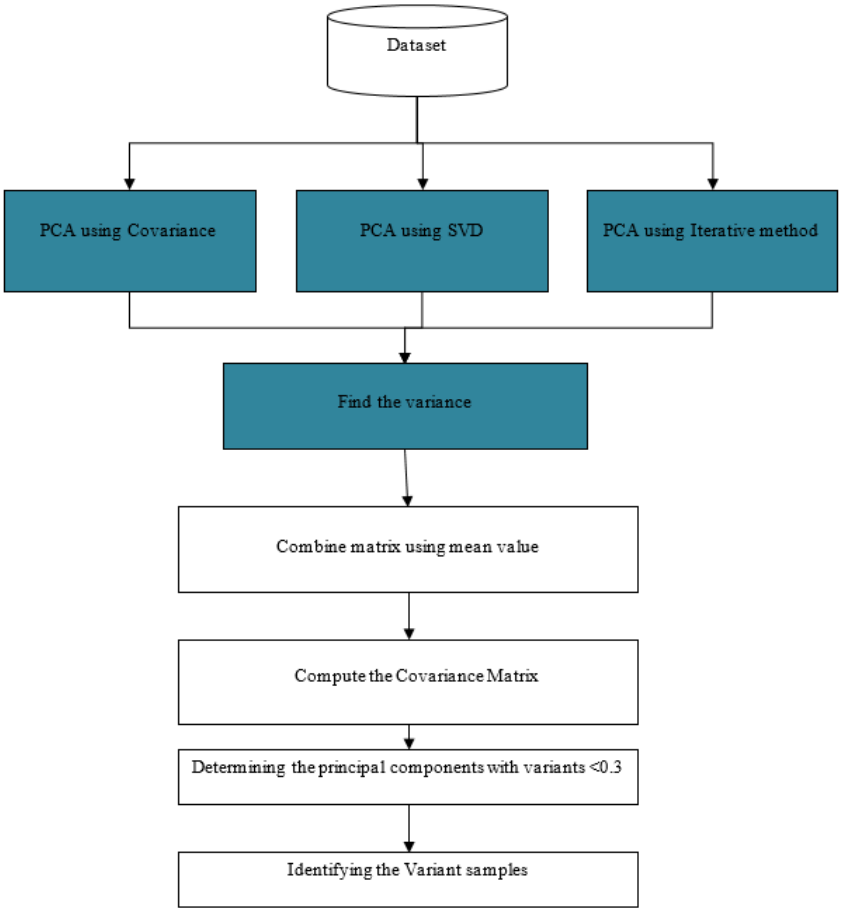

Fig 2. Dimensionality reduction using Modified PCA

Previous studies have demonstrated that PCA is effective in data compression for all classes within the imaged area. In most image processing applications, it is better to deal with a fewer classes and some classes present in the image may be neglected. The PCA method cannot guarantee that the information related to the relevant classes is effectively compressed. The major limitations of PCA are [12].

- Standard PCA struggles with Big Data when we need out-of-core computation.
- Standard PCA can detect only linear relationships between variables/features.
- The transformed data we generate after applying PCA should ideally be sparse. Thus, standard PCA always generates dense expressions in certain datasets.

The above mentioned limitations are solved in the proposed Modified PCA. In MPCA, instead of linear assumption, three matrices are constructed with the help of covariance, SVD and iterative methods. In the new method MPCA, training samples, which are relevant for a given application, were selected from a scene, and the transformed matrix T' was obtained from these training samples.

$$Y' = T'X \tag{3}$$

Comparing the two equations (2) and (3), the difference lies in the transform matrix, and essentially lies in the samples for calculating the covariance matrix, one is from training samples, the other is from the whole image sample. The above steps describe the basic steps of PCA where it is modified by constructing the three PCA based on covariance, SVD and Iterative method. From these modifications in PCA, dimensionality reduction is carried out more effectively. The detailed procedure of MPCA including the pseudo code of MPCA is given below:

Input: Data samples in matrix format M, Number of epoch e, Number of dimension d, initial vector q_1

Output: Reduced data

1. Construct PCA using Covariance
   For t=1,2,..,e do
   v = Mq_t //t^{th} vector generation
   where M→Datamatrix
   For i = 1,2,..., t do
   $h_{it} = q_i^T$ v
   v = v-$h_{it}q_i$//Orthogonalization into ith vector
   End for
   $H_{t+1,t}$ = ||v||
   q_{t+1} = v/H_{t+1},t //Normalization
   End for
   [Y, Σ, Y^T] = evd (H_t)
   W 1= Q_t Y
2. PCA using SVD (Singular Value Decomposition)
   Generate n x k matrix G
   Compute Y = MG
   Compute an orthogonal column basis Q of Y
   Form B= Q^TM
   Compute eigen decomposition of $BB^T = X\Sigma^2X^T$ W2= QX
3. PCA using iterative method
   rv = a random vector of length p
   rv = rv / |rv|
   Repeat:
   S=0(a vector of length p)
   For each row x belongs to X
   s= s + (x.r)x
   eigenvalue W = r^Ts
   error = |eigenvalue.r-s|
   r = s/|s|
   exit if error <tolerance
   return eigenvalue W3, r

4. Calculate variance extracted from the 3 PCA
5. Combine the variance of PCA using mean function
6. If $\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x_{approx}^{(i)}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2} < 0.3$
   Eliminate as irrelevant data
7. End if
8. End

The variances extracted from the 3 PCA are combined using mean function. Average values of the 3 PCA values are checked using a threshold value of 0.3. The samples which are having the coefficients less than 0.3 is eliminated. Out of 583 samples 578 samples are selected for next feature selection. The implemented result sample of MPCA is shown in Figure 3.

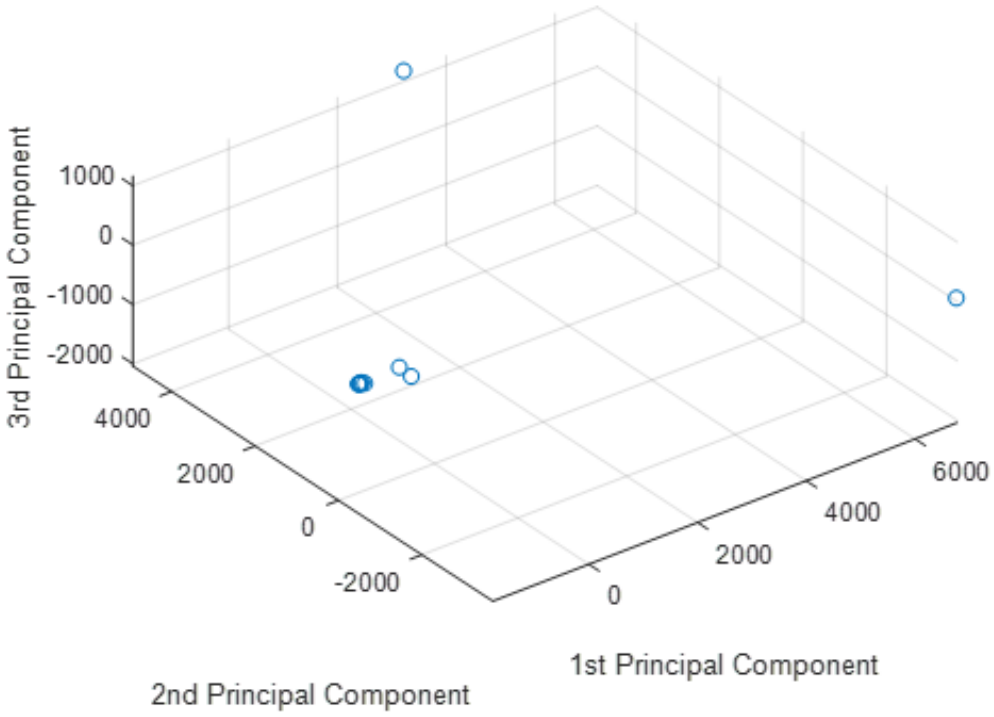

Fig 3. MPCA Implemented Screen shot

2.3 Feature selection using score based artificial fish swarm algorithm

After preprocessing, it is required to select the most relevant features from the dimensionality reduced data samples, in order to obtain the most accurate and reliable outcome. This optimal feature selection can be done by introducing the optimization algorithm which can select the most optimal features from the given input dataset. In this work, optimal feature selection is done by Score based Artificial Fish Swarm Algorithm (SAFSA). Here, information gain and entropy values are taken as fitness values.

In general AFSA (artificial fish-swarm algorithm) is one of the best methods of optimization among the swarm intelligence algorithms [13] [14]. This algorithm is inspired by the collective movement of the fish and their various social behaviors. Based on a series of instinctive behaviors, the fish always try to maintain their colonies and accordingly demonstrate intelligent behaviors. Searching for food, immigration and dealing with dangers all happen in a social form and interactions between all fish in a group will result in an intelligent social behavior. This algorithm has many advantages including high convergence speed, flexibility, fault tolerance and high accuracy.

Consider the state vector of artificial fish $x$consists of $n$samples such that $X = (X_1, X_2, \ldots, X_n)$. The present state of an artificial fish sample is assumed to be x and $x_v$is assumed to be the new state of artificial fish sample in the visual of$x$ chosen based on equation (4).

$$X_v = X + Visual * r \tag{4}$$

Then the basic movement process can be expressed as in equation (5).

$$\mathrm{X_{next}} = \frac{\mathrm{X_v - X}}{\|\mathrm{X_v - X}\|} \cdot \mathrm{Step.\ r + X} \tag{5}$$

Where r produces random numbers between zero and 1, Step is the step size of a move and $dis(X_i, X_j)$ is a distance measure between two artificial fish samples[15]. In formula (8), X represents the global optimal position of food concentration. Figure 4 shows the visual and step functionality of artificial fish.

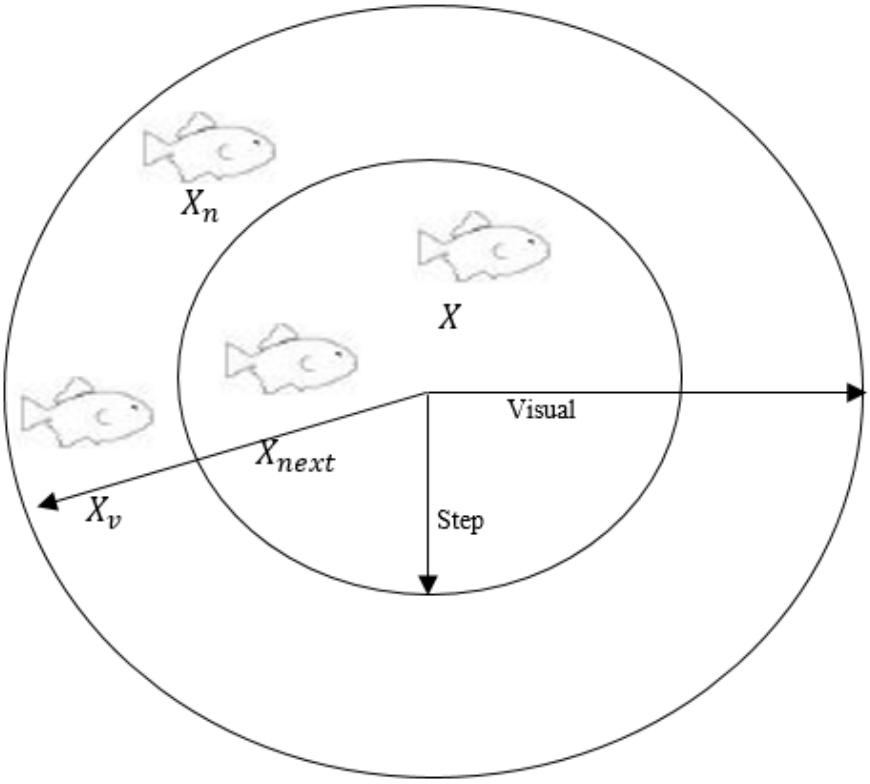

Fig 4. Visual and Step functionality of Artificial Fish

In the standard artificial fish algorithm, Step, Visual two parameters are fixed. Bigger Step, and Visual parameter values can guarantee fast convergence at the early stage of the algorithm, but reduces accuracy, and even lead to the local optimal search results. For balancing algorithm's convergence speed and precision, dynamic parameter is introduced namely regulate factor l(0<l<1). Let Step, and Visual parameters are adapting the dynamic changes in the score based artificial fish algorithm. At the time of artificial fish movement, the standard algorithm failed to obtain global optimal values. To overcome the disadvantages, the mobile reference factor is expanded from the original food concentration centre combined with the global optimal position in foraging behavior.

$$Visual = Visual\,(1 - \lambda) \tag{6}$$

$$Step = Step\,(1 - \lambda) \tag{7}$$

$$X_{next} = \frac{X_{best} + X_v - 2 \cdot X}{\|X_{best} + X_v - 2 \cdot X\|} \cdot \text{Step.}\, r + X \tag{8}$$

Using artificial fish algorithm search out the optimal value which is the optimal value of objective function theory for continuous function optimization, so it is possible to get as high as a limited time precision that it is the key of the artificial fish algorithm (optimization algorithm). Experimental data shows that artificial fish algorithm late iteration of the effect on accuracy function finally, belong to "invalid iterative calculation.

Limitations of AFSA [16]

- High structural and computational complexities
- Lack of using AFs' previous experiences
- Lack of appropriate balance between exploration and exploitation to improve the optimization process.

Proposed Score based AFSA

In this section, the above mentioned limitations are handled and eliminated in order to improve the overall performance of AFSA. Late to eliminate the waste and improve the artificial fish algorithm of rapidity and precision of the permitted error introduced precision K and iterative adaptive termination number Z, and connecting with the grid search method, make the artificial fish after the operation accuracy of convergence to the range of allowable error timely termination of iteration, saves the operation time. As a result of the existence of random behavior in artificial fish algorithm for computing the global optimal solution of the late, after the expiration of the iteration, a local grid traversal can overcome much of random behavior influence on final precision, improve the computing accuracy.

The pseudocode of Score based artificial fish swarm algorithm is given as follows:

Pseudocode: Score based Artificial Fish Swarm Algorithm

Input: Parameter initialization, visual, try number, crowd factor, Feature subset X_i for each fish AF_i(i=1,2,..,n)

Output: Best feature subset

1. In initialisation first initialise feature subset using AFS algorithm.
2. Find fitness value, for each and every feature subset
   For each $X_i \in$ Features
   Calculate Information gain
   $$IG(Xt, p(i)) = H(Xt) - \sum \left(\frac{|\{x \in Xt | value(x, p(i)) = v\}|}{|Xt|} \right) H(\{x \in Xt \mid value(x, p(i)) = v\})$$
   Calculate entropy
   $E(S) = \sum_{i=1}^{c} -p_i log_2 p_i$
   Where $p_i \rightarrow$frequent probability of an element
   End for
   For each feature subset $X_i \in$ Features
   Find fitness score using Modified hardy-weinberg formula
   z=(p2w11 + 2pqw12 + q2w22)
   where,w11=0.1,w12=0.1,w22=0.1.
   p=information gain
   q=entrophy
   z=fitness value which is going to be maximised
   End for
3. while (t <MaxGeneration)
   For each AF_i do
   Perform Follow behaviour on X_i(t) and compute $X_{i,follow}$ Perform swarm behaviour on X_i (t) and compute $X_{i,swarm}$ Calculate r value
   phi1=2.05;
   phi2=2.05;
   phi=phi1+phi2;
   chi=2/(phi-2+sqrt(phi^2-4*phi));

r=chi
If $F(X_{i,\ follow}) < F(X_{i,\ swarm})$
$X_{next} = \frac{X_v - X}{\|X_v - X\|} \cdot \text{Step.}\ r + X$
Else
$X_{next} = \frac{X_{best} + X_v - 2 \cdot X}{\|X_{best} + X_v - 2 \cdot X\|} \cdot \text{Step.}\ r + X$
End if
End for

4. Find best feature subset based on maximum z value.
5. Repeat the process until convergence attained

Convergence is attained using maximum iteration or repetition of same value in atleast 5 iteration.

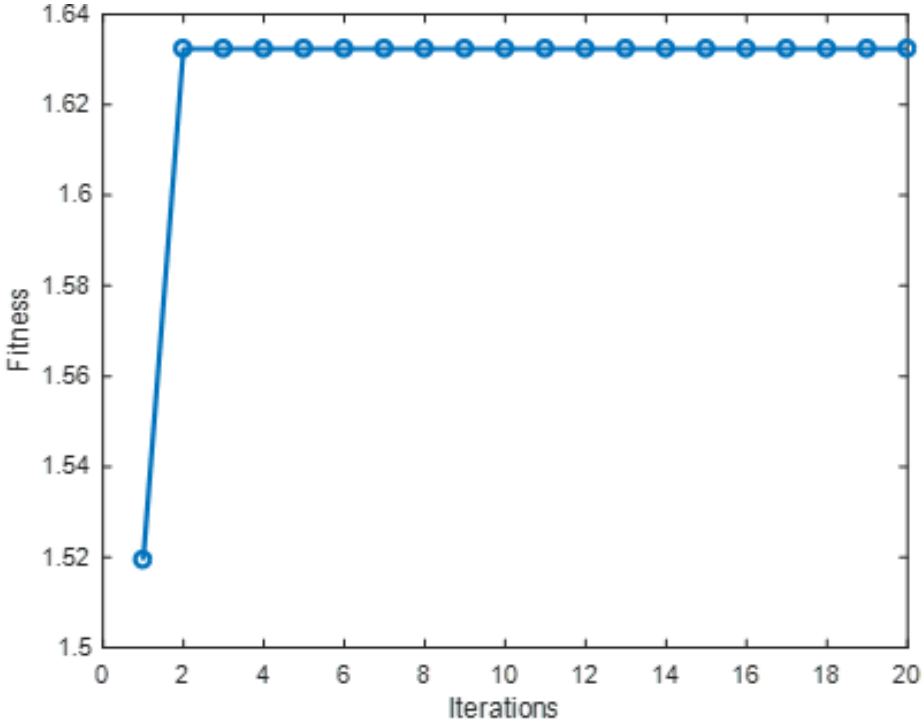

Fig 5. Implemented Screenshot of SAFSA

Figure 5 shows the proposed feature selected results of SAFSA. In the proposed score based artificial fish swarm algorithm, r is updated at each iteration in the step and visual equation in order to converge towards the best solution. This r weight value is multiplied with the equations. Next modification is in fitness value calculation which is done using modified hardy-weinberg formula. The z value obtained in our algorithm is 1.713. By using this algorithm 4 features are selected from the total number of 10 features. Those are Alkaline_Phosphotase, Alamine_Aminotransferase, Albumin_and_Globulin_Ratio, Direct_Bilirubin.

2.4 Classification using modified convolutional neural network

Classification is done using modified convolutional neural network is used for classification of the dataset. In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics.

In this work there are three layers applied to ensure computation overhead reduced accurate prediction. Those are input layer, convolution layer, pooling layer, and finally soft max or fully connected layer as shown in Figure 6. CNN is usually composed of two parts. In part 1, convolution operation is used to generate deep features of the raw data. And in part 2, the features are connected to an MLP for classification. Here are some details for each layer:

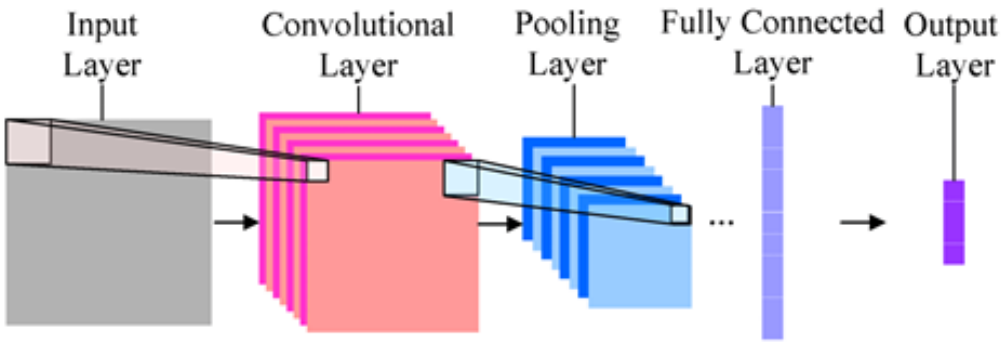

Fig 6. General CNN Architecture

1. **Input layer:** Input layer has N × k neurons, where k denotes the variate number of input data and N denotes the length of each data
2. **Convolutional layer:** Perform convolution operations on the data of preceding layer with convolution filters. Here are some filter parameters to be determined previously according to domain knowledge or just depending on experiments, such as, filter numbers m, convolution stride s and the size of filter k × l, where k denotes the variate number of the data in the preceding layer and l denotes the length of filter. A nonlinear transformation function f also needs to be determined in this layer. For instance, if the preceding layer contains k-variate data and the length of each data is N, after the Convolutional operation, we get m-variate data and the length of each univariate is $\left\lfloor \frac{N-l}{s} + 1 \right\rfloor$, where $\lfloor . \rfloor$ denotes rounding down.
3. **Pooling Layer:** Pooling layers or downsampling layers, perform dimensionality reduction, minimizing the number of parameters in the input sample. The pooling process sweeps a filter across the whole data sample, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array.
4. **Softmax layer or Fully Connected Layer:** A Softmax function is a type of squashing function. SoftMax function calculates the chances distribution of the event over n different events. generally, a way of claiming, this function will calculate the chances of every target text over all possible target text. Later the calculated probabilities are helpful for determining the target text for the given inputs. The most advantage of using SoftMax is that the output probabilities range. The range will be 0 to 1, and also the sum of all the changes is adequate. If the SoftMax function is used for the multi-classification model it returns the chances of every class and also the target text will have a high probability. The formula computes the exponential (e-power) of the given input value and also the sum of exponential values of all the values within the inputs. Then the ratio of the exponential of the input value and also the sum of exponential values is that the output of the SoftMax function. It's used for the multi-classification task and within the different layers of neural networks. The high value will have a better probability than other values. SoftMax layers are good at determining multi-class probabilities, however, there are limits. SoftMax can become more expensive as the number of classes grows. In those situations, candidate sampling can be a more effective workaround. With candidate sampling, a SoftMax layer will limit the scope of its calculations to a particular set of classes.
5. **Output layer:** The output layer has n neurons, corresponding to n classes of features. It is fully connected to the feature layer. The most popular method is taking the maximum output neuron as the class label of the input emotion in classification task.

Limitations of General CNN

The major limitation of CNN is its inability to encode Orientational and relative spatial relationships, view angle. CNN do not encode the position and orientation of data. Lack of ability to be spatially invariant to the input data sample.

Proposed Modified CNN

The CNN is trained via a sequence of training examples $((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N))$ with $x_t \in R^{N \times k}, y_t \in R^n$ for $1 \leq t \leq N$. The high-order features x_t is given as input to the network, while the vector **y**t denotes the target output. The network is trained

according to the following several steps:

Step 1 Initialize the network. Determine the CNN architecture, composed of Convolutional layer, and softmax layer, as shown in Figure 6. Fix the neuron number of input layer and output layer according to the classification task. Set all the CNN parameters. Initialize the weights and bias with a small random number. Select a learning rate η, and an activation function f, and the commonly used example is the sigmoid function as in equation (9):

$$f(x) = sigmoid(x) = \frac{1}{1+e^{-x}} \tag{9}$$

Step 2 Choose a training sample from the training set randomly.

Step 3 Select the optimal values of bias and weight value using genetic algorithm. In this work, Genetic Algorithm based bias and weight optimized CNN is proposed. The three key parts of genetic algorithm (GA) is: selection, crossover, and mutation. First, the mechanism selects the elite parents to the gene pool (an array that keeps track of the best matrix of weights) for child production to realize the elitism. Second, crossover is implemented. Among the best genes (weighted matrix), the mechanism selects two genes randomly and recombines them in a certain approach. In genetic algorithm, population size, number of generations, crossover rate, and mutation rate and its probability also need to be considered when building the ANN.

When tuning the weights and biases of CNN, total number of tuning parameters should be calculated based on the CNN structure. Each individual in GA then holds a number of candidate solutions equal to the number of tuning parameters. At each iteration CNN calculates an output to the problem based on the parameters specified by GA. A separate cost function should also be defined which compares the deviations between output and real target values. GA minimizes this cost function in several iteration until a point where no further improvements could be made and then optimization is terminated. Optimized values are replaced in the final CNN structure.

Initialize parameter values
Generate the initial population
While i<MaxIteration and Bestfitness<MaxFitness do
Fitness calculation
Perform selection
Perform cross over
Perform mutation
End while
Return the best solution

Step 4 Calculate the output of each layer.

(i) The output of the Convolutional layer can be written as

$$C_r(t) = f\left(\sum_{i=1}^{l}\sum_{j=1}^{k} x(i+s(t-1), j)\,\omega_r(i,j) + b(r)\right) \tag{10}$$

Where $x \in R^{N\times k}$denotes the input higher order features or the output of the preceding layer, s denotes the convolution stride, $C_r(t) =$ refers to the tth component of the rth feature map, ωr∈Rl×k and **b**(r) refer to the weights and bias of the rth convolution filter.

(ii) The output of the output layer can be written as

$$O(j) = f\left(\sum_{i=1}^{M} z(i)\,\omega_f(i,j) + b_f(j)\right),\ j = 1,2,\ldots,n \tag{11}$$

Where z denotes the final feature map in the feature layer, $b_f$is the bias of the output layer and $\omega_f \in R^{M\times n}$refers to the connection weights between the feature layer and the output layer.

So, the mean-square error can be written as

$$E = \frac{1}{2}\sum_{k=1}^{n} e(k)^2 = \frac{1}{2}\sum_{k=1}^{n} (O(k) - y(k))^2 \tag{12}$$

Step 5 Update the weights and bias by the gradient descent method.

$$p = p - \eta\frac{\partial E}{\partial p} \tag{13}$$

Where p is the value of the parameter, and p refers to ω_r, ω_f , b,or $b_f$in this CNN.

Step 6 Choose another training sample and go to Step 3 until all the samples in the training set have been trained.

Step 7 Increase the iteration number. If the iteration number is equal to the maximum value which is set previously, terminate the algorithm. Otherwise, go to Step 2. Based on the above steps the social emotion is classified.

In this work, performance of the CNN is improvised by introducing the optimal parameter selection, in which CNN parameter values will be selected more optimally using genetic algorithm. This optimal parameter selection process would lead to accurate and efficient classification outcome. Parameter value estimation is the most important step in the CNN classifier which tends to provide the optimal classification outcome. Appropriate selection of parameter values would lead to accurate decision making. In this work, given data set is divided into three subsets for the accuracy and optimal selection of parameter values.

Training set: a set of examples used for learning: to fit the parameters of the classifier g In the MLP case, we would use the training set to find the "optimal" weights with the back-prop rule

Validation set: a set of examples used to tune the parameters of a classifier g In the MLP case, we would use the validation set to find the "optimal" number of hidden units or determine a stopping point for the back propagation algorithm

Test set: a set of examples used only to assess the performance of a fully-trained classifier. In the MLP case, we would use the test to estimate the error rate after we have chosen the final model (MLP size and actual weights). After assessing the final model with the test set, you must not further tune the model

The procedure of parameter selection process is given below:

1. Divide the available data into training, validation and test set
2. Select architecture and training parameters
3. Train the model using the training set
4. Evaluate the model using the validation set
5. Repeat steps 2 through 4 using different architectures and training parameters
6. Select the best model and train it using data from the training and validation set
7. Assess this final model using the test set

In CNN, the optimal bias and weight values are calculated using genetic algorithm, in order to improve prediction of liver disease. The bias and variance values obtained are listed below:

Bias: 0.7151 -0.9522 -0.1391 0.8866 -0.9818 0.1841 0.5042 -0.3220 0.2596 0.7990 (Neurons =10)

Weight: 0.9572 0.8442 0.9221 0.8800 0.6134 0.8967 0.5626 0.8254 0.8708 0.7949

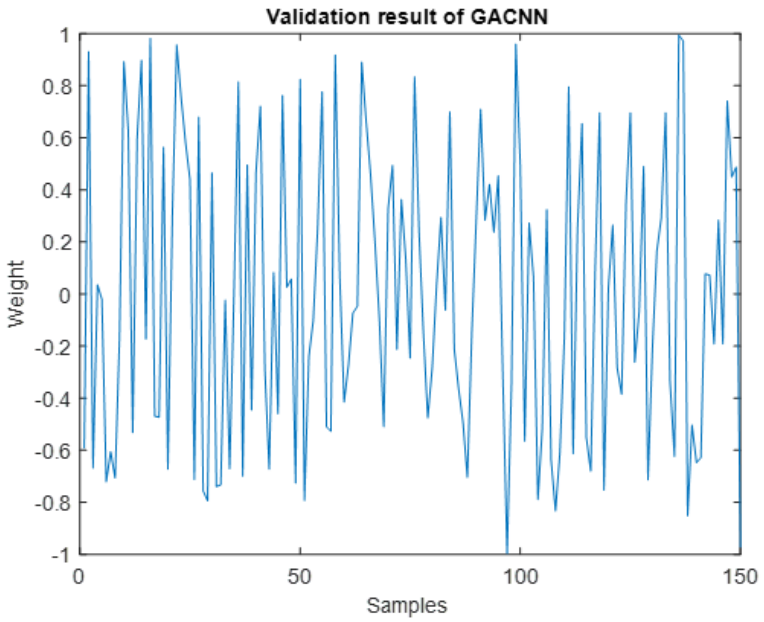

Fig 7. Weight Optimized Validation Screenshot of MCNN (GA based weight optimized CNN)

3 Results and discussion

In this section, numerical evaluation of the proposed research methodology is done in terms of various performance measures to analyze the performance improvement of the proposed and existing research methodologies. The MATLAB simulation environment is used to implement the proposed research methodology. The performance measures considered in this work are listed as follows: "Accuracy, Precision, Recall and F-measure".

The comparison is made between the proposed Modified Convolutional neural network based Liver disease prediction system (MCNN-LDPS) and the existing methodologies Multi layer perceptron neural network (MLPNN) [17].

The performance metrics values are given in the following Table 1.

Table 1. Performance Evaluation Results

Metrics	Methods	
	MLPNN	MCNN-LDPS
Accuracy (%)	86.70	90.75
F-Measure (%)	70.02	91.25
Precision (%)	84.35	88.57
Recall (%)	59.85	94.11

3.1 Accuracy

Accuracy score represents the model's ability to correctly predict both the positives and negatives out of all the predictions. Mathematically, it represents the ratio of sum of true positive and true negatives out of all the predictions.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

In the following Figure 8, comparison evaluation of the proposed CNN-LDPS and existing MLPNN technique in terms of accuracy metric is shown.

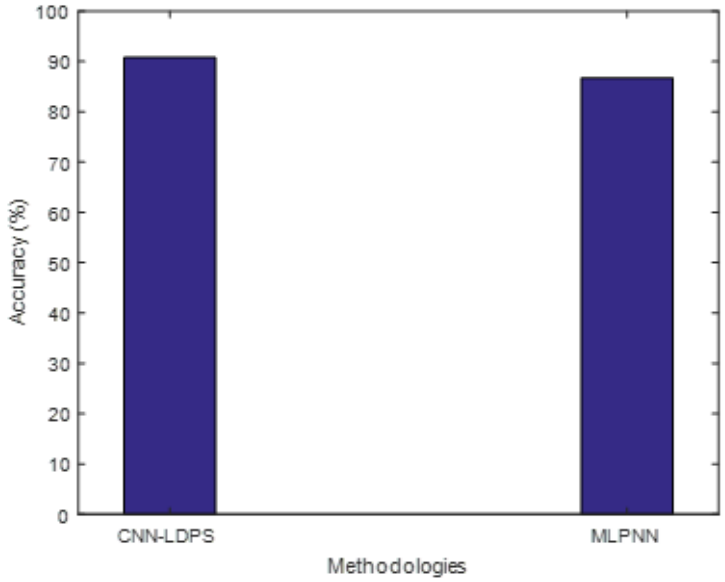

Fig 8. Accuracy comparison

From this analysis it is proved that the proposed shows better performance than the existing technique. Proposed CNN-LDPS shows improved increased accuracy than MLPNN. This is mainly because of the proposed formulations of SAFSA and MCNN. Here proposedCNN-LDPSattains4.05% increased accuracy than the existing MLNNN.

3.2 Precision

Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

$$Precision = Truepositives/(Truepositives + Falsepositives) = TP/(TP + FP)$$

The performance analysis in terms of Precision metric is shown in Figure 9. It is clearly observed from the graphical evaluation that the proposed MCNN-LDPS method attains better Precision than the existing MLPNN method. This significant performance of MCNN-LDPS is due to the modifications carried out in the feature selection and classification steps of the overall system.

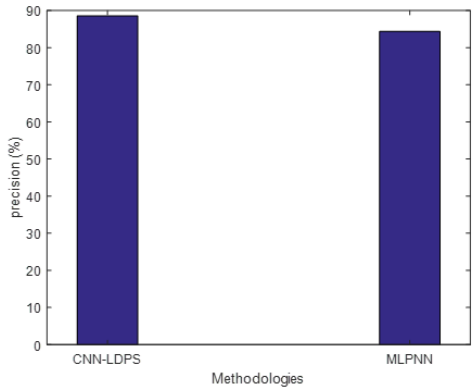

Fig 9. Precision metric comparison

In Figure 9. Comparison analysis of the proposed and existing method in terms of precision metric is given. Here precision of the proposed methodology CNN-LDPS attains 4.22% increased precision than the existing MLPSS

3.3 Recall

Recall score represents the model's ability to correctly predict the positives out of actual positives.

$$Recall\ Score\ =\ TP\ /\ (FN\ +\ TP)$$

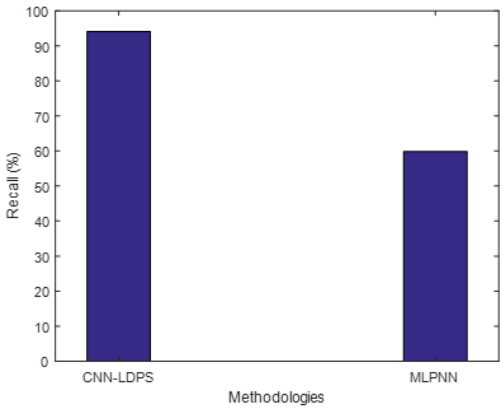

Fig 10. Recall Comparison

The graphical evaluation of the recall score is clearly depicted in Figure 10. The graph clearly shows the difference in the recall score between the proposed MCNN-LDPS approach and the existing MLPNN approach. This is mainly because of the inefficiency of the existing MLPNN approach and it lacked the improvements carried out in the proposed system. Reall of the proposed methodology MCNN-LPDS attains 34.26% increased recall than the existing method MLPNN.

3.4 F- Measure

F-Measure provides a way to combine both precision and recall into a single measure that captures both properties. The traditional F measure is calculated as follows:

$$F-Measure = (2 * Precision * Recall) / (Precision + Recall)$$

The F Measure graphical evaluation is clearly shown in Figure 11. It is observed that F-measure score of proposed CNN-LDPS is 91.25 % whereas the F-measure score of the existing MLPNN is 70.02%. Thus, the performance of the proposed CNN-LDPS is efficient and better compared to the existing model taken for comparison.

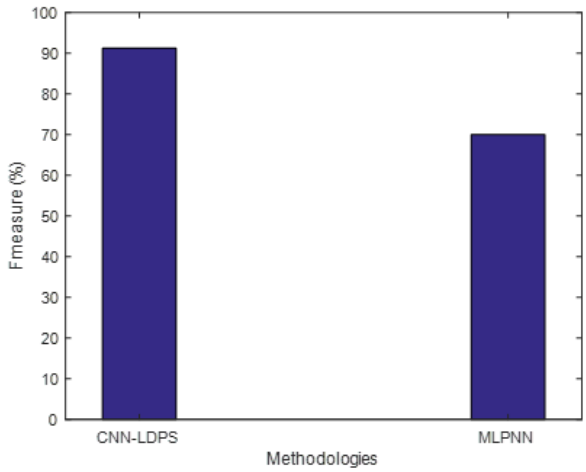

Fig 11. Measure Comparison

4 Conclusion

The proposed methodology CNN-LDPS ensures the accurate liver disease prediction outcome. The accuracy and efficiency of the classifier is improvised by performing the feature selection before classification which is done by using Score based artificial fish swarm algorithm. Here the performance of the CNN classifier is improvised by choosing the weight and bias values optimally using genetic algorithm. And also feature selection process is improvised by introducing the improved fish swarm algorithm where position update is done using the new equation. The numerical analysis of the research work has been carried out in the matlab from which it is proved that the proposed technique can ensure the 4.05% increased accurate liver disease classification outcome. Here accuracy of disease prediction is improved by integrating the genetic algorithm with the Convolutional neural network which is novelty of this research.

References

1) Rajeswari P, Sophia RG. Analysis of liver disorder using data mining algorithm. *Global journal of computer science and technology*. 2010. Available from: https://computerresearch.org/index.php/computer/article/view/652.
2) Alfisahrin SNN, Mantoro T. Data mining techniques for optimization of liver disease classification. In: and others, editor. 2013 International Conference on Advanced Computer Science Applications and Technologies. 2013;p. 379–384. doi:10.1109/ACSAT.2013.81.
3) Dhamodharan S. Liver disease prediction using bayesian classification . *COMPUSOFT: An International Journal of Advanced Computer Technology*. 2014;p. 1–3. Available from: https://ijact.joae.org/index.php/ijact/article/view/443/378.
4) Seker SE, Unal Y, Erdem Z, Kocer HE. Ensembled Correlation Between Liver Analysis Outputs. *International Journal of Biology and Biomedical Engineering*. 2014;8:1–5. Available from: https://arxiv.org/abs/1401.6597. doi:2014.
5) Aneeshkumar AS, Venkateswaran CJ. A novel approach for Liver disorder Classification using Data Mining Techniques. *Engineering and Scientific International Journal*. 2015;2(1):15–18.
6) Thangaraju P, Mehala R. Performance analysis of PSO-KStar classifier over liver diseases. *International Journal of Advanced Research in Computer Engineering & Technology* . 2015;4(7):3132–3137.
7) Vijayarani S, Dhayanand S. Liver disease prediction using SVM and Naïve Bayes algorithms. *International Journal of Science, Engineering and Technology Research*. 2015;4(4):816–820.
8) Olaniyi EOO, Adnan K. Liver disease diagnosis based on neural networks. *Advances in Computational Intelligence*. 2013;p. 48–53.

9) Baitharu TR, Pani SK. Analysis of Data Mining Techniques for Healthcare Decision Support System Using Liver Disorder Dataset. *Procedia Computer Science*. 2016;85:862–870. Available from: https://dx.doi.org/10.1016/j.procs.2016.05.276.
10) Hassan EA, Hafez AI, Hassanien AE, Fahmy AA. Community Detection Algorithm Based on Artificial Fish Swarm Optimization", Intelligent Systems. In: Advances in Intelligent Systems and Computing;vol. 323. Springer. 2014;p. 509–521.
11) Lever J, Krzywinski M, Altman N. Points of significance: Principal component analysis. *Nature Methods*. 2017;14(7):641–642. Available from: https://www.nature.com/articles/nmeth.4346.pdf.
12) Naik GR. Advances in Principal Component Analysis: Research and Development. and others, editor;Springer. 2017. Available from: https://link.springer.com/content/pdf/10.1007/978-981-10-6704-4.pdf.
13) Kambhatla N, Leen TK. Dimension Reduction by Local Principal Component Analysis. *Neural Computation*. 1997;9(7):1493–1516. Available from: https://dx.doi.org/10.1162/neco.1997.9.7.1493.
14) Neshat M, Sepidnam G, Sargolzaei M, Toosi AN. Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artificial intelligence review*. 2014;42(4):965–997.
15) Gulia AA, Vohra RR, Rani PP. Liver patient classification using intelligent techniques. *International Journal of Computer Science and Information Technologies*. 2014;5(4):5110–5115.
16) Azizi R. Empirical Study of Artificial Fish Swarm Algorithm. *International Journal of Computing, Communications and Networking*. 2014;3(1):1–7. Available from: https://arxiv.org/abs/1405.4138.
17) Neshat M, Adeli A, Sepidnam G, Sargolzaei M, Toosi AN. A review of artificial fish swarm optimization methods and applications. *International Journal on Smart Sensing and Intelligent Systems*. 2017;5(1):108–148. Available from: https://doi.org/10.21307/ijssis-2017-474.