Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 2580 – 2599
© TÜBİTAK
doi:10.3906/elk-2105-43

A hybrid convolutional neural network approach for feature selection and disease classification

Prajna Paramita DEBETA*, **Puspanjali MOHAPATRA**
Department of Computer Science and Engineering, International Institute of Information Technology
Bhubaneswar, India

Received: 06.05.2021 • **Accepted/Published Online:** 09.08.2021 • **Final Version:** 04.10.2021

Abstract: Many researchers have analyzed the high dimensional gene expression data for disease classification using several conventional and machine learning-based approaches, but still there exists some issues which make this task nontrivial. Due to the growing complexities of the unstructured data, the researchers focus on the deep learning approach, which is the latest form of machine learning algorithm. In the presented work, a kernel-based Fisher score (KFS) approach is implemented to extract the notable genes, and an improvised chaotic Jaya (CJaya) algorithm optimized convolutional neural network (CJaya-CNN) model is applied to classify high dimensional gene expression or microarray data. This model is tested on two binary class and two multi class standard microarray datasets. Here, the presented hybrid deep learning model (KFS based CJaya-CNN) has been compared with other standard machine learning classification models like CJaya hybridized multi-layer perceptron (CJaya-MLP), CJaya hybridized extreme learning machine (CJaya-ELM), and CJaya hybridized kernel extreme learning machine (CJaya-KELM). The suggested model is evaluated by classification accuracy percentage, number of significant genes selected, sensitivity and specificity values with receiver operating characteristic (ROC) curves. Eventually, the experimental outcomes obtained from the presented model has also been compared with the recent existing feature selection and classification models for a suitable research in analysing high dimensional microarray data. The presented model offered the classification accuracy percentage of 98.2, 99.96, 99.78, and 99.87 for colon cancer, leukemia, lymphoma-3, and small round blue cell tumor (SRBCT) datasets, respectively. All the experimental outcomes reveal that the KFS based CJaya-CNN model is outperforming. Hence, the presented method can be used as a dependable framework for disease classification.

Key words: High dimensional gene expression data, deep learning approach, improvised meta-heuristic algorithm, kernel-based Fisher score, convolutional neural network

1. Introduction

Analysis of gene expression has become an essential part of biological research as genes contain all the basic biological, functional, and structural information of living beings. With the advent of new technologies in this era, DNA microarray is an attractive tool for molecular diagnostic testing of thousands of genes at a time [1]. This technology facilitates the researchers to explore which genes are pointed out absolutely within a tissue under different circumstances. However, researchers have to face many challenges to reveal new information from gene expression or microarray data due to some issues [2] such as the curse of high dimensionality, missing data or imbalanced data, redundant gene issue, retrieval of biological information, and biased by different factors.

*Correspondence: elektrik@tubitak.gov.tr

Due to these issues, a plethora of data mining approaches grabs the attention of analyzing the microarray data [3]. In the analysis of microarray data, diagnosis of disease by classifying the unstructured high-dimensional microarray data is a challenging task. Due to high dimensionality, the classification of this data produces high computational complexity. Therefore, several feature extraction/feature selection techniques with classification models have been proposed by the researchers on various benchmark datasets. However, designing the best classification model is a time (NP)-hard (nondeterministic polynomial-time) problem. Hence, an opportunity always lies in implementing new algorithms in this area. In this work, a hybridized deep learning-based approach is suggested to classify the high-dimensional microarray data. Deep learning is the latest machine learning algorithm that grabs the attention of classifying diseases from the gene expression data [4]. This is based on an artificial neural network (ANN) with multiple hidden layers and has found considerable traction for many bioinformatics applications. Among the different deep learning approaches, convolutional neural network (CNN) model is proved [5] as suitable for handling the unstructured high dimensional data. In this paper, a kernel Fisher score-based (KFS) feature selection algorithm is implemented to filter the informative genes, and an improvised version of the Jaya optimization algorithm [6] viz., chaotic Jaya (CJaya) is implemented to optimize the parameters of CNN and classify the gene expression or microarray data.

1. Initially, the KFS algorithm is applied to extract the key features/genes.
2. After that, parameters of CNN are optimized by CJaya, and the microarray datasets are classified by CJaya optimized CNN (CJaya-CNN) classifier.
3. As per the author's concern, in high dimensional microarray data classification, the KFS-based CJaya-CNN (KFS-CJaya-CNN) algorithm is employed for the first time where the parameters of CNN are optimized by CJaya.
4. A comparison has been performed between the presented hybrid deep learning model (KFS-CJaya-CNN) and other standard machine learning classification models like CJaya hybridized multi-layer perceptron (CJaya-MLP), CJaya hybridized extreme learning machine (CJaya-ELM), and CJaya hybridized kernel extreme learning machine (CJaya-KELM).

The residual portion of the work is organized as follows: Section 2 covers the related work, and Section 3 gives an abstract view of the suggested model. All supported methods including the algorithm of the proposed work are described in Section 4. Section 5 wraps the setup criteria of experimentation, and Section 6 discusses the experimental result part. Finally, the concluding portion and the future scope are discussed in Section 7.

2. Related work

Several researchers have been proposed various feature selection techniques and robust classification models to classify the high dimensional microarray data and diagnose the diseases efficiently. Some wrapper methods use several metaheuristic algorithms enfolded with machine learning approaches for the selection of the most significant features viz., cat swarm optimization (CSO) wrapped kernel ridge regression (KRR) [7], fuzzy backward feature elimination (FBE) wrapped support vector machine (SVM) [8], particle swarm optimization (PSO) wrapped k-nearest neighbor (KNN) [9], genetic algorithm (GA) wrapped extreme learning machine (ELM) [10], genetic bee colony (GBC) wrapped SVM [11], multi-swarm optimization algorithm wrapped SVM [12], GA wrapped SVM [13], Markov blanket (MB) wrapped naïve bayes (NB) [14], artificial bee colony (ABC) wrapped SVM [15], ensemble of error-correcting output codes (HE-ECOC) wrapped SVM [16], and distributed ranking filter (DRF) wrapped correlation-based feature selection (CFS) [17]. The wrapper methods reveal the interaction between the genes and improves the efficiency of the gene selection approaches.

Different hybrid classification approaches have been used to select the relevant features and classify the microarray data efficiently. A MapReduce (MR) based fisher score (FS) feature selection technique and MR-Probabilistic neural network (PNN) have been applied by S. K Baliarsingh et al. [18] to classify genomic data. Another MapReduce feature selection technique with MapReduce SVM has been applied by M. Kumar et al. [19] in the field of genomic data classification. P. Mohapatra et al. [7] applied cat swarm algorithm optimized kernel ridge regression approach to classify the microarray data efficiently. To classify leukemia and colon cancer microarray data, Wang et al. [20] presented adaptive elastic net with conditional mutual ınformation (AEN-CMI) approach. Random forest and a fuzzy decision tree algorithm have been used by Diaz and Ludwig [21, 22], respectively to classify microarray data. Medjahed et al. [23] suggested binary dragonfly (BDF) with SVM-Recursive Feature Elimination (SVM-RFE) approach for feature selection and microarray data classification. A hybrid approach of stacked autoencoder with CNN model has also been used for gene expression data classification [24]. S. Kilicarslan et al. [5] proposed a hybrid model of ReliefF with CNN for genomic data classification. A multi-task deep learning (MTDL) algorithm has also been applied by Liao [25] in this field. Zeebaree et al. [26] also implemented CNN approach for microarray data classification, but this model did not prove its superiority over traditional machine learning models in all microarray dataset. K. Polat [27] presented a kernel-based Fisher score feature selection approach for medical data classification. Erik et al. [28] implemented an evolutionary algorithm optimized CNN for classifying the data. Debata et al. [29] have applied the chaotic Jaya algorithm with Kernel ELM (KELM) to select the most informative genes and classify the high-dimensional cancerous data. In this paper, we have reduced the computation time and have improved the performance by using a deep learning approach.

In the above literature survey, all the presented models have used either classical machine learning algorithm or deep learning approach for genomics data classification. In this work, we have presented a comparison between classical machine learning algorithms and a deep learning approach for genomics data classification. Moreover, we have proposed a two-phase hybrid approach, i.e. KFS-based filter for feature selection and chaotic Jaya (CJaya) optimized CNN (CJaya-CNN) model for data classification. The primary aim of this presented work is to help the medical practitioners in diagnosing the diseases from high dimensional microarray data within an effective time and with high accuracy. In the next section, a detailed description of the presented model is given.

3. Suggested model description

In this work, a two-phase hybrid approach is implemented for feature selection and classification of high dimensional microarray data. The overall architecture of the KFS-CJaya-CNN model is depicted in Figure 1. Before processing, the missing cells are loaded with the frequently recurring value of that specific attribute or feature, and all datasets are normalized using min-max normalization [7]. After normalization, the datasets are separated into two parts: training set and testing set. Then, a filter technique, i.e. KFS is used to select the highly relevant attributes/genes. The genes filtered by KFS are redirected to the CJaya-CNN hybrid model for classification. Simultaneously, the random parameters (kernel sizes (KS), padding (P), types of pooling (Po)), and no. of feature maps (NFM) of CNN are optimized by CJaya. Eventually, the KFS-CJaya-

CNN approach is evaluated by the testing set of data with the optimal feature subset, and the outcomes are acquired by classification accuracy percentage (CA%). Moreover, a comparison has been performed between the CJaya-CNN model and other standard machine learning classification models like CJaya-MLP, CJaya-ELM, and CJaya-KELM.

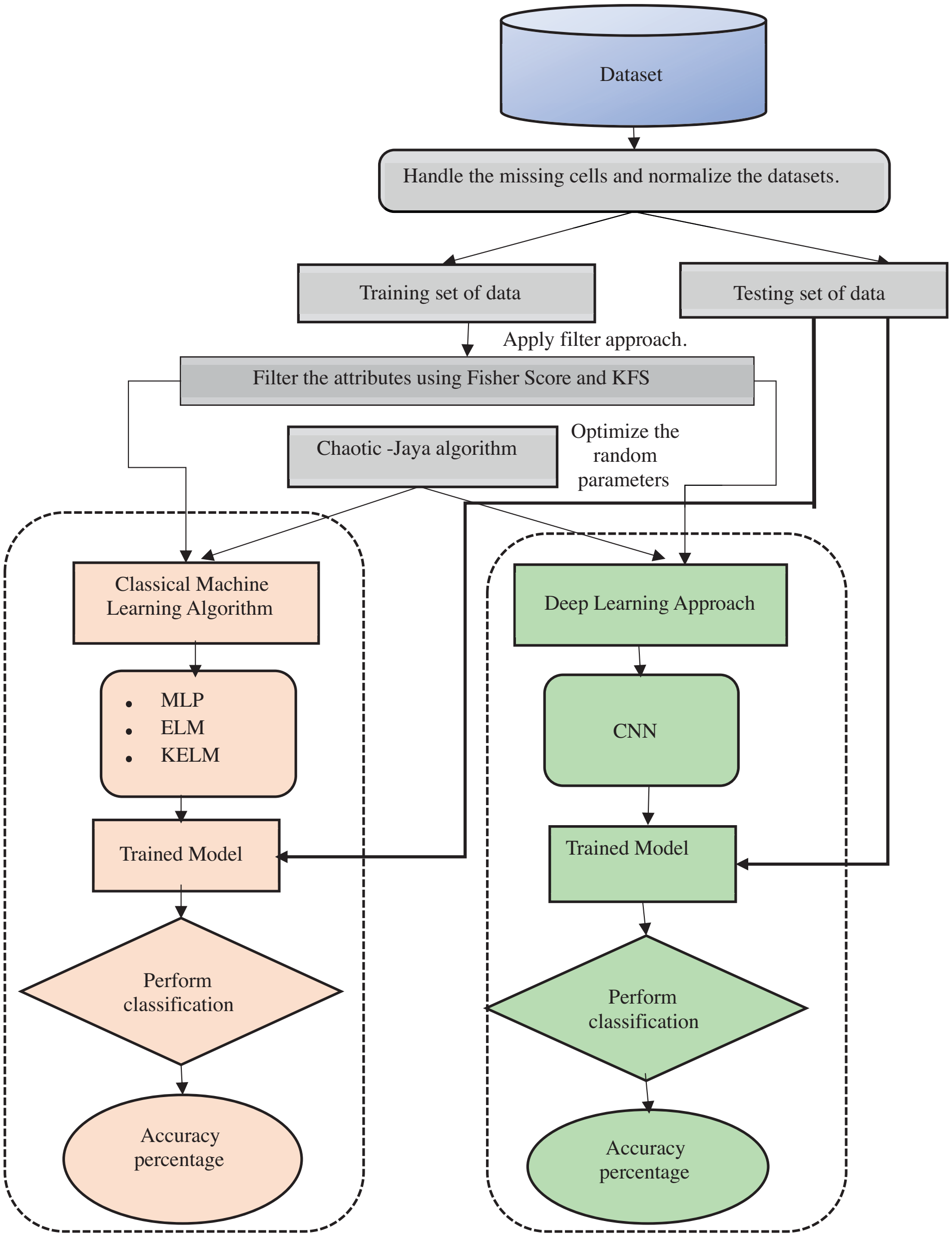


Figure 1. The overall architecture of the KFS-CJaya-CNN model.

4. Supported methodologies

4.1. CNN model

LeCun [30] has presented the CNN model, which has multiple layers. Mostly, this model is applied for image processing and finding the relationships among the attributes of the non-linear data [31]] by hidden layers. CNN model contains six layers such as convolution layers, batch normalization layer, activation layer, max-pooling layer, fully connected layer, and output layer or softmax layer. Here, high-dimensional microarray datasets are classified by one-dimensional CNN architecture. Generally, in artificial neural networks (ANNs), input layer neurons are connected with output layer neurons of the succeeding layer. A bit different from the ANNs, a convolution operation is incorporated into the input layer of the CNN model with the connections. In each layer of CNN, selected filters are used, and the outcomes are merged. During the training phase, the features are learned in each layer. Figure 2. presents the overall structure of the established CNN model. In this model, raw input data is given to the convolution layer after preprocessing. The function of three major layers of the CNN model has been described below:

1. **Convolution layer:** In CNN, the convolution layer extracts the features using filters from the input data as per the stipulated dimension. Here, weights are generated arbitrarily. With these weights, when the convolution process with a 3x3 filter is applied on 1-D data, a new feature map is generated. Till the completion of the entire dataset, this process will continue. RELU activation function is applied to the obtained dataset, which acts as a threshold function. Then, the normalization process is applied to balance the distribution of data, which may change after the convolution process.

2. **Max pooling layer:** After the normalization process, for obtaining better feature maps, the pooling operations are performed in the output layer. Generally, pooling operations are performed to minimize the size of the input in the next stage convolution process. In most of the pooling techniques, the pooling process is carried out by considering the pool size and stride value = 2. In the training phase, for avoiding the overfitting problem during training, the neuron dropout method is applied. The value of the neuron dropout is considered as 0.2. The density of the neuron is set to 1024 for connecting with the previous layer neurons and performing classification with the fully connected layer.

3. **Softmax layer:** At last, the probability-based softmax function is used to improve the classification accuracy. The softmax function normalizes the output values and transfers these values into probability values. Finally, the test data is classified based on these probability values.

The major difference between the CNN model and classical models is that CNN performs the classification process with fewer steps than classical methods. The CNN approach also performs a better estimation of important parameters that one needs to define in classical classification approaches. The CNN model produces fixed-size inputs and outputs. Instead of Recurrent neural networks (RNN), CNN is chosen for classification from the deep learning approaches because the memory size required for a high dimensional dataset can be handled more efficiently by CNN, which results in a higher classification accuracy rate. As CNN uses the ReLU activation function, the vanishing gradients problem of RNN is solved [5].

4.2. Gene selection by kernel based Fisher score (KFS)

In basic Fisher score (FS) approach, the FS value of a gene is calculated according to Eq. (1). Then, by computing the mean value of all genes FS values, a threshold value (TV) is obtained. The gene, whose FS value

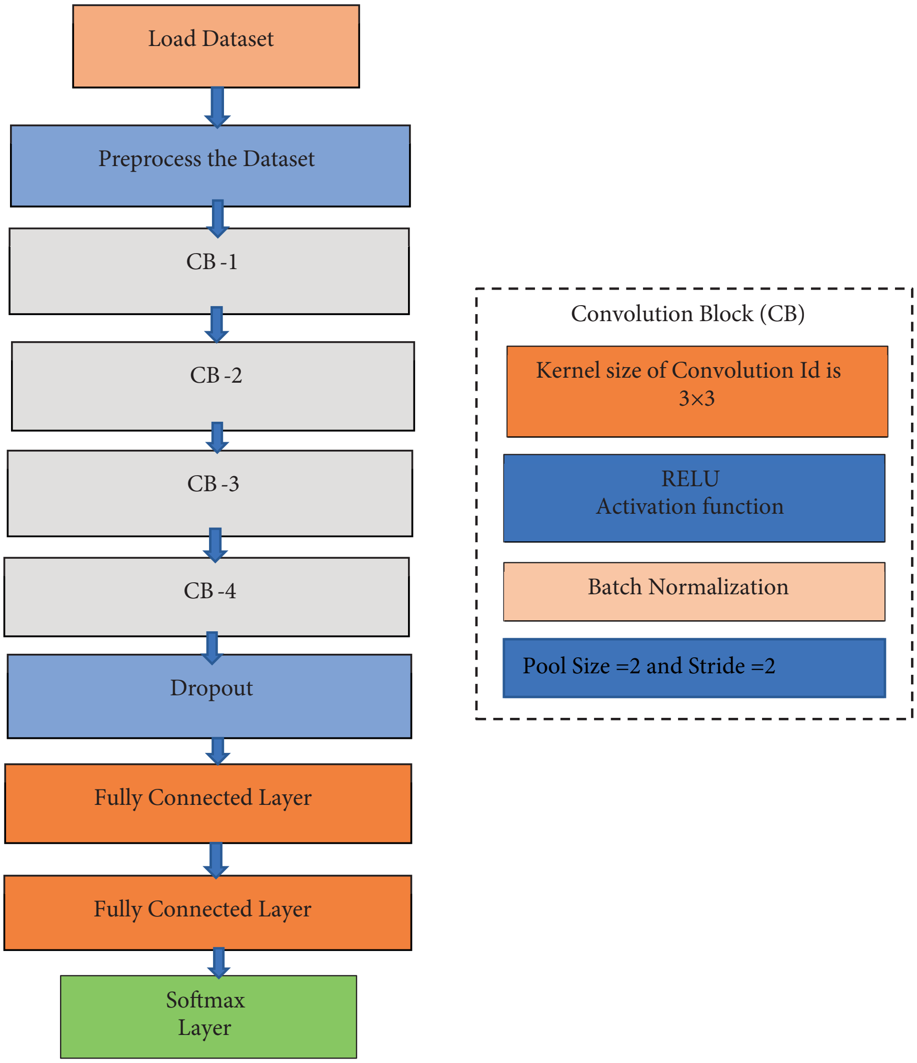


Figure 2. The overall process structure of the CNN model.

is higher than the TV, will be added in the feature space, and the gene whose FS value is lower than the TV will be removed from the feature space.

$$FS(g^i) = ((\bar{y_{\mathrm{i}}}^{(+)} - \bar{y_{\mathrm{i}}})^2 + (\bar{y_{\mathrm{i}}}^{(-)} - \bar{y_{\mathrm{i}}})^2)/(1/(m_+ - 1)\sum_{k=1}^{m_+}(y_{\mathrm{k,i}}{}^{(+)} - \bar{y}_{\mathrm{i}}^{(+)})^2) - (1/(m_{\text{-}} - 1)\sum_{k=1}^{m_{\text{-}}}(y_{\mathrm{k,i}}{}^{(-)} - \bar{y}_{\mathrm{i}}^{(-)})^2)) \quad (1)$$

In Eq. (1), y_{k} is the training vector, m_+ and $m_{\text{-}}$ are the number of +ve and -ve instances, respectively, $\bar{y_{\mathrm{i}}}$ is the i^{th} gene or attribute of the entire datasets, $\bar{y_{\mathrm{i}}}^{(+)}$ is the i^{th} gene or attribute of the +ve datasets and $\bar{y_{\mathrm{i}}}^{(-)}$ is the i^{th} gene or attribute of the -ve datasets. Similarly, $y_{\mathrm{k,i}}{}^{(+)}$ is the i^{th} gene or attribute of the k^{th} +ve instances and $y_{\mathrm{k,i}}{}^{(-)}$ is the i^{th} gene or attribute of the k^{th} -ve instances. In basic FS, the mutual information (MI) among genes is not considered, which is a major demerit [32]. Kernel-based Fisher score (KFS) [27] performs both transformation of non-linearly separable dataset into linearly separable dataset and reduces the cost of computational overhead. The steps followed by KFS are summarized in algorithm 1. The main advantage of

```
Algorithm 1: Kernel based Fisher score (KFS).
Input: Normalized dataset
Output: Reduced dataset with significant feature subset
Begin.
Calculate The input features spaces of dataset are mapped to kernel space by using kernel function
  such as radial basis function (RBF) or linear functions.
After mapping, the FS values of the datasets having high dimensional feature space are determined
  by Eq.1.
After that, the mean of the estimated FS is calculated, and the calculated result is considered as TV.
Finally, the feature, whose FS value is higher than the TV, will be merged in the feature space and
  the gene whose FS value is lower than the TV will be removed from the feature space.
End.
```

using KFS is that the insignificant genes are extracted from high dimensional input feature space due to the transformation of the dataset to high dimensional feature space using a kernel function.

4.3. Jaya algorithm with chaotic learning method

Jaya algorithm [6] does not need any algorithm-specific parameters. The main advantage of this algorithm is that it is implemented with less computational complexity and less computational time. The step-wise description of the Jaya algorithm is given in algorithm 2.

```
Algorithm 2: Jaya optimization algorithm.
Input: No. of iterations (I), Size of population (P), and design variables (DV)
Output: G-best (Global best) solution
1 Begin.
2 Initialize the value for DV, P and I (stopping limit).
3 Obtain the best and worst solutions among the specified population.
4 The result as per the best and worst solution will be modified by using Eq. 2.

      $.X_{\mathrm{m,n,i}}{}^{(')} = X_{\mathrm{m,n,i}} + r_{1,\mathrm{m,i}}[(X_{\mathrm{m,best,i}}) + |(X_{\mathrm{m,n,i}})|] - r_{2,\mathrm{m,i}}[(X_{\mathrm{m,worst,i}}) + |(X_{\mathrm{m,n,i}})|]$   (2)

    During $i^{th}$ iteration, for the $n^{th}$ candidate, $X_{\mathrm{m,n,i}}$ shows the value of the $i^{th}$ variable. In Eq. (2),
    $n$ is the size of the population, $i$ represents no. of iteration and $m$ represents the design variables.
5 Then, compare the current solution and modified solution. If the modified one is better than the
    previous solution, then modified solution will be kept, otherwise the previous one will be stored.
6 Continue Steps 3 to 5, till the stopping criteria is touched.
7 End.
```

In the presented work, chaotic-Jaya (CJaya) [33] algorithm is applied, which is the improved version of Jaya algorithm. This algorithm is interpreted with chaos concept. Integration of chaos concept improves the convergence speed of the algorithm faster and produces the better exploration [34] of the search space without stopping in local optima. Mathematically, the term Chaos is coined as the randomness of a deterministic dynamical system. To interpret chaos theory in different optimization algorithms, different chaotic maps with different mathematical equations are used in various optimization algorithms to define chaos theory. In the presented work, the logistic map function is interpreted for creating chaotic random numbers due to its simplicity.

This logistic map function is clarified by Eq. (3).

$$k_{\mathrm{t+1}} = 4.k_{\mathrm{t}}.(1 - k_{\mathrm{t}}) \tag{3}$$

where k_{t} is the observed result of the chaotic map at i^{th} iteration. The working principle of the CJaya algorithm is the same as the Jaya algorithm. The main difference between CJaya and Jaya algorithms is that the random numbers in the CJaya algorithm are created by using a chaotic random number creator. In the Jaya algorithm, the two random variables (r_1 and r_2) are exchanged by the logistic chaotic variables. The solution is modified by Eq. (4).

$$X'_{\mathrm{m,n,i}} = X_{\mathrm{m,n,i}} + k_{\mathrm{t,m,i}}[(X_{\mathrm{m,best,i}}) + |(X_{\mathrm{m,n,i}})|] - k_{\mathrm{t,m,i}}[(X_{\mathrm{m,worst,i}}) + |(X_{\mathrm{m,n,i}})|] \tag{4}$$

In Eq. (4), t defines the iteration size, k_{t} shows the value of i^{th} chaotic iteration, whereas, in the beginning, k_0 value is randomly taken between [0,1].

4.4. CJaya optimized CNN (CJaya-CNN) Algorithm

In this presented work, the random parameters (KS, P, NFM, and Po) of CNN are optimized by CJaya. As the parameter values are integers, the search of a parameter is carried out by using a floating-point value, then this value is rounded. The parameter value is specified within a dynamic range. If the parameter values will exceed the range, then these are returned to their specified dynamic range. In this work, the dynamic ranges of these variables are defined in Table 1. Figure 3 presents a pictorial representation while algorithm 3 discusses the step-wise flow of the suggested algorithm.

Table 1. The parameters which are to be optimized and their dynamic ranges.

Name of the layer	**Name of the hyper parameter**	**Range of the parameters**
Convolution layer	No. of feature maps (NFM)	50 to 200
	Pad size (P)	0 to 7
	Size of the kernel (KS)	1 to 8
Pooling	Pooling algorithm (Po)	Max, Ave
	Size of the kernel (KS)	1 to 8

4.5. Computational complexity of the presented methods

The computational complexities of the presented methods viz., KFS, CJaya, and CNN are described in Table 2. In Table 2, I appear as the number of iterations, T shows the training instances, and F presents the no. of features in the expression of KFS complexity. The time complexity for modifying the locations of the solutions in CJaya depends upon of the population and the dimension of the dataset. In the expression of computational complexity of CJaya, N represents the size of the population, and X represents the dimension of the dataset. In the expression of computational complexity of CNN, D represents the number of layers in convolution layer, L^{th} means layer, M^{L-1} represents no. of inputs in the L^{th} layer, F^L represents no. of filters, S^L represents size of the filter, and A^L represents the size of the attribute map.

Algorithm 3: CJaya-CNN.

```
    Input: Population size (N), no. of iterations (I), upper and lower bound for kernel sizes (KS),
           padding (P), number of feature maps (NFM), and types of pooling (Po).
    Output: Classification accuracy percentage (CA%)
 1  Begin.
 2  Initialize $N$, $Itr$, $KS$, $P$, $NFM$ and $Po$.
 3  For each solution, find out the fitness $(F)$ value (CA% using CNN with the initialised value of $N$,
      $Itr$, $KS$, $P$, $NFM$ and $Po$ parameters, and preselected features by KFS.
 4  Arrange the $F$ values in sorted (descending) manner. Get the best and worst values among them
 5  Sort the population $(N)$ as per the index of the sorted $F$ value
 6  Place the best value of $F$ and position as the fitness value and location of the solution, respectively.
 7  Estimate the mean of the $F$ values.
 8  while $I < Max_I$ do
 9    if $I == 1$ then
10      Compute the $k_{\mathrm{m}}$ (chaotic map), by interpreting Eq. 3
11      Modify the values $r1$ and $r2$ (two random values of Jaya algorithm) in Eq.2 employing the Eq.3
12    for $j = 1 : N$ do
13    Modify the position of the solution with $N$, $Itr$, $KS$, $P$, $NFM$ and $Po$ by using Eq.4.
14    end for
15    else
16      if $(cur_{m}ean_{F}prev_{m}ean_{F})/curr_{m}ean_{F} > 0.001$, then
17        Repeat Steps 10 to 14
18      else
19      break.
20      end if
21    end if
22    for every modified candidate solution do
23      Test the $LB$(lower bound) and $UB$(upper bound) for solution position, $KS$, $P$, $NFM$, $Po$.
24      Repeat Steps 2 to 3 for obtaining the $new_F$ values
25      if $cur_F > prev_F$, then
26        update the F of the solution.
27        Update the solution position, $KS$, $P$, $NFM$ and $Po$.
28      else
29        Keep the $F$ value of previous solution.
30        Keep the solution position, $KS$, $P$, $NFM$ and $Po$ of the previous one.
31      end if
32      Continue Steps 4 to 7
33    end for
34  end while
35  Obtain the final $F$ value (CA%).
36  End
```

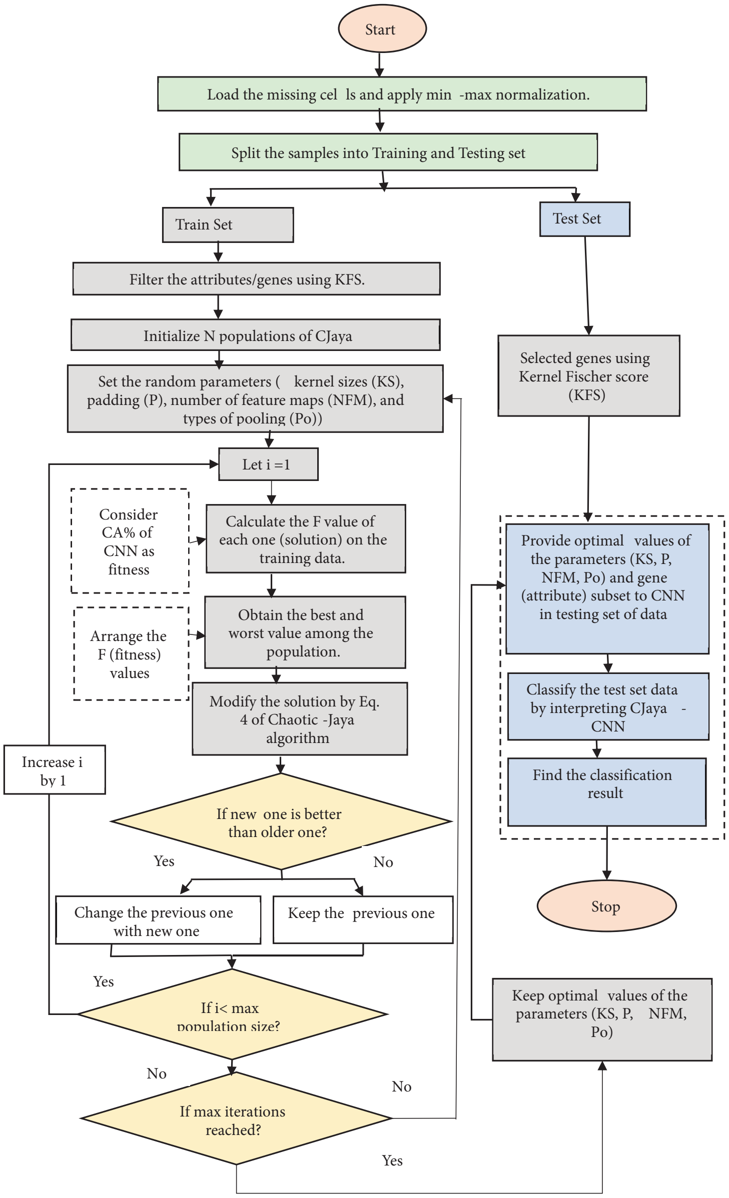

Start
Load the missing cells and apply min-max normalization.
Split the samples into Training and Testing set
Train Set
Test Set
Filter the attributes/genes using KFS.
Initialize N populations of CJaya
Set the random parameters (kernel sizes (KS), padding (P), number of feature maps (NFM), and types of pooling (Po))
Selected genes using Kernel Fischer score (KFS)
Let i =1
Consider CA% of CNN as fitness
Calculate the F value of each one (solution) on the training data.
Provide optimal values of the parameters (KS, P, NFM, Po) and gene (attribute) subset to CNN in testing set of data
Arrange the F (fitness) values
Obtain the best and worst value among the population.
Classify the test set data by interpreting CJaya-CNN
Modify the solution by Eq. 4 of Chaotic-Jaya algorithm
Find the classification result
Increase i by 1
If new one is better than older one?
Yes
No
Stop
Change the previous one with new one
Keep the previous one
Yes
If i< max population size?
Keep optimal values of the parameters (KS, P, NFM, Po)
No
No
If max iterations reached?
Yes

Figure 3. Flowchart of the presented model.

Table 2. Computational complexity of the presented methods.

Model used	**Computational complexity**
KFS	$O(I * T * F)$
CJaya	$O(N * X)$
CNN	$O(\sum_{L=1}^{D} M_{\text{L-1}} * F_{\text{L}} * S_{\text{L}} * A_{\text{L}})$

5. Setup for experimentation

5.1. Configuration for experimentation

Here, the experiments are carried out in the following environment: Ubuntu 14.04 (operating system), Processor: $Intel^R$ $Core^{TM}$ i3-8100 CPU (3.60 GHz), Memory: 8 GB RAM and Language Used: Python 2.7 (64 bits).

5.2. Detailed description of the datasets

Here, all the experiments are carried out with two binary-class and two multiclass microarray datasets. The colon cancer dataset consists of 62 samples with 2000 genes (or attributes). It is collected from colon cancer patients. In 62 samples, 40 samples belong to malignant tumors, and 22 samples are found healthy. This dataset consists of 6500 attributes (or genes). Among them, 2000 genes are considered according to the confidence in the measured expression levels. The Leukemia dataset contains 72 samples with 7129 genes. In 72 samples, 25 samples are found as acute myloid leukemia (AML), and 42 samples belong to acute lymphoblastic leukemia (ALL). The lymphoma-3 dataset consists of 62 samples with 4026 genes. Among 62 samples, 42 samples are belonging to DLBCL (diffuse large B-cell lymphoma), 9 samples belong to FL (follicular lymphoma), and 11 samples belong to CLL (chronic lymphocytic leukemia). The SRBCT (small round blue cell tumor) dataset contains 83 samples with 2308 genes. This dataset has four classes such as neuroblastoma (NB), burkitt lymphoma (BL), rhabdomyosarcoma (RMS), and Ewing family of tumors (EWS). Among 83 samples, 29, 11, 18 and 25 samples belong to EWS, BL, NB and RMS classes, respectively. Table 3 summarizes the description of the four standard microarray datasets.

Table 3. Detailed description of the four standard microarray datasets.

Dataset name	**Sample size**	**Feature size**	**No. of classes**
Colon tumor [35]	62	2000	2
Leukemia [36]	72	7129	2
Lymphoma-3 [37]	62	4026	3
SRBCT (small round blue cell tumor) [37]	88	2308	4

5.3. Parameter initialization

A comparison has been performed between the CJaya-CNN model and other standard machine learning classification models like CJaya hybridized multi-layer perceptron (CJaya-MLP), CJaya hybridized extreme learning machine (CJaya-ELM), and CJaya hybridized kernel extreme learning machine (CJaya-KELM). Table 4 gives the detailed description of initialization values of parameters for all the supported algorithms.

5.4. Evaluation measures

Here, for unbiased experimentation, each dataset is randomly separated in three different forms of training and testing data like 60%–40%, 70%–30%, and 80%—20%. In the current work, mean value is considered as final output from these three tests (i.e. test1(60%–40% partition), test2(70%–30% partition), test3(80%–20% partition)). Here, accuracy percentage, sensitivity [17], and specificity [18] values with ROC [7] are taken as performance evaluating measures.

Table 4. Initialization values of parameters for all the supported algorithms.

MLP	**ELM**	**KELM**	**CNN**	**CJaya**
No. of iterations =100	No. of iterations= 100	$C\text{and}\gamma$ =$[2^7, 2^8, \ldots, 2^7, 2^8]$	Dropout (FC) = 50%	No. of population=100
No. of Hidden layers =3	No. of nodes in the hidden layer =15	Activation function = Sigmoid	Activation function = ReLU	No. of iterations =100
No. of Nodes in each Hidden layer=5	-	-	Output function = Sofmax	-
-	-	-	Batch size = 64	-
-	-	-	Momentum = 0.9	-
-	-	-	Learning rate = 0.001	-
-	-	-	Training Epochs = 100	-

6. Experimental result and discussion

Here, the experiments have been carried out with a hybrid method of dimensionality reduction and an optimized deep learning approach (KFS-based CJaya-CNN model) for the diagnosis of the disease on four high dimensional gene expression datasets. Initially, top-ranked feature subsets are selected by the KFS approach from each dataset, then these features are forwarded to the CJaya optimized CNN model for classification. Here, we have performed three tests (i.e. test1(60%–40% partition), test2 (70%–30% partition), test3 (80%–20% partition)), and the accuracy percentage (ACC%), sensitivity and specificity values of each test are recorded. The mean values of these tests are considered as the final output of accuracy, sensitivity and specificity. Table 5 and 6 show the experimental result of ACC%, sensitivity, specificity values with the number of selected features (NSF) by FS and KFS approach in binary class and multi class datasets, respectively. In tumor dataset colon, FS and KFS techniques select 80 and 50 features and obtain 94.62% and 98.2% accuracies, respectively. In Leukemia dataset, FS and KFS techniques select 120 and 80 features and obtain 97.02% and 99.96% accuracies, respectively. In Lymphoma-3 dataset, FS and KFS techniques select 40 and 30 features and obtain 97.78% and 99.78% accuracies, respectively. In SRBCT dataset, FS and KFS techniques select 70 and 40 features and obtain 97.67% and 99.87% accuracies, respectively. From Tables 5 and 6, it is observed that the KFS-based CJaya-CNN model gives high accuracy with a smaller number of features than FS based CJaya-CNN model. The sensitivity and specificity values are also higher in the suggested approach, which reveals that both positive and negative samples are well classified.

Table 7 represents comparison of results among the KFS based machine learning approach and proposed deep learning method (KFS-CJaya-CNN) in four microarray datasets. According to Table 7, four microarray datasets (i.e. colon tumor (98.2%), Leukemia (99.96%), Lymphoma-3 (99.78%), SRBCT (99.87%)) give better accuracy in suggested deep learning-based approach.

The recorded time for feature selection and classification is shown in Table 8. It is clear from Table 8 that the total time taken in the KFS-based CJaya-CNN model is higher than FS based CJaya-CNN. As the ACC%, sensitivity, and specificity rate of the KFS-based CJaya-CNN model are higher than others; it can be considered as the best model for microarray data classification.

Table 5. A comparison of experimental results found in two binary class datasets.

Dataset	**Feature selection with NSF**	**Classification method**	**Experiment**	**Acc%**	**Sensitivity %**	**Specificity %**
Colon tumor	FS (80)	CJaya-CNN	Test1(60%–40%)	93.56	82.01	83.96
			Test2(70%–30%)	94.85	94.8	89.45
			Test3(80%–20%)	95.45	94.90	90.9
			Mean	94.62	90.57	88.1
	KFS (50)	CJaya-CNN	Test1(60%–40%)	96.15	82	84
			Test2(70%–30%)	98.82	95.60	89.5
			Test3(80%–20%)	99.65	95.8	91
			Mean	**98.2**	**91.13**	**88.16**
Leukemia	FS (80)	CJaya-CNN	Test1(60%–40%)	96.7	84	84.9
			Test2(70%–30%)	96.56	96.90	90.49
			Test3(80%–20%)	97.82	97.80	91.99
			Mean	97.02	92.90	89.09
	KFS (50)	CJaya-CNN	Test1(60%–40%)	99.89	84.01	85
			Test2(70%–30%)	100	97.60	90.5
			Test3(80%–20%)	100	98.8	91.8
			Mean	**99.96**	**93.47**	**89.1**

Bold values are the best values obtained after evaluation.

Further, Figure 4 presents the convergence graph of CJaya-CNN, CJaya-KELM, CJaya-ELM, and CJaya-MLP approaches in 4 microarray datasets. These figures reveal that there is a successive improvement of accuracies among 1 to 100 iterations in all the datasets. In the Figure 4(a), the accuracy percentage of colon cancer dataset is converging next to 44th, 49th, 68th, and 80th iteration in CJaya-CNN, CJaya-KELM, CJaya-ELM, and CJaya-MLP approaches, respectively. In the Figure 4(b), the accuracy percentage of Leukemia cancer is converging next to 60th, 61st, 74th, and 81th iteration in CJaya-CNN, CJaya-KELM, CJaya-ELM, and CJaya-MLP approaches, respectively. In the Figure 4(c), the accuracy percentage of Lymphoma-3 dataset is converging next to 54th, 60th, 69th, and 73rd iteration in CJaya-CNN, CJaya-KELM, CJaya-ELM, and CJaya-MLP approaches, respectively. In the Figure 4(d), the accuracy percentage of SRBCT dataset is converging next to 46th, 54th, 68th, and 77th iteration in CJaya-CNN, CJaya-KELM, CJaya-ELM, and CJaya-MLP approaches, respectively. According to the above convergence graphs, it is transparent that the rate of convergence of the presented CJaya-CNN approach is notably faster than other approaches due to the interpretation of chaotic theory in the basic Jaya algorithm. Moreover, in Figure 5, the ROC curves have been plotted between the sensitivity and specificity values obtained from the KFS based CJaya-CNN (KFS-CJaya-CNN) and FS based CJaya-CNN (FS-CJaya-CNN) methods in four microarray datasets. According to Figure 5, the suggested hybrid model of feature selection and deep learning approach (KFS-CJaya-CNN) is outperforming. In this paper, the implemented technique has been compared with eighteen existing standard approaches. This comparison cannot offer an incontrovertible conclusion because various techniques use dissimilar evaluating measures and distinct

Table 6. A comparison of experimental results found in two multi class datasets.

Dataset	**Feature selection with NSF**	**Classification method**	**Experiment**	**Acc%**	**Sensitivity %**	**Specificity %**
Lymphoma-3	FS (80)	CJaya-CNN	Test1(60%–40%)	96.77	90.01	83.79
			Test2(70%–30%)	97.82	96.50	90.27
			Test3(80%–20%)	98.76	96.80	92.8
			Mean	97.78	94.43	88.95
	KFS (50)	CJaya-CNN	Test1(60%–40%)	99.66	90.1	84
			Test2(70%–30%)	99.79	97.80	90.7
			Test3(80%–20%)	99.89	98.9	93
			Mean	**99.78**	**95.6**	**89.23**
SRBCT	FS (80)	CJaya-CNN	Test1(60%-40%)	96.5	91.98	85.9
			Test2(70%–30%)	97.85	97.42	95.68
			Test3(80%–20%)	98.67	97.9	95.97
			Mean	97.67	95.77	92.51
	KFS (50)	CJaya-CNN	Test1(60%–40%)	99.75	92	86.1
			Test2(70%–30%)	99.86	98.62	95.7
			Test3(80%–20%)	100	98.9	96
			Mean	**99.87**	**96.5**	**92.6**

Bold values are the best values obtained after evaluation.

Table 7. Comparison of results between the KFS based machine learning approach and proposed deep learning method (KFS-CJaya-CNN) in four microarray datasets.

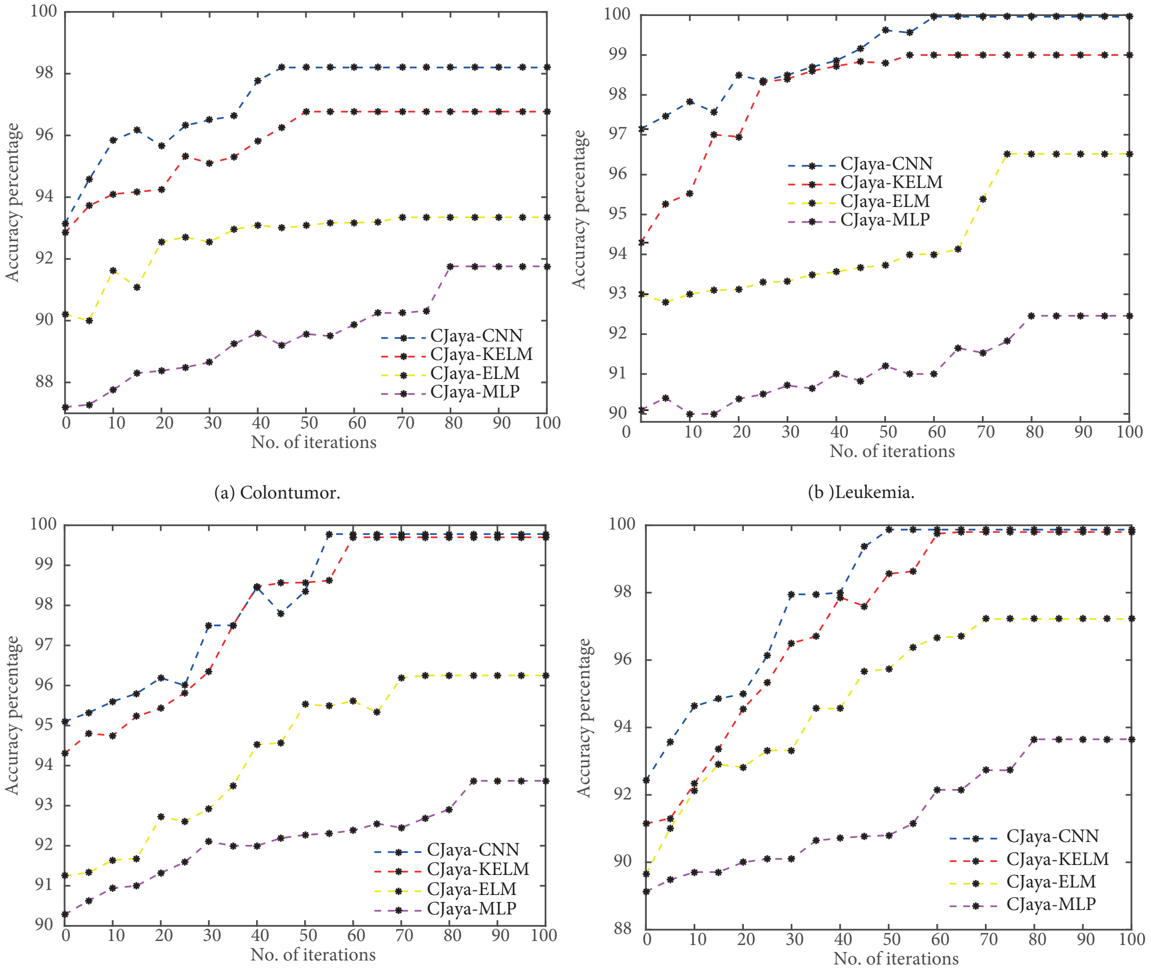
Dataset	**Methods used**	**Acc%**	**Sensitivity%**	**Specificity%**
Colon tumor	CJaya-MLP	91.76	88.75	89.62
	CJaya-ELM	93.35	92.67	92.85
	CJaya-CNN	**98.2**	**97.86**	**97.17**
Leukemia	CJaya-MLP	92.46	91.86	90.75
	CJaya-ELM	96.52	95.82	96.15
	CJaya-CNN	**99.96**	**99.87**	**99.91**
Lymphoma-3	CJaya-MLP	93.62	93.45	92.73
	CJaya-ELM	96.25	97.12	96.34
	CJaya-CNN	**99.78**	**99.51**	**99.39**
SRBCT	CJaya-MLP	93.65	92.83	91.64
	CJaya-ELM	997.22	96.72	96.85
	CJaya-CNN	**99.87**	99.54	**99.52**

Bold values are the best values obtained after evaluation.

datasets to estimate their efficiency. Still, an approximate estimation of the established method with other standard approaches can be presented through comparison especially in this domain. Table 9 presents a comparison between the established method with other standard approaches w.r.t CA% in all datasets.

Table 8. The execution time (in seconds) of feature selection and classification of four microarray datasets.

Dataset	**Methods used**	**Total time is taken (in second) for feature selection and classification**
Colon tumor	KFS-CJaya-CNN	8.65
	FS-CJaya-CNN	7.32
Leukemia	KFS-CJaya-CNN	7.52
	FS-CJaya-CNN	6.35
Lymphoma-3	KFS-CJaya-CNN	8.42
	FS-CJaya-CNN	6.95
SRBCT	KFS-CJaya-CNN	9.56
	FS-CJaya-CNN	8.62



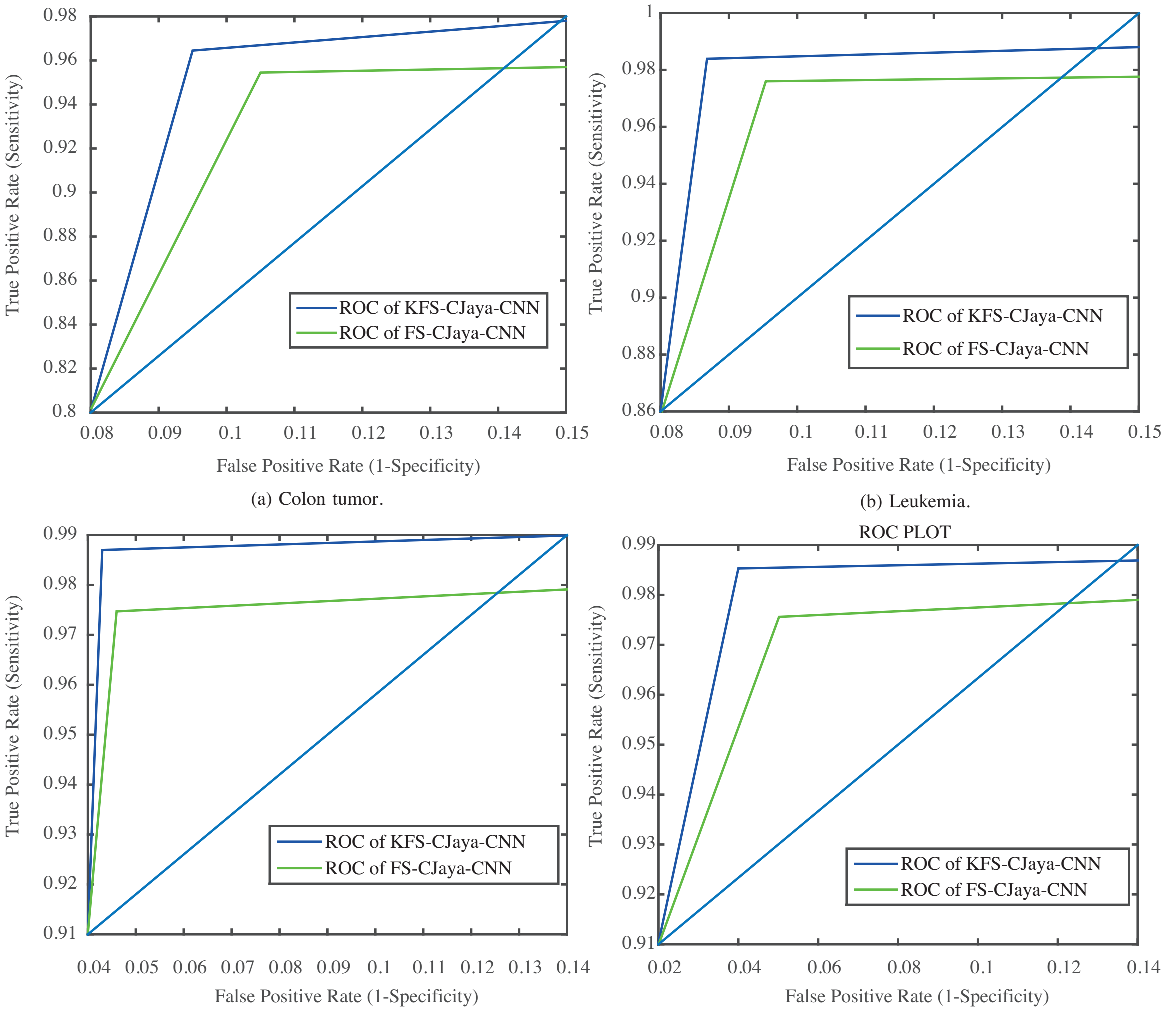
(a) Colontumor.

(b)Leukemia.

(c) Lymphoma-3.

(d) SRBCT.

Figure 4. Convergence graphs.



(a) Colon tumor.

(b) Leukemia.

(c) Lymphoma-3.

(d) SRBCT.

Figure 5. ROC of KFS-CJaya-CNN and FS-CJaya-CNN.

Table 9 gives a transparent view that the presented method (KFS based CJaya-CNN) outperforms in colon tumor (98.2% accuracy), Leukemia (99.96% accuracy), and Lymphoma-3 (99.78% accuracy). However, in the case of SRBCT (99.87% accuracy) datasets, the WCSSA-KELM [38] and Fisher score based WCGWO-mrPNN [18] methods give a better result. In the ovarian cancer dataset, both WCSSA-KELM [38] model and Fisher based WCGWO-mrPNN model give 100% accuracy, whereas only WCSSA-KELM model gives 100% accuracy in SRBCT dataset. Our proposed method stands out due to the following reasons:

(i) Initially, the KFS algorithm is applied to extract the key genes, which perform both transformations of a non-linearly separable dataset into a linearly separable dataset and reduces the cost of computational overhead. The main advantage of using KFS is that the insignificant genes are extracted from high dimensional input feature space due to the transformation of the dataset to high dimensional feature space using a kernel function.

(ii) Here, we have used the Jaya optimization algorithm to optimize the random parameters of CNN. Jaya algorithm does not need any algorithm-specific parameters. The main advantage of this algorithm is that it is implemented with less computational complexity and less computational time. Moreover, the random numbers in the CJaya algorithm are created by using a chaotic random number creator, which reduces the instability of this algorithm.

Table 9. A comparison between the presented model and other models w.r.t CA% in all datasets (The '–' sign shows missing of data) CA% in all datasets.

Methods used	**Colon tumor**	**Leukemia**	**Lymphoma-3**	**SRBCT**
Cat swarm algorithm-KRR [7]	95	95.45	-	85.71
PSO-AKNN [9]	-	-	–	94
GBC-SVM [11]	-	-	98.48	96.38
Multi-swarm-SVM [12]	94.2	98.1	-	-
GA-SVM [13]	91.2	91.5	-	-
IWSS-MB-NB [14]	86	97.1	-	-
mRMR-ABC [15]	-	-	96.96	96.30
D-ECOC [16]	-	-	-	98.7
DRFO-CFS [17]	90	91.18	-	-
FS-WCGWO-mrPNN [18]	95.06	99.21	-	-
AEN-CMI [20]	85.12	83.98	-	-
Fuzzy Decision Tree [22]	-	87.50	-	-
SVM-RFE + BDF [23]	-	95.81	-	-
WCSSA-KELM [38]	95.5	99	99.71	**100**
ReliefF–CNN [5]	84.9	57.9	-	-
SE1D-CNN [24]	-	99.86	-	-
Seven-layer deep learning approach [39]	94.8	99.26	-	-
Laplacian score-CNN [40]	98.6	99	98	92
CJaya-KELM [29]	96.77	99	99.71	99.80
CJaya-MLP	91.76	92.46	93.62	93.65
CJaya-ELM	93.55	97.14	98.39	97.59
KFS-CJaya- CNN	**98.2**	**99.96**	**99.78**	99.87

Bold values are the best values obtained after evaluation.

7. Conclusion

In recent years, many dreadful diseases are threatening human beings due to the rapidly defiled environment. Therefore, a robust classification model is required to diagnose these diseases with high accuracy and less computational complexity. In this paper, due to the growing complexities of unstructured data, the researchers focus on the deep learning approach, which is the latest form of the machine learning algorithm. In this work, the KFS approach is considered to extract the highly effective genes, and an improvised chaotic Jaya (CJaya) algorithm optimized CJaya-CNN model is used to classify four high dimensional microarray data. The presented KFS-based CJaya-CNN model gave 98.2%, 99.96%, 99.78%, and 99.87% classification accuracy for colon cancer, leukemia, lymphoma-3, and SRBCT datasets, respectively. This model will reduce the human errors occured by inexperience or fatigue and assist to consider a decision before the biopsy in different cancer diseases. For future, we have planned to classify on very large scale high dimensional gene expression datasets like GSE13159 and GSE13204. To reduce the computational overhead during the classification of these datasets, parallel computing approach can be used. Eventually, different multi-objective optimization algorithms can be applied to perform the feature selection and classification job simultaneously with less time complexity.

References

[1] Shilaskar S, Ghatol A, Chatur P. Medical decision support system for extremely imbalanced datasets. Information Sciences. 2017; 384:205-19. doi: 10.1016/j.ins.2016.08.077

[2] Ang JC, Mirzal A, Haron H, Hamed HN. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. IEEE/ACM transactions on computational biology and bioinformatics. 2015; 13 (5):971-89. doi: 10.1109/TCBB.2015.2478454

[3] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M et al. Bloomfield CD. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science. 1999;286 (5439):531-7. doi: 10.1126/science.286.5439.531

[4] Panda M. Elephant search optimization combined with deep neural network for microarray data analysis. Journal of King Saud University-Computer and Information Sciences. 2020 ;32 (8):940-8. doi: 10.1016/j.jksuci.2017.12.002

[5] Kilicarslan S, Adem K, Celik M. Diagnosis and classification of cancer using hybrid model based on ReliefF and convolutional neural network. Medical hypotheses. 2020 ;137. doi: 10.1016/j.mehy.2020.109577

[6] Rao R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations. 2016;7 (1):19-34. doi: 10.5267/j.ijiec.2015.8.004

[7] Mohapatra P, Chakravarty S, Dash PK. Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system. Swarm and Evolutionary Computation. 2016; 28:144-60. doi: 10.1016/j.swevo.2016.02.002

[8] Aziz R, Verma C, Srivastava N. A fuzzy based feature selection from independent component subspace for machine learning classification of microarray data. Genomics data. 2016; 8:4-15. doi: 10.1016/j.gdata.2016.02.012

[9] Kar S, Sharma KD, Maitra M. Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique. Expert Systems with Applications. 2015 ;42 (1):612-27. doi:10.1016/j.eswa.2014.08.014

[10] Shukla AK, Singh P, Vardhan M. A two-stage gene selection method for biomarker discovery from microarray data for cancer classification. Chemometrics and Intelligent Laboratory Systems. 2018; 183:47-58. doi: 10.1016/j.chemolab.2018.10.009

[11] Alshamlan HM, Badr GH, Alohali YA. Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification. Computational biology and chemistry. 2015 ;56:49-60. doi: 10.1016/j.compbiolchem.2015.03.001

[12] García-Nieto J, Alba E. Parallel multi-swarm optimizer for gene selection in DNA microarrays. Applied Intelligence. 2012 ;37 (2):255-66. doi: 10.1007/s10489-011-0325-9

[13] Hernandez JC, Duval B, Hao JK. A genetic embedded approach for gene selection and classification of microarray data. InEuropean conference on evolutionary computation, machine learning and data mining in bioinformatics 2007:90-101. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-540-71783-6_9

[14] Wang A, An N, Chen G, Yang J, Li L et al. Incremental wrapper based gene selection with Markov blanket. In2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2014 : 74-79. IEEE. doi: 10.1109/BIBM.2014.6999251

[15] Alshamlan H, Badr G, Alohali Y. mRMR-ABC: a hybrid gene selection algorithm for cancer classification using microarray gene expression profiling. Biomed research international. 2015 ;2015. doi: 10.1155/2015/604910

[16] Liu KH, Zeng ZH, Ng VT. A hierarchical ensemble of ECOC for cancer classification based on multi-class microarray data. Information Sciences. 2016 ;349:102-18. doi: 10.1016/j.ins.2016.02.028

[17] Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A. Distributed feature selection: An application to microarray data classification. Applied soft computing. 2015 ;30:136-50. doi: 10.1016/j.asoc.2015.01.035

[18] Baliarsingh SK, Vipsita S, Gandomi AH, Panda A, Bakshi S et al. Analysis of high-dimensional genomic data using MapReduce based probabilistic neural network. Computer methods and programs in biomedicine. 2020 ; 195:105625. doi: 10.1016/j.cmpb.2020.105625

[19] Kumar M, Rath SK. Classification of microarray using MapReduce based proximal support vector machine classifier. Knowledge-Based Systems. 2015 ; 89:584-602. doi: 10.1016/j.knosys.2015.09.005

[20] Wang Y, Yang XG, Lu Y. Informative gene selection for microarray classification via adaptive elastic net with conditional mutual information. Applied Mathematical Modelling. 2019 ;71:286-97. doi: 10.1016/j.apm.2019.01.044

[21] Díaz-Uriarte R, De Andres SA. Gene selection and classification of microarray data using random forest. BMC bioinformatics. 2006 ;7 (1):1-3. doi: 10.1186/1471-2105-7-3

[22] Ludwig SA, Jakobovic D, Picek S. Analyzing gene expression data: Fuzzy decision tree algorithm applied to the classification of cancer data. In2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) 2015: 1-8. IEEE. doi: 10.1109/FUZZ-IEEE.2015.7337854

[23] Medjahed SA, Saadi TA, Benyettou A, Ouali M. Kernel-based learning and feature selection analysis for cancer diagnosis. Applied Soft Computing. 2017 ; 51:39-48. doi: 10.1016/j.asoc.2016.12.010

[24] Liu J, Wang X, Cheng Y, Zhang L. Tumor gene expression data classification via sample expansion-based deep learning. Oncotarget. 2017;8 (65):109646. doi: 10.18632/oncotarget.22762

[25] Liao Q, Jiang L, Wang X, Zhang C, Ding Y. Cancer classification with multi-task deep learning. In2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC) 2017: 76-81. IEEE. doi: 10.1109/SPAC.2017.8304254

[26] Zeebaree DQ, Haron H, Abdulazeez AM. Gene selection and classification of microarray data using convolutional neural network. In2018 International Conference on Advanced Science and Engineering (ICOASE) 2018 : 145-150. IEEE. doi: 10.1109/ICOASE.2018.8548836

[27] Polat K, Güneş S. A new feature selection method on classification of medical datasets: Kernel F-score feature selection. Expert Systems with Applications. 2009 ;36 (7):10367-73. doi: 10.1016/j.eswa.2009.01.041

[28] Bochinski E, Senst T, Sikora T. Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. In2017 IEEE international conference on image processing (ICIP) 2017: 3924-3928. IEEE. doi: 10.1109/ICIP.2017.8297018

[29] Debata PP, Mohapatra P. Selection of informative genes from high-dimensional cancerous data employing an improvised meta-heuristic algorithm. Evolutionary Intelligence. 2021:1-9. doi: 10.1007/s12065-021-00593

[30] LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015 ;521 (7553):436-44. doi: 10.1038/nature14539

[31] Deng L, Yu D. Deep learning: methods and applications. Foundations and trends in signal processing. 2014 ;7 (3–4):197-387. doi: 10.1561/2000000039

[32] Chen YW, Lin CJ. Combining SVMs with various feature selection strategies. InFeature extraction 2006: 315-324. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-540-35488-8_13

[33] Rao RV. Jaya: an advanced optimization algorithm and its engineering applications.(2019): 770-780. doi: 10.1007/978-3-319-78922-4

[34] Yu J, Kim CH, Wadood A, Khurshiad T, Rhee SB. A novel multi-population based chaotic JAYA algorithm with application in solving economic load dispatch problems. Energies. 2018 ;11 (8):1946. doi: 10.3390/en11081946

[35] Alon U, Barkai N, Notterman DA, Gish K, Ybarra S et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences. 1999 ;96 (12):6745-50. doi: 10.1073/pnas.96.12.6745

[36] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science. 1999 ;286 (5439):531-7. doi: 10.1126/science.286.5439.531

[37] Zhu Z, Ong YS, Dash M. Markov blanket-embedded genetic algorithm for gene selection. Pattern Recognition. 2007 ;40 (11):3236-48. doi: 10.1016/j.patcog.2007.02.007

[38] Baliarsingh SK, Vipsita S, Muhammad K, Dash B, Bakshi S. Analysis of high-dimensional genomic data employing a novel bio-inspired algorithm. Applied Soft Computing. 2019 ;77:520-32. doi: 10.1016/j.asoc.2019.01.007

[39] Basavegowda HS, Dagnew G. Deep learning approach for microarray cancer data classification. CAAI Trans. Intell. Technol.. 2020 ;5 (1):22-33. doi: 10.1049/trit.2019.0028

[40] Shah SH, Iqbal MJ, Ahmad I, Khan S, Rodrigues JJ. Optimized gene selection and classification of cancer from microarray gene expression data using deep learning. Neural Computing and Applications. 2020 Oct 6:1-2. doi: 10.1007/s00521-020-05367-8