ISSN: 0067-2904

Deep Learning and Machine Learning via a Genetic Algorithm to Classify Breast Cancer DNA Data

Noor Alhuda Khalid Hussein*, Basad Al-Sarray
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Received: 20/7/2020 Accepted: 26/9/2022 Published: 30/7/2022

Abstract

This paper uses Artificial Intelligence (AI) based algorithm analysis to classify breast cancer Deoxyribonucleic (DNA). Main idea is to focus on application of machine and deep learning techniques. Furthermore, a genetic algorithm is used to diagnose gene expression to reduce the number of misclassified cancers. After patients' genetic data are entered, processing operations that require filling the missing values using different techniques are used. The best data for the classification process are chosen by combining each technique using the genetic algorithm and comparing them in terms of accuracy .

Keywords: deep learning, genetic, DNA, CNN, PNN, KNN and LSTM.

التعلم العميق عبر الخوارزمية الجينية لتصنيف بيانات الحمض النووي لسرطان الثدي

نور الهدى خالد، بسعاد علي

قسم علوم الحاسبات، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة:

تهدف هذه الورقة إلى تحليل خوارزمية قائم على الذكاء الاصطناعي لتصنيف الحمض النووي لسرطان الثدي. الفكرة الرئيسية هي التركيز على تطبيق تقنيات التعلم الآلي والتعلم العميق. علاوة على ذلك، خوارزمية جينية لتشخيص التعبير الجيني لتقليل عدد السرطانات المصنفة بشكل خاطئ. حيث يتم إدخال البيانات الجينية للمرضى وعمليات المعالجة التي تتطلب ملء القيم المفقودة باستخدام تقنيات مختلفة، ومن ثم يتم اختيار أفضل البيانات لعملية التصنيف عن طريق الجمع بين كل تقنية باستخدام الخوارزمية الجينية ومقارنتها من حيث الدقة.

Introduction

The biology used to be a science that lacks data. However, because of recently more evolved and upgraded technologies, biologists are now able to transform an incredible quantity of biological information into useful data. This makes the study of gene functionality possible on a global scale. Machine learning and deep learning are two of the study of algorithms, which can learn from experience then predict. Given their rootedness in statistics and informatics, computational considerations are also indispensable. Given the complexity of

*Email: engnooralhuda87@gmail.com

biological information, it can play an important role within the analytical process [1] During a literature review of tissue classification, it had been note that breast tissue cancer is that the most typical malignant cancer and thus one of the main causes of mortality among women. Early detection of tumors is the most effective way to avoid mastectomy, reduce the prospect of them returning and the mortality rate. Different classification results could be obtain relying on the classifiers chosen to help physicians or radiologists to classify the breast genes (benign, malignant)**.**

.1 Related work

DNA classification has recently seen a great deal of scientific contributions using intelligent computing models. According to [2] Classification of DNA methylation datasets to identify carcinogens and the introduction of BIGBIOCL, an algorithm that can apply supervised classification methods to datasets containing hundreds of thousands of Features. The algorithm is designed to extract equivalent and alternative classification models by repeatedly deleting pre-selected traits. Then in another work [3] a hybrid algorithm called Genetic Algorithm and Learning Automata (GALA), derived from a combination of genetic algorithm and machine learning, introduced the problem of selecting genes in accurate cancer datasets. The SVM classifier has been applied as a proposed classification model algorithm, and results obtained by implementing GALA on six datasets for cancer gene expressions are remarkable compared to other recently introduced algorithms. Then, [4] based on the application of AEs survival analysis, a methodology for dealing with high-dimension, low-sample size (HDLSS) data sets for DNA methylation was presented. This procedure has been used to obtain useful information about breast cancer recurrence based on key genes. Then in another work [5] The Vector machine has been created. In comparison to most of the current splice site predictions in use today, this technique demonstrates a significant improvement. The results of the experiments showed that the second-order Markov model is a good preprocessing strategy. When paired with a supporting vector machine, this method improves the accuracy of predicting splice sites.

Breast tissue cancer is the most common malignant cancer among women and the cause of many terminal

cases. Early detection of tumors is the best solution to avoid mastectomy, reduce the chance of them returning and reduce the death rate. Doing so will help doctors or radiologists to classify the breast genes (benign, malignant). The pilot study aimed to find an estimated performance of the disclosure model proposed in the current study. Using genetic algorithm and classifier (convolutional neural network (CNN), long short-term memory (LSTM), probabilistic neural network (PNN), or K-nearest neighbor(KNN)) to generate and compare results faster. The first step involved is collecting the complete DNA data set, cleaning and filling in the missing value, then applying a number of deep learning and machine learning (CNN, LSTM, PNN, or KNN) and hybridizing each technique using a genetic algorithm to classify and compare the DNA of the dataset. Figure (1) illustrates the architecture of the proposed system

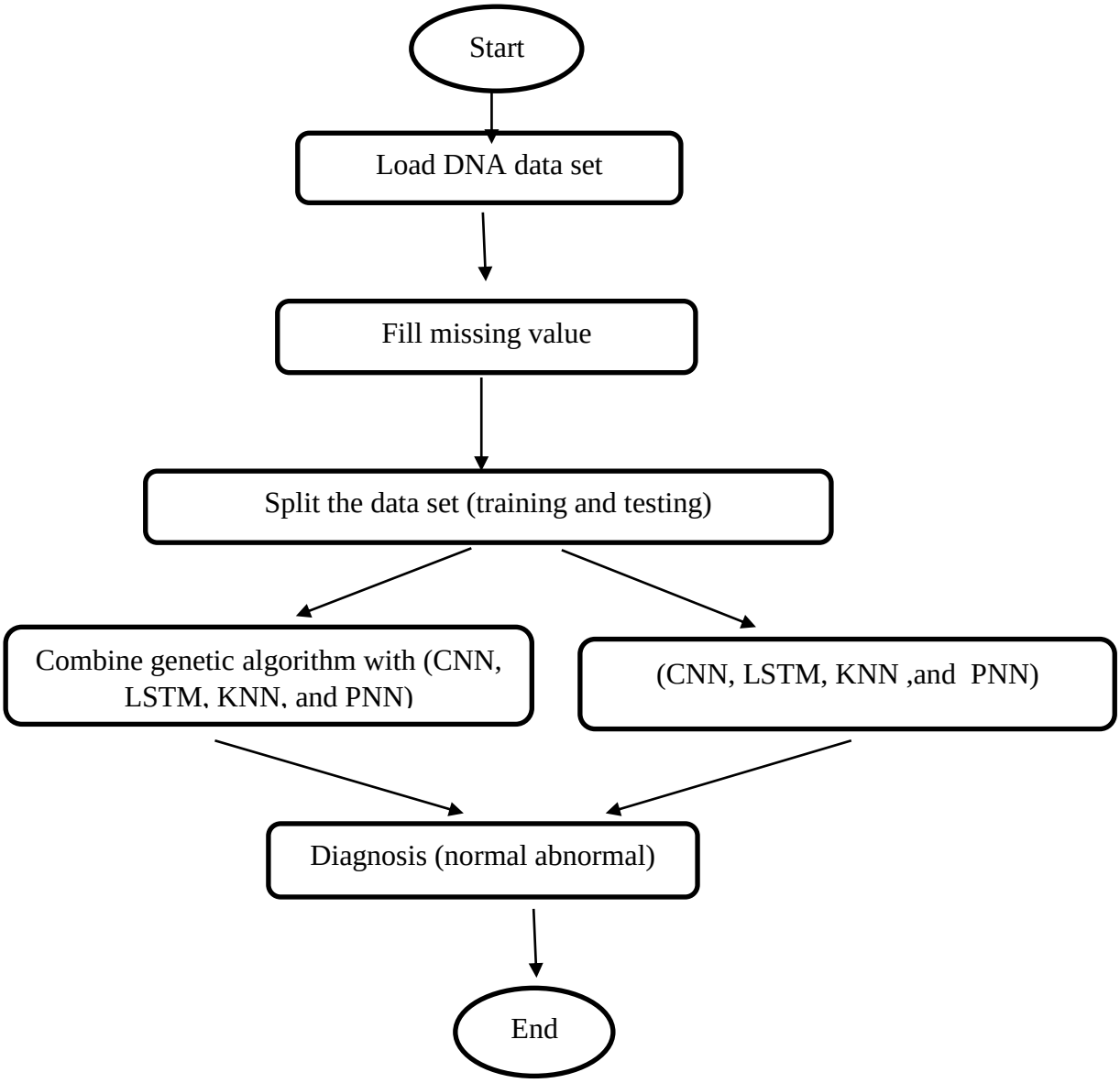

Figure 1-The architecture of the proposed system

4. Proposed algorithms or methods

4.1. Classification with machine learning:

Neural networks are sometimes used to categorize patterns via learning from examples. Varied neural network models have different learning techniques, but they always use a set of training samples to discover the statistics of the patterns and then categorize the new patterns based on these statistics. The learning model was created to classify the data after preparing the input data set as the missing values were filled in and the data was cleaned up. A model (PNN, KNN) is created which are two of the machine learning algorithms to classify the data set. Before building the learning model, the training data and the test data are separated so that the training data includes 80% of the total data and the model is built on this basis and this means that the input data includes 207 fields in 70 columns.

4.1.1 Classification with Probabilistic Neural Network (PNN)

PNN is a sort of self-observing feed-forward network with Bayesian least danger standards (i.e. Bayesian decision theory) as its theoretical basis [6] [7]. In the statistical

classification computing process, Parzen window estimations can be used to derive class conditional probability density and hence classification samples. It is not necessary to train the samples' connect weights; instead, the hidden layer is created based on the training samples provided to input layer, pattern layer, summation layer, and output layer make up the structure of the PNN model.

The training data samples from the input layer is received, which means that the training samples' input feature vector is supplied into the PNN neural network. The number of input layer neurons is the same as the number of features in the training sample parameter. For classification, pattern layer neurons with the same number of training instances are used. The connection between the input training samples and the various training sample patterns is computed and shown in algorithm (1) .

Algorithm (1) : PNN Classifier [8]

Input : dataset

Output : build model

Read the new input data X_{new}

Step one : compute each known input vector's Gaussian kernel using (All of the Gaussian values for Class k are summed and the sum is scaled to unity so that the probability volume beneath the sum function is unity and the sum creates a probability density function) at the output node for Class k (k = 1 or 2 here).

$$\omega_{i,j} = \frac{1}{(2\pi)^{d/2}\sigma^{d}} \exp\left(-\frac{\left(X_{\text{new}} - X_{i,j}\right)^{T} \cdot \left(x_{\text{new}} - X_{i,j}\right)}{2\sigma^{2}}\right)$$

Step two : Using classmate kernels, compute each class's class-conditional probability.

$$\{\omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \dots, \omega_{1,|c_1|}\} \rightarrow P_1 = \frac{1}{|c_1|}\sum_{j=1}^{|C_1|} \omega_{1,j}$$

$$\{\omega_{2,1}, \omega_{2,2}, \omega_{2,3}, \dots, \omega_{2,|C_2|}\} \rightarrow P_2 = \frac{1}{|c_2|}\sum_{j=1}^{|C_2|} \omega_{2,j}$$

$$\vdots$$

$$\{\omega_{NC,1}, \omega_{NC,2}, \omega_{NC,3}, \dots, \omega_{NC,|C_{NC}|}\} \rightarrow P_{NC} = \frac{1}{|C_{NC}|}\sum_{j=1}^{|C_{NC}|} \omega_{NC,j}$$

Step three: choose the class that has the highest class-conditional probability. Assign the specified class to the new input data X_{new} class.:

$$\underset{1\le i\le NC}{\operatorname{argmax}}\{P_i\}$$

4.1.2 Classification with K_Nearest Neighbor (KNN)

K_nearest neighbors (KNN) algorithm is a form of supervised machine learning method that can be used for both predictive classification and regression problems, but is commonly used in industry to classify predictive problems. It's a classifier algorithm where the learning is based on "how similar" a data (a vector) is from the other. In the case of KNN, the new unclassified data is not compared to the remainder; instead, it performs a mathematical computation to determine the distance between the data, which is then categorized using Euclidian distance, as shows in Algorithm (2) [9].

Algorithm (2) : KNN Classifier
Input : dataset
Output : build model
Load the dataset **Step one** : Set the value of k to zero. **Step two** Iterate from 1 to the total number of training data points to determine the projected class. 1. Determine the distance between each row of training data and the test data. As a distance metric, we'll utilize Euclidean distance. $$\boldsymbol{distance}\ (\boldsymbol{p},\boldsymbol{q}) = \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \ (\boldsymbol{q_i} - \boldsymbol{p_i})^{\boldsymbol{2}}}$$ 2. Using the distance numbers as a guide, sort the computed distances in ascending order. 3. From the sorted array, get the top k rows. 4. Determine the most common class of these rows. **Step three** : Return the predicted class

4.2. Classification with deep learning:

Deep learning is a type of machine learning technique that can combine raw data into layers of intermediate characteristics. Recently, these algorithms have shown excellent results in a range of disciplines. While biology and medicine are data-intensive fields, the data is frequently complicated and misunderstood. As a result, deep learning approaches might be useful in resolving difficulties in these domains [10]. Deep learning is not entirely new, but thanks to the improved performance of today's computers, it is possible to train large models with a massive amount of training data (big data) in an efficient way, in addition to the many algorithms and methods in the field that are constantly being improved and expanded. Some of the main aspects mentioned by the experts regarding deep learning are: the use of special structures such as multi-layered neural networks and convolutional neural networks, the reverse propagation algorithm, and the use of different types of layers such as (SoftMax, Dropout, Clustering). In this study was used two of these techniques (CNN, LSTM). The experience of each algorithm separately and comparing them to find out which gives higher accuracy and among the experiments. The highest accuracy was obtained when applying the CNN algorithm and less accurate in LSTM .

.4.2.1 Classification with Long Short-Term Memory (LSTM).

Long short-term memory (LSTM) is a recurrent artificial neural network (RNN) architecture [11]. Unlike standard feed forward neural networks, LSTM has feedback connections. A cell, an input gate, an output gate, and a forget gate make up a typical LSTM module. The three gates control the flow of information in and out of the cell, and the cell remembers values at random time intervals. Information can continue to flow across networks in the loop. Each network in the loop receives data and input from the previous networks, conducts the necessary operation, and generates the output while also transferring the data to the next network as show in Algorithm (3).

Algorithm (3) : LSTM Classifier [12]
Input : dataset
Output : build model
Step one: To define an LSTM Network, set IP units, LSTM units, op units, and the optimizer (L)

Step two : Normalize the dataset (Di) into values ranging **zero** to **one** using

$$X_{\text{norm}} = \frac{X - min(X)}{max(X) - min(X)}$$

Step three : Determine the size of the training window (tw) and arrange Di appropriately.
Step four : for n epochs and batch size do
Train the Network (L)
end for
Step five : Use L to run predictions
Step six : Calculate the loss function using

$$f_{obj} = min\left(\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2\right)$$

3.2.2 Model Design of CNN

A convolutional neural network (CNN) is a multi-layered neural network with a unique architecture that can identify and find objects in a single step [13]. The steps that make up this network structure are as follows:
To begin, enter the vector, second the input vector is extract for feature extraction using the CNN architecture. In each layer of CNN, there are layers of neural networks that convert one volume of activations to another using a different technique. CNN is made up of three layers:
a. Convolutional layer, b. Aggregation layer, and c. Completely connected layer.
The convolutional layer is the first level, which processes the data set to extract just the most important features. To make a feature map or an activation map, filter the inputs. Twisted is a three-stage process that is entirely mathematical:

The first stage is the sliding mask property and the corresponding correction input, the second stage is multiplying each input value by matching with the mask feature cell, and the third stage is to summarize all of them and calculate the average of the results and the final step is to fill in the output. It's a new array of features. The second stage in the CNN model assembly process structure is to shrink the input data set matrix for each feature obtained from the previous convolutional step. This was achieved by first selecting the appropriate mask of size 2 or 3, then selecting the second step in a moving region of the cells of the data set usually 2. Third, moving the mask over the complex data sets. Finally forming each mask, the maximum value is chosen.

Algorithm (4) : CNN Pseudo code

Input :data set
Step one: Convolution Layer
X represent the number of row of dataset , r represent the mask size , N represent number of filter

```
For count=1: N
    For  i=1 To x
        Sum =0
            For  r1=-1 to r-2
            Sum= sum  + mask  (r1) * Dataset(x+r1)
            End   for  r1
      End   for  i
  End   for  count
```

Step two :Rule layer

```
Input :output of Convolution Layer
For  i=1To x
                If   Dataset(i , j)<0
          Dataset(i,j)=0
       End  if
End   for  i
Step three : Max pool layer
Input :output of Rule layer
For  x=1 : end  of  row
    For   y=1: inc : end  of column
            Value  =0
                 For tmp =0 :  inc
                  value = max(value, tmp)
              end for
        output  =  [ output;  value ]
    end for
end for
Step four :   Fully Connected Layer
Input :output of Max pool layer
 for i=1 : m do
      tmp = 0
          for  j=1: n       do
              tmp = tmp + W[i] [j] x[j]
           end for
        Y [i] = tmp
 end for
 return [Y]
```

3.3. Classification using genetic algorithm and (PNN, KNN, LSTM, CNN)

The learning model is designed to classify the data after preparing the input data set as the missing values are filled in and the data is cleaned up. The genetic algorithm works with binary string structures similar to biological organisms (genomes). Using a random and structured information sharing method, these structures grow over time according to the survival of the fittest base. As a result, a new set of binary chains is generated. Each generation, based on fragments from the fittest members of the old set. Genetic algorithms work using a binary representation of the parameter space. This coding (which is an important element of the genetic algorithm design process) produces binary strings of data. The coding system in our instance consists of parameters that reflect the data used to diagnose illness. Each argument has been encoded as a binary string. This binary string is the GA's evolving genome. Each binary string represents a gene in the genetic algorithm's genome. The genetic algorithm population is the asset of such genomes. The genetic algorithm was used to reduce the input data in this study. It selects the best 20 columns among the 70 in order to obtain the highest accuracy and less time in determining the data category. After these columns are entered into one of the algorithms and the accuracy is compared before and after the use of the genetic algorithm and the fitness function depends on the function (PNN, KNN, LSTM, CNN) is shown in algorithm (5).

Algorithm (5) : Method steps : Combine Genetic algorithm And CNN, PNN, KNN, or LSTM Classifier
Input : dataset
Output: build model
Step one : After cleaning and missing value The DNA dataset data consists of(296 raw and 70 column) **Step two**: Random population generation, where 1000 individuals (chromosome) are generated and each chromosome contains 20 cells, as a cell contains the column heading in the used dataset, with no similar numbers being repeated in one chromosome **Step three**: Fitness function, the calculation of the efficiency of each chromosome using **CNN , PNN, KNN, or LSTM** each chromosome contains 20 columns of data is taken ignore the rest and the accuracy of the data entered into the **CNN, PNN, KNN, or LSTM** is calculated, knowing that the separation of training data and test data is 80% randomly)Ion Selection: Two chromosomes are randomly selected as parents ***/Step four** : Crossover operation, In this step the two chromosome exchange is separated and the second part of the first chromosome is replaced by the second part of the second chromosome using (single point) and the formation of sons **Step five:** Mutation operation: from the previous step, two or more numbers may be repeated. In this step, the number repeated is replaced with a new number. **Step six:** Fitness The fitness calculation for children is produced using **CNN, PNN, KNN, or LSTM** **Step seven :** Update the population **Step eight :** Repeat previous operations (from the selection step, until the method reach the desired goal or 100 iterations

4 - Data Description

The gene expression profile of 70 genes associated with risk of early distant metastases was determined in young patients with lymph node-negative breast cancer. In the current study, this profile was tested in a series of 295 consecutive patients who were treated at the Netherlands Cancer Institute Hospital [14]. Using a microarray analysis, the dataset of patients with primary breast cancer was classified as having a signature of gene expression associated with either a poor prognosis or a good prognosis. All patients had stage 1 or 2 breast cancer; 151 had lymph node disease negative, and 144 had positive lymph node disease. Out of the 295 patients, 82 patients had a poor prognostic signature and 115 had a good prognostic signature. Multivariate Cox regression analysis showed that the prognostic profile was a robust and independent factor in predicting disease outcome as shown in Table (1 and 2).

Table 1-Describe the Label of 71 Geneusedin dataset

The Name Of 70 Gene Used In Dataset(Header)						
G1	BBC3	G9	G12	COL4A2	G17	NMU
G2	ALDH4	G10	RAB6B	L2DTL	G18	AKAP2
G3	DCK	ECT2	G13	HSA250839	CFFM4	G20
G14	DC13	GMPS	LOC51203	KIAA1442	MCM6	PRC1
G4	CEGP1	HEC	UCH37	SERF1A	AP2B1	G21
G8	EXT1	WISP1	PECI	DKFZP564	G19	CENPA
G5	FLT1	PK428	KIAA1067	SLC2A3	IGFBP5	SM.20
G15	GNAZ	G16	FGF18	PECI.1	LOC57110	CCNE2
G6	OXCT	G11	TGFB3	ORC6L	MP1	ESM1
G7	MMP9	GSTM3	KIAA0175	RFC4	IGFBP5.1	FLJ11190

Table 2-Sample of DNA dataset

Id	G1	G2	…	CCNE2	ESM1	FLJ11190	Label
127	0.151	-0.21	…	0.278	-0.16	-0.144	0
130	0.308	0.035	…	0.003	0.061	0.083	0
131	0.553	-0.136	…	0.247	-0.152	0.54	0
132	-0.096	-0.038	…	-0.049	0.187	-0.106	0
134	0.089	-0.181	…	-0.177	-0.097	-0.327	0
. . .							
380	-0.373	0.083	…	-0.128	0.015	0.096	1
191	-0.388	-0.149	…	-0.308	0.204	-0.269	1
303	-0.022	0.151	…	-0.387	0.348	0.019	1
345	-0.094	-0.081	…	-0.326	0.495	-0.2	1

4.1. Data Cleaning Dataset

Once the data was selected, we had to clean the data in order to raise the quality of the data to the required level. In this work, all the same fields are deleted in terms of data as similar data negatively affect the learning model, and the mechanism of action is to take the first values in the column and subtract them from all cells In the field. Then, we take the sum of the cells if the value is not equal to zero. This means that all the fields are the same.

4.2. Preprocessing Dataset

After cleaning the dataset, the missing data in the dataset is fill in depending on the missing value techniques (sampling value, next value, closest value, segment, slice, linear equation, and mean) where the blank cell within the single column is determined and then the root mean square calculation to know the minimum error to fill the empty cell with it.

4.3. Classification the DNA Dataset

4.3.1. The LSTM Classification of the DNA dataset

In the proposed work, use LSTM to classify a nucleic acid dataset. LSTM classifiers for trained taxonomists store prior probabilities, parameter values, support vectors, computational LSTM implementation information and training data. In LSTM, for optimal training performance, these various parameters LSTM to get optimize training performance these given various parameter

Layer1 : sequence Input Layer(72)
Layer2: lstm Layer(num Hidden Units ,'Output Mode' ,'last')
Layer3: fully Connected Layer(2)
Layer4: softmax Layer

Layer5: classification Layer

A set of options was created to train a network using randomized scaled ratios with momentum. The learning rate was reduce by a factor of 0.2 in every 5 periods and the maximum number of training periods was set to 20. A small batch was used with 64 notes per repetition as shown in Table (3). Data simulation training 80% of the data set was used for training. Table (3) displays the results of the training classification determined by LSTM. The performance of the LSTM classifiers was obtained and without the feature selection in terms of classification, the 0.6540 resolution evolution of LSTM is presented in Table (4) and Fig(2).

Table (3): Training Options LSTM with properties

Attribute	Value
Momentum:	0.9000
InitialLearnRate:	0.0100
LearnRateSchedule:	'piecewise'
LearnRateDropFactor:	0.2000
LearnRateDropPeriod:	5
L2Regularization:	1.0000e-04
GradientThresholdMethod:	'l2norm'
GradientThreshold:	Inf
MaxEpochs:	20
MiniBatchSize:	64
Verbose:	1
VerboseFrequency:	50
ValidationData:	[]
ValidationFrequency:	50
ValidationPatience:	Inf
Shuffle:	'once'
CheckpointPath:	''
ExecutionEnvironment:	'auto'
WorkerLoad:	[]
OutputFcn:	[]
Plots:	'training-progress'
SequenceLength:	'longest'
SequencePaddingValue:	0
SequencePaddingDirection:	'right'
DispatchInBackground:	0
ResetInputNormalization:	1

Table 4-the Training-stage of-LSTM

```
Training on single CPU.
|=========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|=========================================================================================|
|       1 |           1 |       00:00:01 |       29.69% |       0.7043 |          0.0100 |
|      17 |          50 |       00:00:15 |       56.25% |       0.6870 |       8.0000e-05 |
|      20 |          60 |       00:00:18 |       60.94% |       0.6540 |       8.0000e-05 |
|=========================================================================================|
```

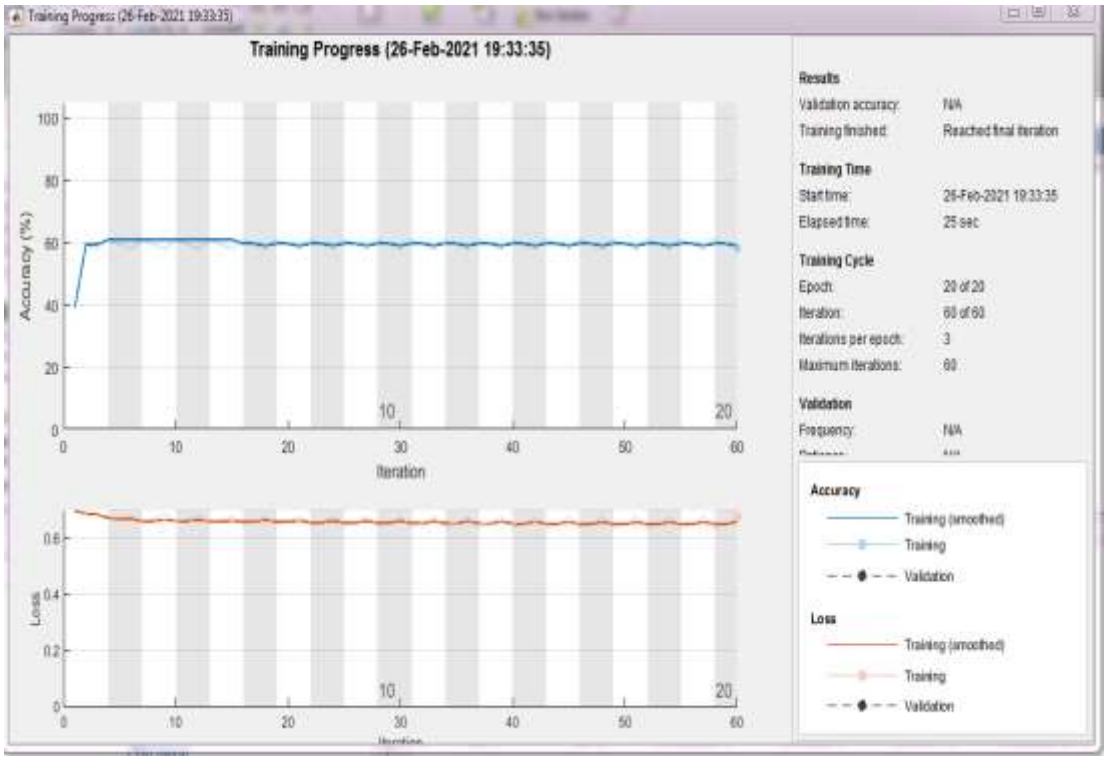

Figure 2 -Training_state of LSTM show-accuracy (60%) and Loss value (0.6)

4.3.2 The probabilistic-Neural Network-Classification of the DNA dataset

In the proposed work, applying the probabilistic-neural network to-classify the dataset of DNA by creating a two-layer network. Only the first layer of ANN has biases to get an optimized training performance. Given various parameter, PNN sets the first-layer weights to input, and resulting in radial basis functions that cross 0.5 at weighted inputs of +/– spread. The second-layer weights W2 are set to target. Figure (3) describes the proposed PNN-model for classify the dataset.

Training Data Simulation 80% of the database was used for training. 181 samples of the training data belong to benign class and 81 samples belong to malignant class. PNN classifier has nearly perfect accuracy value of 0.9153 when original features without select the best feature or reduce the amount of feature

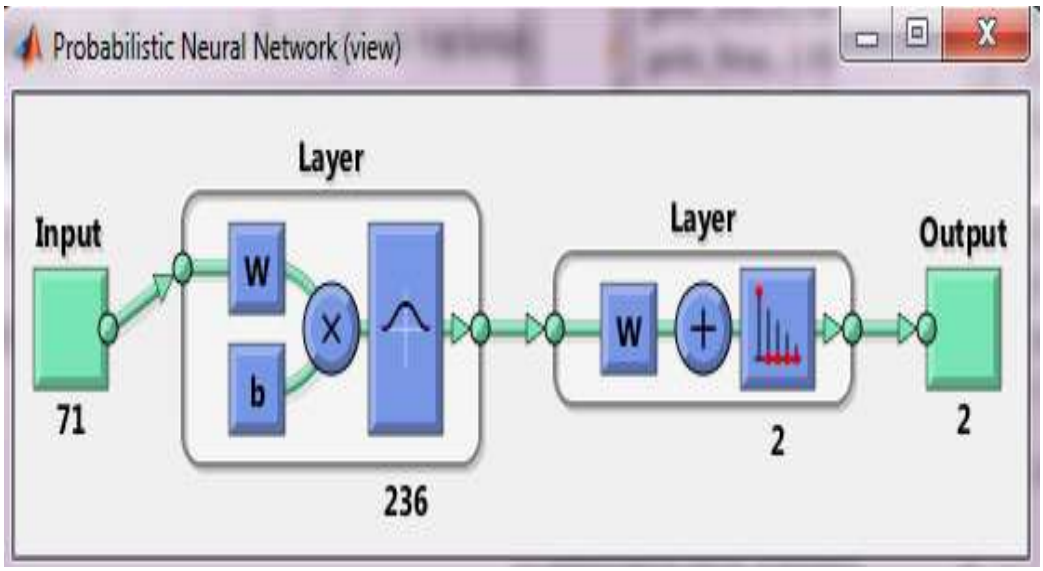

Figure 3-PNN model of right eye strabismus detection

4.3.3 The KNN-Classification the DNA dataset

In the proposed work, when applying a KNN-classification, searching a DNA dataset, and the closest KNN-location (KNN) allows finding the closest k-points in an INPUT to a query-point or a set of point naming. The KNN-search technology and KNN-based algorithms are widely used as standard-educational bases. The relative simplicity of the KNN-search technology makes it easy to compare results from other classification-techniques to the results of KNN. In KNN for optimum training performance these various parameters are given.

Table 5-KNN to get optimize training performance these given various parameter.

Md l =
Classification KNN
Response Name: 'Y'
Categorical Predictors: []
Class Names: {‘normal ;"abnormal'}
Score Transform: 'none'
Num Observations: 150
Distance: 'Euclidean'
Num Neighbors: 5

Training Data Simulation 80% of the database was used for training. 126 samples of the training data contained by benign class while 82 samples contained by malignant class. The classification results of the test set by KNN. The performance evaluations of each type distances and classification rules are chosen in function of: classification accuracy rate and time classification for each value of the nearest neighbors parameter. To validate the results, a number of tests have been carried out. The nearest rule was used for the classification rules to classify a new element. The high rate of classification accuracy was 90% recorded by the algorithm that utilizes Euclidean distance with a value of k = 1. Figure (2) illustrates that when K increases, the rate of classification accuracy decreases then takes a stable state at approximately 50, with almost 90% classification accuracy rate. However, the optimum result is generated with Euclidean distance (90%), this corroborates with what was presented in the literature. The smallest rate of classification achieved in attrition 20 with 65%. In contrast, Euclidean is time-consuming classification, these results are illustrated in Figure (4)

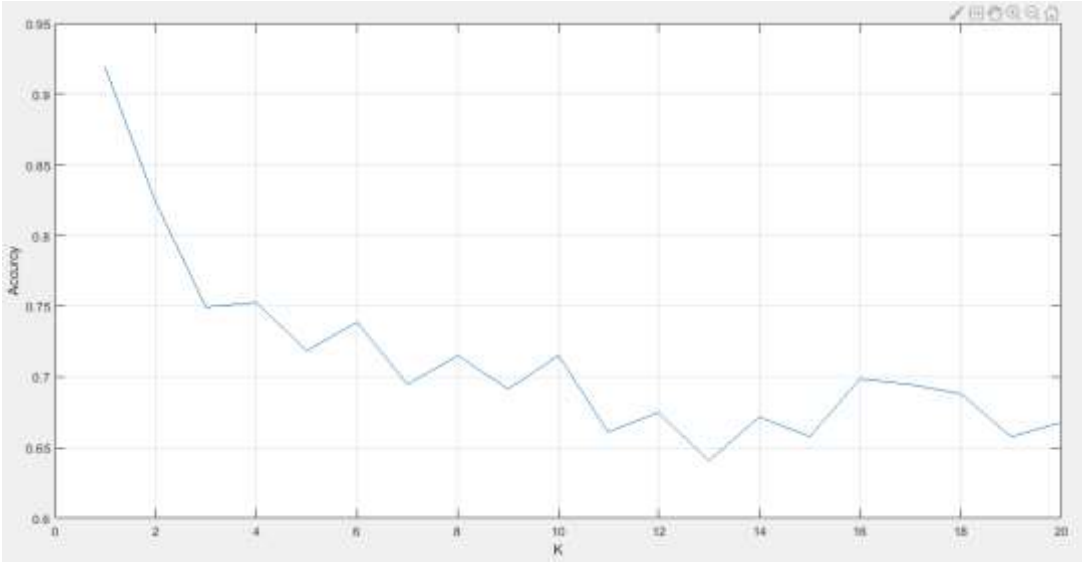

Figure 4-Representation of classification accuracy rate for each value of parameter k based on random rule.

4.3.4 Classify DNA dataset by the CNN

After preprocessing the DNA dataset by filling missing values and normalization the dataset, then introduce the CNN training model the size of enters dataset is 1*70*1 and Training setup of CNN is introduced as follows. Learning rate is 0.001 and momentum is 0.9. The network architecture consists of four convolutional layers and four pooling layers, followed by two fully connected layers each convolutional layer is followed by a Relu layer, an effective activation function to improve the performance of the CNNs. The dropout ratio is set as 0.5. The learning rate is initially set as 0.001 and the training is stopped after 1000 epoch shown in table (6).

Table 6 -Analysis result of CNN model

Layer	Name	Activations	Learnable	Properties
1	Input	70*1*20	-	The approach of zero center normalization
2	Convolution1	70*1*20	Weight 3*1*1*20 Bias 1*1*20	The size of convolution mask is 3*1 with 20 filter and stride [1 1] and padding [0 0 0 0]
3	Relu1	70*1*20	-	If $\begin{cases} x > 0 \rightarrow x \\ x < 0 \rightarrow 0 \end{cases}$
4	Pool max1	35*1*20	-	The pooling is tack the maximum value in window 1x2 with padding [0 0 0 0] and stride [1 2]
5	Fully Connected Layer	35*1*20	-	weight matrix is Multiplied by a input image and then a bias vector is added up.
6	SoftMax Layer	35*1*20	-	function of activation
7	Classification Layer	-	-	Compute the cross entropy

After building the network architecture (as shown in Table (6)), train CNN model starting in epoch 1 the parameter of time Elapsed is 1 second , parameter of accuracy 57.81% , parameter of mini batch loss 0.6932. At epoch 250 parameter of accuracy reach to 100% , parameter of mini batch loss .0861, time Elapsed is 18 second as shown in (Table (7) and Figure (5)).

Table 7-Description the-tanning state of CNN

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:01	57.81%	0.6932	0.0100
50	50	00:00:05	63.28%	0.6159	0.0100
100	100	00:00:08	85.94%	0.4215	0.0100
150	150	00:00:11	92.97%	0.2624	0.0100
200	200	00:00:15	98.44%	0.1528	0.0100
250	250	00:00:18	100.00%	0.0861	0.0100
300	300	00:00:22	100.00%	0.0525	0.0100
350	350	00:00:25	100.00%	0.0353	0.0100
400	400	00:00:29	100.00%	0.0256	0.0100
450	450	00:00:32	100.00%	0.0197	0.0100
500	500	00:00:35	100.00%	0.0158	0.0100

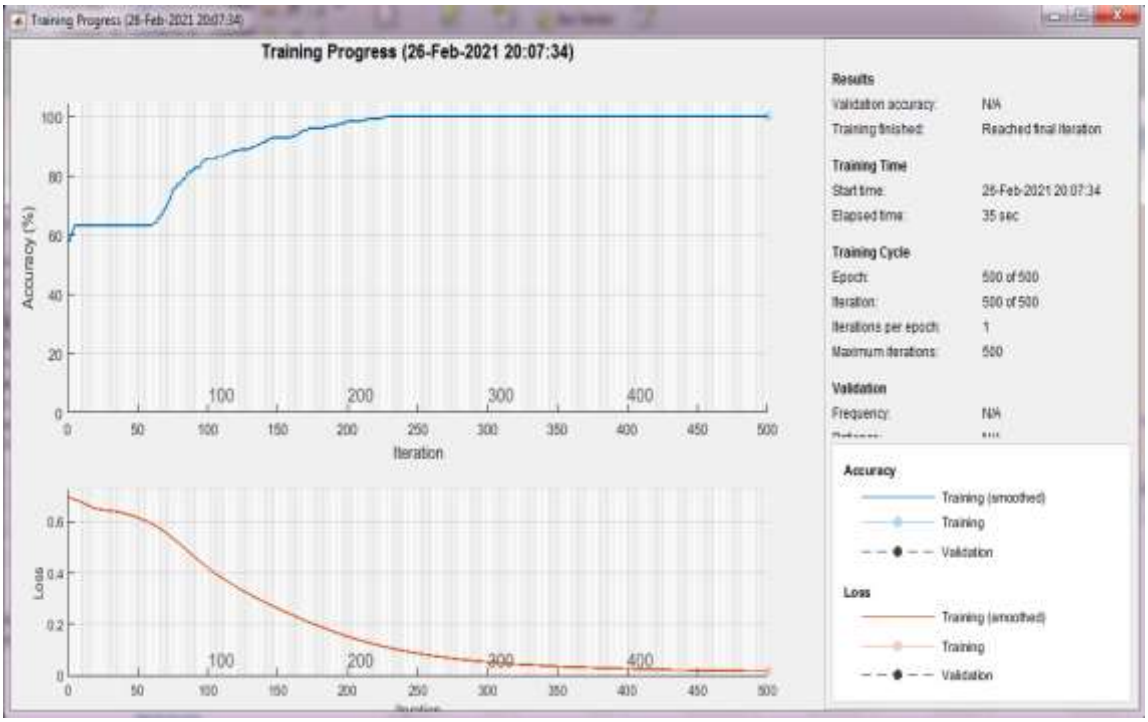

Figure 5- Accuracy and mini batch loss at 500 epochs

4.3.5. The Genetic Algorithm With Machine Learning And Deep Learn

After the previous algorithms (CNN, LSTM, PNN, and KNN) were applied separately, now the algorithms (CNN, LSTM, PNN, and KNN) will be combined with genetic algorithm to obtain better results. Where through genetic we choose the most effective columns in the classification process. As mentioned previously, the DNA dataset consists of 72 columns (71 attribute, 1 Label). By the genetic algorithm, we will choose the best 20 columns and that is as follows, at first we will create 1000 chromosomes and each chromosome contains 20 cells and each cell carries the label of the attribute of dataset. After calculating the fitness function of each chromosome, two chromosomes are randomly selected then crossover of the two chromosome (single point) and then the mutation occurs to specialize from the repeated columns with new columns (unique). This process continues until it reaches 100 turns. Note that the efficiency (accuracy) of the chromosome is calculated after applying PNN, KNN, LSTM , or CNN. Result will be obtained as shown in Table(9):

5. Results Discussion

After applying the system to the previously mentioned data set by divided it into (80% for training and 20% for testing) and comparing the results of its classification with the test data set, the four algorithms KNN, CNN, PNN or LSTM achieved this accuracy, 0.73559, 0.9595, 0.91864 and 0.61017, respectively. While achieving better accuracy when applying the same algorithms after combining them with the genetic algorithm. The accuracy ratio was arranged as follows: GA-KNN, GA-CNN, GA-PNN, GA-LSTM 0.92881, 0.97627, 0.94915 and 0.8339.

As for the rest of the comparison results, they are mentioned in Tables (8,9) and Figure (6) .

Table 8-Results obtained when applying algorithms KNN, CNN, PNN, and LSTM

Header	KNN	CNN	PNN	LSTM
'TP'	150	173	180	140
'TN'	67	111	91	30
'FP'	30	8	0	40

'FN'	48	4	24	85
'ACC'	0.73559	0.9595	0.91864	0.61017
'TPR'	0.75758	0.9774	0.88235	0.622
'FPR'	0.30928	0.0672	0	0.571
'TNR'	0.69072	0.9328	1	0.42
'FNR'	0.24242	0.0226	0.11765	0.37
'Precision'	0.83333	0.9558	1	0.77

Table 9-The results obtained when applying the algorithms in combination with the genetic algorithm

Header	Combine Genetic with			
	KNN	LLCNN	LLPNN	LLLSTM
'TP'	168	124	180	152
'TN'	106	46	100	94
'FP'	12	56	0	28
'FN'	9	69	15	21
'ACC'	0.92881	0.97627	0.94915	0.8339
'TPR'	0.94915	0.64249	0.92308	0.87861
'FPR'	0.10169	0.54902	0	0.22951
'TNR'	0.89831	0.45098	1	0.77049
'FNR'	0.050847	0.35751	0.076923	0.12139
'Precision'	0.93333	0.68889	1	0.84444

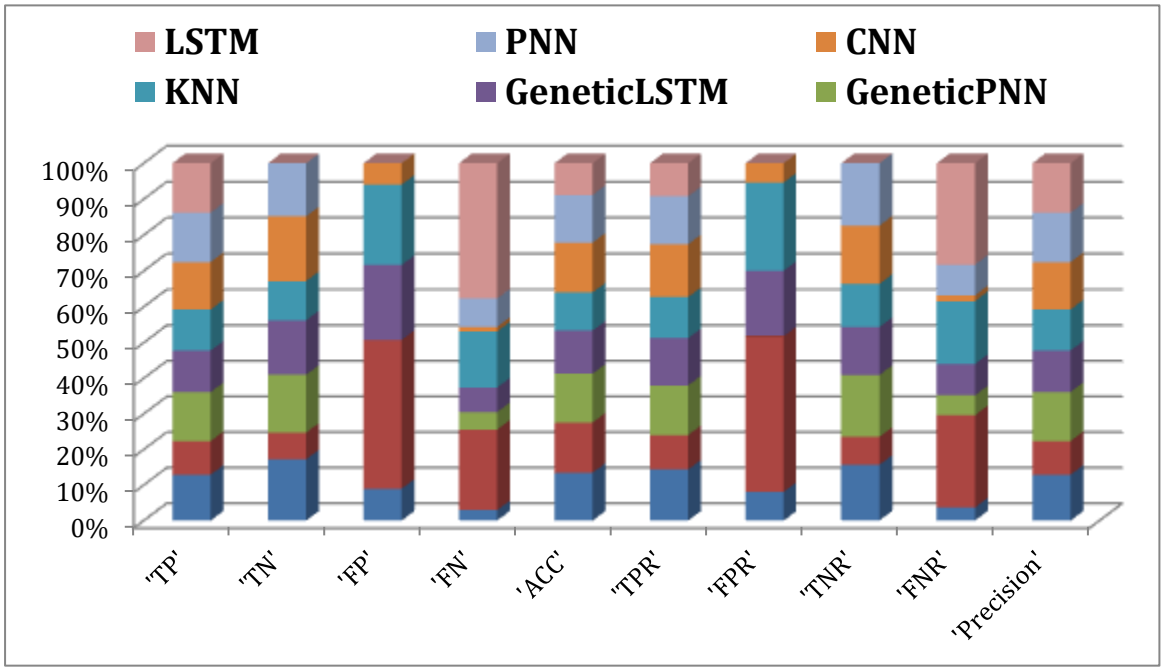

Figure 6- Accuracy of all algorithms

6. .Conclusion

The proposed system is a robust methodology for detecting and classifying breast DNA genes with better accurate results. The goal was to build a complete system to assist the physician or radiologist in classifying breast gene expression cancer.

In this work, we applied different number of gene missing packing value to a breast cancer DNA dataset.

This work was done to obtain better results using two different systems proposed.

Reference

[1] J. e. a. Reimand ،""Pathway enrichment analysis and visualization of omics data using g: Profiler, GSEA, Cytoscape and EnrichmentMap،" Nature protocols ،pp. 482-517 ،2019 .

[2] A. E. E. T. A.-S. a. N. I. G. Hassanien ،"Computational intelligence techniques in bioinformatics،" Computational biology and chemistry ،pp. 37-47 ،2013 .

[3] Y. Y. B. ,. E. S. ,. M. K. T. Umit Atila ،"Classification Of DNA Damages On Segmented Comet Assay Images Using Convolutional Neural Network،" Computer Methods and Programs in Biomedicine ،p. 105192 ،2020 .

[4] M. M.-B. J. M. L.-R. J. M. G.-H. J. G.-G. ´. e. J. C. R.-S. Laura Mac´ıas-Garc´ıa ، "Autoencoded DNA Methylation Data to Predict Breast Cancer Recurrence،" Machine Learning Models ،p. 1019768 ،2020 .

[5] S. S. T. C. A. Neelam Goel ،"An improved method for splice site prediction in DNA sequences using support vector machines،" Procedia Computer Science ،57 ،p. 358 – 367 ،2015 .

[6] G. A. ,. A. A. Dimitrios Mantzaris ،"Genetic algorithm pruning of probabilistic neural networks in medical disease estimation،" Neural Networks ،pp. 831-835 ،2011 .

[7] H. A. Ziad Sankari ،"Probabilistic neural networks for diagnosis of Alzheimer's disease using conventional and wavelet coherence،" Journal of Neuroscience Methods ،pp. 165-170 ،2011 .

[8] B. A. S. Yasha Zeinali ،"Competitive probabilistic neural network،" Department of Civil and Environmental Engineering, Southern Methodist University, Dallas, TX, USA , Integrated Computer-Aided Engineering ،1 ،pp. 1-14 ،2017 .

[9] S. V. M. ,. S. R. Mohammad Bazmara ،"KNN Algorithm for Consulting Behavioral Disorders in Children،" Journal of Basic and Applied Scientific Research December ،2013 .

[10] D. S. H. B. K. B.-J. A. A. K. Travers Ching ،"Opportunities and obstacles for deep learning in biology and medicine،" journal of the royal society interface ،2018 .

[11] J. S. Sepp Hochreiter ،" Long Short-Term Memory،" Neural Computation ،p. 1735–1780 ، 1997 .

[12] R. G. A. K. S. Jitendra Kumara ،"Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters،" Procedia Computer Science ،pp. 676-682 ،2018 .

[13] L. G. Jianxin Wu ،Introduction to Convolutional Neural Networks ،China: National Key Lab for Novel Software Technology, Nanjing University ،2017 .

[14] Y. D. H. L. J. v. V. H. D. A. A. M. H. Marc J van de Vijver ،"A gene expression signature as predictor of survival in breast cancer،" The New England Journal of Medicine ،2002.