APPLIED COMPUTER SCIENCE

Applied Computer Science 2025, vol. 21, no. 2, pp. 111–127
https://doi.org/10.35784/acs_6958

Submitted: 2024-12-17 | Revised: 2025-01-27 | Accepted: 2025-02-13

Keywords: genomic data, Kernal Fisher score, prominent gene selections, sine-cosine ensembled monarch butterfly optimization, two phase ensembled deep learning, classification accuracy

Prajna Paramita DEBATA [1*], *Pournamasi PARHI* [1], *Alakananda TRIPATHY* [1], *Smruti Rekha DAS* [2]

[1] S'O'A Deemed to be University, India, prajnaparamitadebata@soa.ac.in, pournamasiparhi@soa.ac.in, alakanandatripathy@soa.ac.in

[2] GITAM Deemed to be University, India, sdas5@gitam.edu

[*] Corresponding author: prajnaparamitadebata@soa.ac.in

A two phase ensembled deep learning approach of prominent gene extraction and disease risk prediction detection

Abstract

By gaining new insights into the gene expression of individual patient profiles, clinicians and researchers can identify patterns, biomarkers and therapies. In addition, accurate classification enables the development of predictive models for prognosis and treatment response, facilitating personalized medicine approaches. Determining the optimal model for classification remains a time-consuming, nondeterministic, polynomial-time hard problem. However, the available large amount of gene expression data is too much for the traditional data analysis approaches. Therefore, a two-phase ensemble deep learning approach can be considered as a reliable framework for the root-level investigation of genomic data. In this experimental model, a gene extraction approach, a Kernel-Applied Fisher Score (KFScore) method is presented to select the prominent genomes, and a Sine-Cosine Ensemble Monarch Butterfly algorithm (SC-MBO) optimized CNN (Convolutional Neural Network) strategy is implemented for genomic data classification. Here, the SC-MBO ensemble approach is used to obtain the optimal value of hyperparameters in CNN. The effectiveness of the presented model is estimated by accuracy% of classification, number of extracted prominent genomic features, sensitivity, specificity and ROC (Receiver Operating Characteristic) curve. The effectiveness of the proposed methods is successfully tested on GSE13159, GSE15061, GSE13204, breast cancer and ovarian cancer gene expression dataset with 91.6%, 90.22%, 91.9%, 97.93% and 99.6% accuracy. The proposed model is also compared with other existing models. According to the experimental evaluation, the proposed strategy is accurate, reliable and robust. Consequently, the presented method can be treated as a trustworthy basis for disease risk prediction.

1. INTRODUCTION

Despite advances in medical care, the diagnosis of cancer remains a formidable undertaking. A recent breakthrough in cancer research involves the use of microarray-based genomic expression profiling (Shilaskar et al., 2017). This innovative technology allows researchers to accurately identify which genes are involved in a tissue under different conditions. Nevertheless, unlocking new insights from gene expression or microarray data poses significant challenges for researchers, including the hurdles of high dimensionality, redundant gene concerns, missing or imbalanced data, retrieval of biological information, and susceptibility to bias from various factors (Ang et al., 2015).

However, a major obstacle in microarray analysis is the high dimensionality of the data (Aziz et al., 2016). The computational complexity of handling such high-dimensional datasets is increased, underscoring the need to reduce the size of the genomic attribute for effective analysis of cancer genomic data. These challenges have led to a surge of interest in various data mining approaches for the analysis of cancer genomic data (Golub et al., 2015). Classification of unstructured, high-dimensional genomic data for disease diagnosis is a daunting task due to its high dimensionality, resulting in significant computational complexity. Researchers have proposed numerous feature extraction/selection techniques coupled with classification models to address this problem on various benchmark datasets. Nevertheless, determining the optimal model for classification remains a time-consuming, nondeterministic, polynomial-time hard problem. Therefore, there is a constant opportunity for the implementation of novel algorithms in this area.

In this study, an ensemble approach of deep learning is proposed for cancer data classification. In this regard, deep learning algorithms have gained significant interest for disease classification using gene expression data (Panda, 2017). Among the various deep learning methods, the CNN model has demonstrated its effectiveness (Kilicarslan et al., 2020), especially in dealing with unstructured high-dimensional data.

In this study, a feature selection algorithm known as Kernel Fisher Score-based (KFScore) (Polat & Güneş, 2009) is used to select informative genes. In addition, an ensemble approach combining the sine-cosine method with the Monarch Butterfly algorithm (Wang et al., 2019a), referred to as SC-MBO, is applied to optimize the hyperparameters of the CNN and effectively classify high-dimensional cancer data. The key aspects of this research are outlined below:

- First, the KFScore algorithm is used to identify significant genes.
- Then, the SC-MBO algorithm is used to determine the optimal hyperparameter values for the CNN.
- The SC-MBO ensemble CNN classifier is then implemented to classify cancer data.
- To address the author's concern, this study introduces the KFScore-based SC-MBO CNN model, a novel approach to high-dimensional cancer data classification. In particular, the parameters of the CNN are optimized using SC-MBO.
- A comparative analysis is performed between the proposed ensemble deep learning approach (SC-MBO-CNN) and other basic machine learning classifiers.

The remaining papers are organized as follows: While Section 3 provides an overview of the proposed paradigm, Section 2 explores related literature. In Section 4, all backbone approaches (supporting methods) and the proposed algorithm are explained. The experimental design criteria are explained in Section 5, and the results are examined in Section 6. Finally, concluding thoughts and implications for the future are given in Section 7.

2. SURVEY ON EXISTING WORK

To efficiently categorize high-dimensional malignant data, a large number of researchers have presented robust classification models and a variety of feature extraction strategies. To select the most salient features, several ensemble techniques combine machine learning techniques with a variety of metaheuristic algorithms (Mohapatra et al., 2016; Alshamlan et al., 2015). These ensemble techniques clarify how genes interact with each other and improve the effectiveness of gene extraction strategies.

Various ensemble classification techniques have been used to efficiently select key features and classify genomic data. Baliarsingh et al. (2020) used a MapReduce (MR) ensembled Fisher score for gene extraction strategy and an MR-based probabilistic neural network (PNN) for classification of genomic data. Kumar and Rath (2015) applied a MapReduce feature selection technique with MapReduce SVM for cancer classification. Wang et al. (2019b) applied the Adaptive Elastic Net with Conditional Mutual Information (AEN-CMI) technique to classify leukemia and colon data. Mohapatra et al. (2016) employed a cat swarm hybridized Kernel Ridge Regression classifier to efficiently classify genomic data. Diaz and Ludwig (2006) and Ludwig et al. (2015) used random forest with a fuzzy decision tree algorithm to classify medical data. Medjahed et al. (2017) proposed a Binary Dragonfly (BDF) ensembled Support Vector Machine ensembled Recursive Feature Elimination (SVM-RFE) strategy for classification of genomic data. An ensembled model of Stacked Autoencoder with CNN was applied for classification of gene expression data (Liu et al., 2017). Kilicarslan et al. (2020) proposed an ensembled model of ReliefF and CNN for biomedical data classification in this domain. Liao (2017) implemented a multi-task deep learning (MTDL) strategy for data classification. Zeebaree et al. (2018) presented a CNN classifier for medical data classification. Polat and Güneş (2009) introduced a kernel function applied FS strategy for biomedical data classification. A new evolutionary algorithm optimized CNN is also presented by Erik et al. (Bochinski et al., 2017) for data classification. In this domain, Debata and Mohapatra (2022) classify high-dimensional tumor data and select the most informative genes by combining CNN and chaotic Jaya algorithm. This work focuses on reducing computational time and improving performance through a hybridized deep learning model.

All the models in the previous literature review used deep learning techniques or traditional machine learning algorithms to classify genomics data. In contrast, this study compares deep learning methodology with traditional machine learning techniques to classify genomics data. In addition, we have proposed an ensemble strategy that combines a KFScore-based filter for feature extraction with an SC-MBO-optimized CNN, or SC-MBO-CNN model, for classification of malignant data. The main motive of this experimental work is to assist physicians in efficiently diagnosing a malignant and non-malignant cancer cell within a reasonable time frame with high accuracy. The following section provides a detailed explanation of the proposed model.

3. PROPOSED TWO-PHASE ENSEMBLED APPROACH

In this study, we present an ensemble two-phase method for the selection of significant genomic features and classification of cancer data. Figure 1 shows the general strategy of the KFScore ensemble SC-MBO-CNN approach. Using min-max normalization, all datasets are normalized and missing cells are imputed with the value that occurs most frequently for that particular feature (Mohapatra et al., 2016). After normalization, the dataset is divided into a training set and a test set. Then, a filtering method called KFScore is used to select the most relevant genetic attribute. For classification, the KFScore filtered genes are given to the optimized SC-MBO-CNN. At the same time, the hyper-parameters, i.e., dropout rate, learning rate, batch size, and number of layers of the CNN are optimized by SC-MBO. Finally, the test set with the best feature subset is used to estimate the KFScore-SC-MBO-CNN method, and the accuracy is used to evaluate the results. Furthermore, the presented method is also compared with other common machine learning classification models.

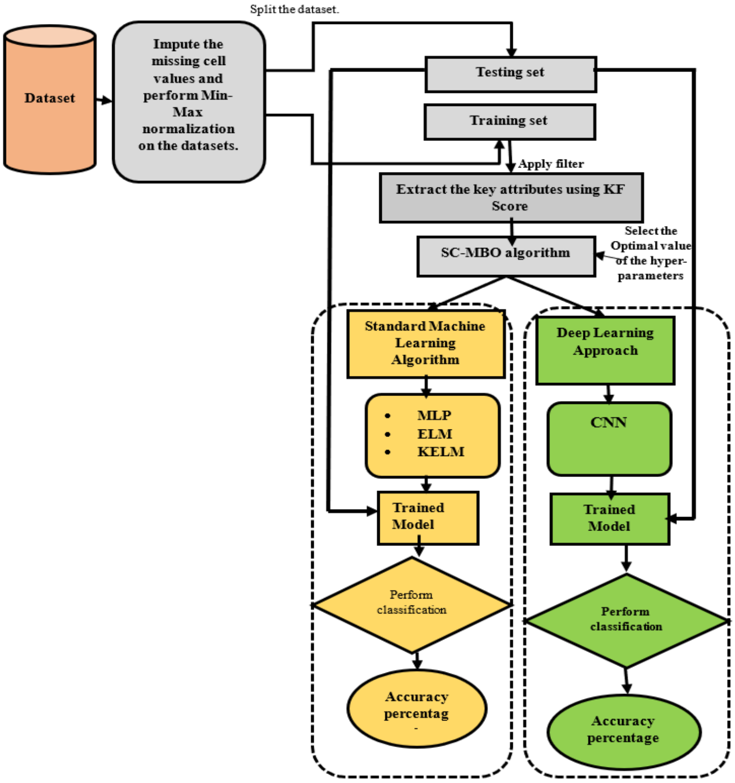


Fig. 1. The overall description of the presented KFScore-SC-MBO-CNN strategy

4. BACKBONE METHODOLOGIES AND PROPOSED ALGORITHM

4.1. CNN model architecture

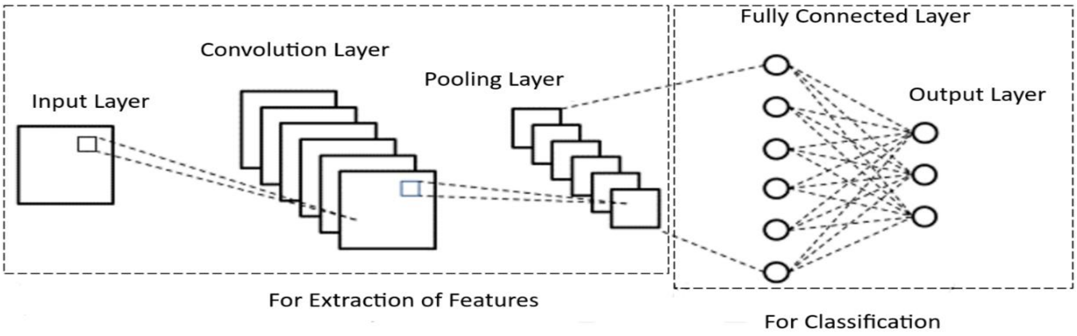


Fig. 2. A comprehensive structure of the CNN model

CNNs (LeCun et al., 2015) involve a convolution operation in the input layer with interconnected connections, unlike standard artificial neural networks (ANNs), where input layer neurons are directly connected to output layer neurons in the following layer. Selected filters are used by each layer of the CNN and the outputs are connected. In the training phase, features are trained in each layer. The general layout of the current CNN is shown in figure 2, where the raw input data is preprocessed before entering the convolution layer. The working principle of the main layers of the CNN is explained below:

Step-1: Input Layer

Receive input of five extensive cancer data, such as GSE13159, GSE15061, GSE13204 breast cancer, ovarian cancer (19) Data in the form of matrices.

Step-2: Convolutional Layer

This layer selects features from the input data based on specified dimensions using filters. At this stage, weights are randomly generated. In this study, these weights are used to apply a 3x3 filter to the one-dimensional data, creating a new feature map. This convolution process is repeated on the entire data set. The resulting data set is subjected to the Rectified Linear Unit (RELU) activation function. Then, a normalization technique is implemented to preserve the dispersion of the data, which may undergo changes during the convolution process.

Step-3: Pooling Layer

Sample feature maps obtained from the convolutional layers to reduce spatial dimensions. This helps in retaining the most relevant information while reducing the computational complexity. In this study, the Max Pooling technique is used with a pool size and stride value of 2 to extract the most important features. During the training phase, the neuron dropout method with a dropout rate of 0.2 is used to mitigate overfitting. For both the categorization within the fully connected layer and the connectivity with the neurons in the layer above, the neuron density is set to 1024.

Step-4: Fully Connected Layer

Process the flattened feature vectors through one or more fully connected layers. These layers learn to classify the extracted features and establish connections between different parts of the input data.

Step-5: Output Layer

Unlike Recurrent Neural Networks (RNN), CNN is preferred in deep learning for classification due to its ability to handle memory more efficiently, especially when dealing with high-dimensional datasets, resulting in higher classification accuracy rates. In addition, the use of the rectified linear unit (ReLU) activation function in CNN addresses the vanishing gradient problem encountered in RNN (LeCun et al., 2015). Finally, this probability-driven layer is used to improve the accuracy. A softmax function is used in this layer to normalize the output values by converting them to probability values. These probability values are then used to classify the test data.

4.2. Selection of genes using KFScore

In the basic Fisher Score (FScore) method, the score (or value) of a gene is determined using equation (1). Then, by calculating the mean value of all FScore values of a gene, a threshold value (THV) is derived. Genes with FScore values above the THV are included in the feature set, while those with FScore values below the THV are excluded from the feature set.

$$\boldsymbol{FScore(gene_i)} = \frac{\left(\bar{z}_i^{(+)} - \bar{z}_i\right)^2 + \left(\bar{z}_i^{(-)} - \bar{z}_i\right)^2}{\frac{1}{(m_+ - 1)} \sum_{k=1}^{m_+} \left(z_{x,i}^{(+)} - \bar{z}_i^{(+)}\right)^2 - \frac{1}{(m_- - 1)} \sum_{x=1}^{m_-} \left(z_{x,i}^{(-)} - \bar{z}_i^{(-)}\right)^2} \tag{1}$$

z_x is the training vector, z_+ and z_-represent the no. of positive and negative instances in Eq. (1). Additionally, $\bar{z}_i$ signifies the ith gene among all the datasets, $\bar{z}_i(+)$ represents the ith attribute of the positive datasets, $\bar{z}_i$(-) denotes the ith attribute among the negative datasets.

Likewise, $z_{k,i}^{(+)}$ corresponds to the ith gene or attribute of the kth positive instances, and $\bar{z}_i^{(-)}$ stands for the ith gene among the kth negative instances. In particular, the basic FScore approach does not take into account the mutual information between genes, which is a significant drawback (Chen & Lin, 2006). To address this limitation, the KFscore (Polat, & Güneş, 2009) is not only nonlinearly differentiable data sets into linearly differentiable ones, but also reduces the computational complexity. The steps involved in KFScore are summarized in Algorithm 1.

Algorithm 1:	Feature selection by KFscore
Input:	Normalized high-dimensional data set
Output:	Reduced cancer dataset with key attribute subset
1.	Start.
2.	By using a kernel function, i.e, linear functions or RBF (Radial Basis Function), the input feature spaces of the data are converted to kernel space.
3.	After the transformation, Fisher Score (FS) values for datasets with high-dimensional attribute spaces are computed using Eq. (1).
4.	The mean of all Fisher Score values is then determined and this result is considered the Threshold Value (THV).
5.	Finally, features with FScore values above the THV are included in the feature space, while genes with FScore values below the THV are excluded from the feature space.
6.	End.

4.3. MBO algorithm

The MBO is a recently developed nature-inspired optimization algorithm inspired by the migration rules of monarch butterflies, particularly those available in North America (Wang et al., 2019a). This algorithm is characterized by its simplicity and ease of implementation, relying on two primary variables: The Migration Operator (MO) and the Butterfly Adjustment Operator (BAO). Figure 3. provides a visual representation of the MBO algorithm.

Fig. 3. Visual depiction of the MBO

4.3.1. Operator-I (MO)

The primary aim of the MO is to facilitate information interchange among two populations within subpopulation-1. Every butterfly in subpopulation 1 updates in accordance with another butterfly's location and is influenced by the migration ratio, denoted as "g" in both populations. The updating equation for the 1st butterfly in subpopulation-1 can be formulated as follows:

$$S_{k,n}^{m+1} = \begin{cases} S_{k1,n}^{m} & if\ p<g \\ S_{k2,n}^{m} & else \end{cases} \tag{2}$$

$S_{l,n}^{m+1}$ is the location of S_l on the 1th dimension in the *(m+1)th* generation, and $k_1$and k_2 are integer indices randomly selected from two subpopulations. The parameter *p* is defined as *p=(Rand*Peri)*, where *Rand* is a randomly chosen real number in the range (0-1) and *Peri* is the migration time.

4.3.2. Operator-II (BAO)

The displacement of each butterfly-l within subpopulation-2 is determined by considering the adjustment ratio p and the butterfly's adjustment rate (ARB).

$$S_{l,n}^{t+1} = \begin{cases} S_{best,n}^{m} & if\ Rand \le g \\ S_{l3,n}^{m} & if\ \ Rand > g \wedge Rand\ \le ARB \\ S_{l,n}^{m} + \xi \times (dS_n - 0.5) & ifRand > g \wedge Rand > ARB \end{cases} \tag{3}$$

$S_{best,n}^{m}$ represents the nth element of the global best, and ($S_{l3,n}^{m}$)is the nth element of a randomly chosen butterfly at generation *m.*

The butterfly chosen from subpopulation-2, denoted as ξ, is expressed as a weighted factor, and can be defined as:

$$\xi = X_{max}/m^2 \tag{4}$$

In this context, X_{max} denotes the highest walking step of each butterfly in all subsequent steps, *m* denotes the current generation, and dS_ndenotes the steps taken by each butterfly i, based on the strategy of the Levy flight approach, as follows:

$$dS_n = Levy(S_n^m) \tag{5}$$

In this context, ξ significantly influences both $dS_n$and $S_{l,n}^{t}$. A higher ξ value accelerates exploration in the search space, while a smaller ξ value facilitates exploitation in the search space.

4.5. SC ensembled MBO algorithm

This section outlines the rationale behind the fusion of the Sine Cosine (Sharma et al., 2022) and MBO algorithms. While SC is known for its exceptional exploration capabilities, there are instances where it struggles to strike a balance between the exploitation and exploration phases, resulting in suboptimal results. In some cases, the SC algorithm may overlook the global best solution, resulting in limited exploitation. This reduces the overall search efficiency of the algorithm. On the other hand, the MBO algorithm is efficient at maintaining a balance between exploration and exploitation during the search process. However, like other evolutionary algorithms, MBO is susceptible to getting stuck at local optima.

To address these issues, a hybrid algorithm, namely SC-MBO, is proposed here. The hybridization of SC and MBO aims to utilize the exceptional exploration capability of the SC algorithm, while improving the exploitation capabilities and avoiding the local optima trap characteristic of MBO. The steps for SC ensemble MBO are detailed in the following two algorithms (Algorithm 2 and 3).

Algorithm- 2: MO-I

```
Start
  for $m = 1$ to P1(population of Region1)
    for $k = 1$ to i (elements of $m^{th}$ MB)
         q= rand * L, here, random ~ R(0,1)
  if q ≤ s then
   $C_{m,k}^{t+1} = C_{q_1,k}^{t+1}$ Here, $q_1$ ~ R (1,2,3, …, X1)                                  (6)
            elseif random () < 0.5
               $C_{m,k}^{t+1} = C_{m,k}^{t} + c_1 \times \sin(c_2) \times c_3 |D_{m,k}^{t} - C_{m,k}^{t}|$     (7)
           else
                $C_{m,k}^{t+1} = C_{m,k}^{t} + c_1 \times cos\ (c_2) \times c_3 |D_{m,k}^{t} - B_{m,k}^{t}|$     (8)
         end elseif
   end if
   if q>s then
              $C_{m,k}^{t+1} = C_{q_2,k}^{t+1}\ where$ $q_2$ ~ R                                    (9)
        elseif random () < 0.5
          $C_{m,k}^{t+1} = C_{m,k}^{t} + c_1 \times \sin(c_2) \times c_3 |D_{m,k}^{t} - Z_{m,k}^{t}|$        (10)
        else
        $C_{m,k}^{t+1} = C_{m,k}^{t} + c_1 \times \cos(c_2) \times c_3 |D_{m,k}^{t} - z_{m,k}^{t}|$          (11)
       end elseif
      end if
     end for k
  end for m
End
```

Algorithm-3: BAO-II

```
Start
for n = 1 to P2(population of region 2)
  for k = 1 to i (elements of $n^{th}$ MB)
    if q ≤ s then,
       $C_{n,z}^{t+1} = C_{best,k}^{t}$ where $random$ ~ $R\ (0,1)$                                (12)
      elseif random () < 0.5
              $C_{n,k}^{t+1} = C_{n,k}^{t} + c_1 \times \sin(c_2) \times c_3 |D_{n,k}^{t} - C_{n,k}^{t}|$      (13)
          else
               $C_{n,k}^{t+1} = C_{n,k}^{t} + c_1 \times cos\ (c_2) \times c_3 |D_{n,k}^{t} - C_{n,k}^{t}|$      (14)
       end elseif
      end if
      if q>s then
        $C_{n,k}^{t+1} = C_{r_3,k}^{t+1}$ $where$, $q_3$ ~ (1,2, 3,…, X2)                           (15)
        elseif rand () < 0.5
        $C_{n,k}^{t+1} = C_{n,k}^{t} + c_1 \times \sin(c_2) \times c_3 |D_{n,k}^{t} - z_{n,k}^{t}|$          (16)
       else
        $C_{n,k}^{t+1} = C_{n,k}^{t} + c_1 \times \cos(c_2) \times c_3 |D_{n,z}^{t} - z_{n,k}^{t}|$          (17)
       end elseif
       end if
    if q> random ()
      $C_{n,k}^{t+1} = C_{n,k}^{t} + \beta \times ((dc_k) - 0.5))$                                  (18)
       end
     end for k
  end for n
End
```

In Algorithm-2, $C_{m,k}^{t+1}$ represents the k^{th} element of C_m at the (t + 1)th generation, signifying the m location of the MB in Region1. Similar, $C_{q_1,k}^{t}$ is the k^{th} element of C_{q_1} in the t^{th} generation, indicating the current position of the q_1 Monarch butterfly after modification. Finally, $C_{q_2,k}^{t}$ is the k^{th} element of C_q in the t^{th} generation, which represents the updated position of the q_2 MB at that time.

Here, $C_{n,k}^{t+1}$ illustrates the k^{th} element of C n in *(t + 1)th* generation, denotes the n location of the Monarch butterfly (MB) in Region2. Similarly, $C_{r_3,k}^{x}$ is the kth element of A_{q_3} in the t^{th} generation, which represents the currently updated position of the q_3 Monarch butterfly. The variable dc_k symbolizes the walking step of the nth Monarch butterfly, determined by applying the Levy flight mechanism. In this context, t is taken as the current generation, β is defined as $\beta = {}^{IB_{max}}/_{t}$ where IB_{max} = represents the maximum walking distance of each butterfly in a single step.

4.6. Proposed SC-MBO optimized CNN (SC-MBO-CNN) algorithm

In this study, the hyper-parameters such as learning rate, batch size, number of layers and dropout rate of Convolutional Neural Network (CNN) are optimized using SC-MBO.

The learning rate (LR) indicates the amount of learning in each iteration. If the value is too small, the learning process may terminate prematurely before training is complete. Conversely, if the value is too large, the learning process may become disconnected and stop learning correctly.

Dropout is the random removal of neural connections. Without dropout, the network tends to overfit the training data, resulting in reduced accuracy on test data. To mitigate overfitting and improve accuracy, the dropout process is recommended. Notably, the dropout effect is less pronounced in convolution layers compared to fully-connected layers due to the relatively fewer parameters in convolution layers (Yoo et al., 2019). Here, dropout rates are assigned to convolutional layers (referred to as DR1) and fully connected layers (referred to as DR2).

Here, the number of layers (NL) needs to be optimized because it plays a crucial role in extracting features from the dataset. The greater the depth of the layer, the more effective the extraction of smaller features. Therefore, it is important to determine the optimal number of layers for the given dataset.

Tab. 1. Dynamic ranges of hyper parameters

Hyper parameter	Range
Learning Rate (LR)	0.0001-0.1
Dropout Rate (DR1)	0-0.5
Dropout Rate (DR2)	0-0.5
Batch Size (BS)	50, 100, 150, 200, 250, 500, 1000
No. of Layer (NL)	1-4

The parameter values are limited to dynamic ranges as shown in Table 1. If the parameter values exceed these ranges, they are adjusted to fall within the specified dynamic range. Figure 4 and Algorithm 4 provide a visual representation and a step-by-step flow of the proposed model, respectively.

Name of the algorithm4:	**SC-MBO-CNN**
Input:	Population size (PS), No. of iterations (NI), Upper and lower bound (LB &UB) for LR, DR1, DR2, BS, NL
Output:	CAP (Classification accuracy percentage)

```
Step(S)
S-1:    Begin
S-2:    Initialize PS, NI, LR, DR1, DR2, BS, NL.
S-3:    For each resulting solution, the fitness(Fit) value is obtained. (Fitness is CAP when CNN is applied to
        previously selected significant features by KFScore with the initialized value of LR, DR1, DR2, BS and NL)
S-4:    Save the fitvalues in descending order. Highlight the best and worst.
S-5:    Sort the population(PS)based on the index position of the ordered fitness values.
S-6:    Assign the fitness value and corresponding location of the best solution as the fitness value and location.
S-7:    Compute the mean_Fit.
S-8:    while $Itr < Max\_ Itr$ do
S-9:        if $Itr = = 1$ then
S-10:           for j=1: PS do
S-11:           Update the solution's position with LR, DR1, DR2, BS and NL by using Eqs. (9-21).
S-12:           end for
S-13:       elseif (current_mean_Fit – previous_mean_Fit)/current_mean_Fit > 0.001
S-14:       then
S-15:           Repeat S-10 to S-12
S-16:           else
S-17:           break.
S-18:       end elseif
S-19:       end if
S-20:       for every upgraded candidate solution do
S-21:           Check the LB and UB values for the candidate's solution position., LR, DR1, DR2, BS and NL
S-22            Repeat S-2 to S-3 for recalculate the new_Fit values.
S-23:         if current_ Fit > previous_Fit, then
S-24:             Change the Fit value of the solution.
S-25:             Upgrade the solution location, LR, DR1, DR2, BS and NL.
S-26:           else
S-27:             Store the Fit result of prior one.
S-28:             Point out the solution location, LR, DR1, DR2, BS, and NL of the prior candidate solution.
S-29:           end if
S-30:           Repeat S-4 to S-7
S-31:       end for
S-32:   end while
S-33:   Achieve the final Fit result (i.e., CAP).
S-34:   End
```

4.7. Complexity analysis of the suggested algorithm

The computational complexity of the presented methods, namely KFScore, SC-MBO and CNN, is detailed in Table 2. In this table, "Itr" denotes the iterations, "TI" denotes the training instances, and "NF" denotes the number of attributes in the computation of the complexity of KFScore. The time complexity of adjusting the solution positions in SC-MBO depends on both the population size and the dataset dimension. In the computational complexity expression of SC-MBO, "PS" corresponds to the population size and "Dim" corresponds to the dataset dimension. For the computational complexity of CNN, the following variables are used: "CV" denotes the number of layers in the convolutional layer, "l" denotes the layer, "Ml-1" denotes the number of inputs in layer l, "Fl" denotes the number of filters, "Sl" denotes the size of the filter, and "Al" denotes the size of the feature map.

Tab. 2. Complexity of the presented methods

Applied Methods	Complexity
KFS	$O\ (Itr * TI * NF)$
SC-MBO	$O\ (PS * Dim)$
CNN	$O\left(\sum_{l=1}^{CV} M_{l-1} * F_l * S_l * A_l\right)$

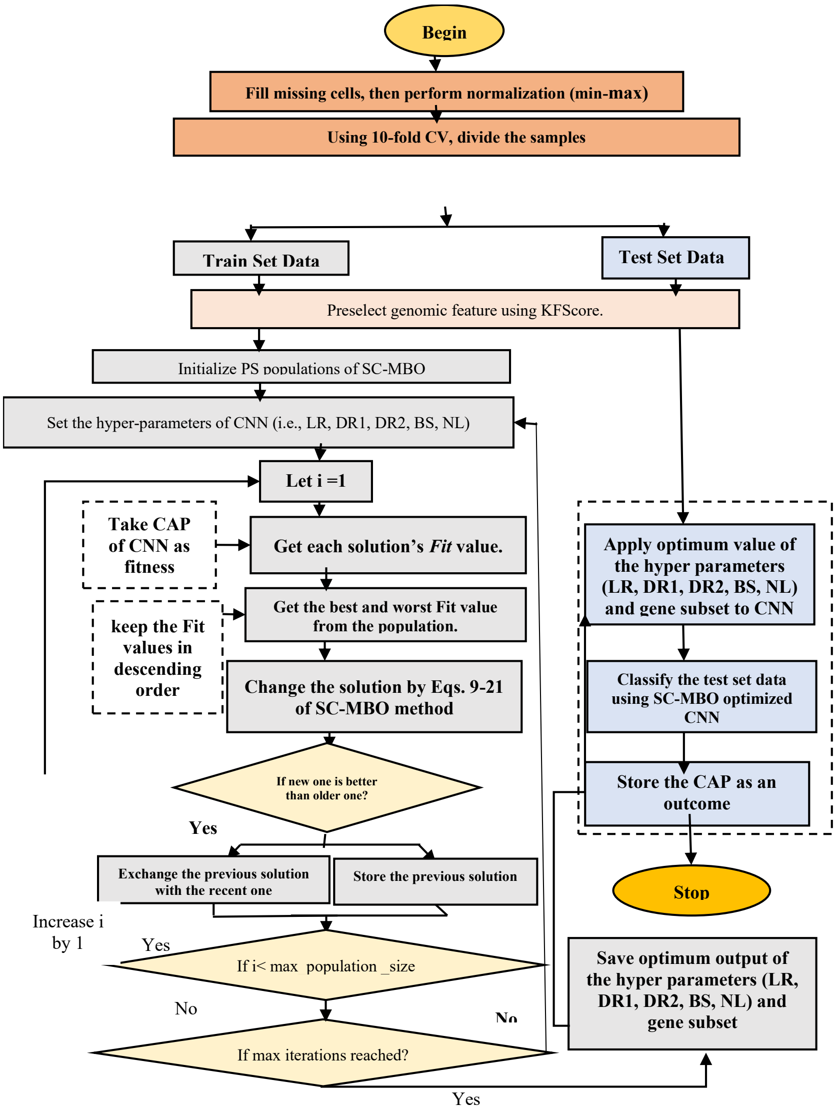
5. EXPERIMENTAL MEASURES

5.1. System setup

The experiments are conducted in a setup consisting of Google Co Lab and Python 2.7 (64 bits) as the programming language.

5.2. Datasets description

The effectiveness of the proposed approach is verified using five very high-dimensional cancer datasets obtained from NCBI GEO. Detailed descriptions of the datasets are given in Table 3.



Begin
Fill missing cells, then perform normalization (min-max)
Using 10-fold CV, divide the samples
Train Set Data
Test Set Data
Preselect genomic feature using KFScore.
Initialize PS populations of SC-MBO
Set the hyper-parameters of CNN (i.e., LR, DR1, DR2, BS, NL)
Let i =1
Take CAP of CNN as fitness
Get each solution's Fit value.
keep the Fit values in descending order
Get the best and worst Fit value from the population.
Change the solution by Eqs. 9-21 of SC-MBO method
If new one is better than older one?
Yes
Exchange the previous solution with the recent one
Store the previous solution
Increase i by 1
Yes
If i< max population _size
No
No
If max iterations reached?
Yes
Apply optimum value of the hyper parameters (LR, DR1, DR2, BS, NL) and gene subset to CNN
Classify the test set data using SC-MBO optimized CNN
Store the CAP as an outcome
Stop
Save optimum output of the hyper parameters (LR, DR1, DR2, BS, NL) and gene subset

Fig. 4. Visual representation of the proposed scheme

Tab. 3. The detailed elaboration of the five high dimensional cancerous datasets

Cancerous data	# Sample	# Feature	# Classes	Memory size
GSE13159 [19]	2096	54674	18	1.93 GB
GSE15061 [19]	870	54675	3	650 MB
GSE13204 [19]	3248	1480	18	1.96 GB
Breast Cancer [19]	97	24481	2	20.2 MB
Ovarian cancer [19]	253	15154	2	32.6 MB

5.3. Initialization value of the parameters

A comparative analysis was conducted between the SC-MBO-CNN model and conventional machine learning approaches, including SC-MBO-ELM, SC-MBO-MLP, and SC-MBO-KELM. Table 4 provides a comprehensive overview of the default values of parameters of all above algorithms.

Tab. 4. A comprehensive overview of the initialization values of the parameters

MLP	ELM	KELM	CNN	SC-MBO
# iterations =100 # Hidden layers (HL) =3 # Nodes in each HL =5	# iterations=100 # nodes HL =15	C, γ = $[2^{-7}, 2^{-8}, \ldots, 2^{7}, 2^{8}]$	Activation function = ReLU Pad size (P) = 2 # Kernel (KS) = 4 Pooling algorithm (Po) = Max Output function = Sofmax Training Epochs = 100	# Population =100 # Iterations =100 Constant (l) = 3.0 c_1 = 20 c_2 = rand(0,2π) c_3 = 20 c_4 = rand (0,1) Default migration period (MP) =1.2 Default migration ratio = 5/12 Adjusting value = 5/12 Size of max step walk =1

5.4. Performance evaluating measures

Here, in 10-fold CV, each fold reserves thirty percent of the samples from the total data for testing, while the remaining seventy percent is used for training. To mitigate the inherent randomness in the meta-heuristic algorithm, this study performs 10 independent runs and the result is defined as the average of these runs. In addition, the performance evaluation includes measures such as accuracy percentage, sensitivity (Mohapatra et al., 2016), specificity (Mohapatra et al., 2016), and receiver operating characteristic (ROC) (Debata & Mohapatra, 2022).

6. DISCUSSION OF EXPERIMENTAL OUTCOME

The enormous amount of unrelated genes in the dataset contributes to the curse of dimensionality problem. Basic Fisher Score approaches and Kernel based Fisher Score techniques are applied to solve this problem and select the most interdependent pertinent features from the input data set. Both techniques assign a score to each attribute. Then, the attributes are ranked in descending order based on their scores and the features with the highest rank are selected. Table 5 shows a comparison of the proposed ensemble classification approaches in terms of accuracy with the number of selected features, specificity and sensitivity. According to table 5, from the GSE13159 dataset, the proposed KFScore-based ensemble classification method extracts 500 genes with 91.6% accuracy while FS-based ensemble classification method extracts 1250 genes with 90.8% accuracy. From GSE15061, the KFS-based ensemble classification method extracts 150 genes with 90.22% accuracy, while the FS-based ensemble classification method extracts 200 genes with 88% accuracy. From the GSE13204 dataset, the KFS-based ensemble classification method extracts 300 genes with 91.9% accuracy, while the FS-based ensemble classification method extracts 500 genes with 89.8% accuracy. From breast cancer dataset, KFS-based ensemble classification method extracts 20 genes with 97.33% accuracy while FS-based ensemble classification method extracts 150 genes with 96.3% accuracy. From ovarian cancer dataset, KFS-based ensemble classification method extracts 100 genes with 99.6% accuracy, while FS-based ensemble classification method extracts 250 genes with 97.8% accuracy.

Tab. 5. A comparison of two ensemble classification approaches accuracy with number of selected features

Dataset	Ensembled Classification Approach	# Selected Features	Accuracy%	Sensitivity%	Specificity%
GSE13159	KFScore-SC-MBO-CNN	500	91.6	90.85	89.62
	FS-SC-MBO-CNN	1250	90.8	88.62	87.53
GSE15061	KFScore-SC-MBO-CNN	150	90.22	90.18	90.2
	FS- SC-MBO-CNN	200	88	85.56	87.26
GSE13204	KFScore-SC-MBO-CNN	300	91.9	91.82	90.64
	FS- SC-MBO-CNN	500	89.8	88.32	89.2
Breast Cancer	KFScore-SC-MBO-CNN	20	97.93	96.65	97.72
	FS- SC-MBO-CNN	150	96.3	96.2	95.87
Ovarian cancer	KFScore-SC-MBO-CNN	100	99.6	99.12	97.53
	FS- SC-MBO-CNN	200	97.8	95.52	97.6

Table 6 shows a comparison of the results (accuracy, specificity and sensitivity) between the standard machine learning model and the proposed ensemble deep learning method on five high-dimensional cancer datasets. According to table 6, in GSE13159 dataset, the proposed ensemble classification method i.e., SC-MBO-CNN yields 91.6% accuracy whereas SC ensemble MBO-KELM, SineCosine ensemble MBO-ELM and SineCosine ensemble MBO-MLP yield 89.6%, 89% and 76.34% respectively. Similarly, in the GSE15061 dataset, the proposed SC-MBO-CNN yields 90.22% accuracy, while the SC-Ensembled MBO optimized KELM, SC ensembled MBO optimized ELM, and SC ensembled MBO optimized MLP yield 86.9%, 83.6%, and 77.4%, respectively. In GSE13204 dataset, the proposed SineCosine ensemble MBO optimized CNN yields 91.9% accuracy while SC ensemble MBO optimized KELM, SC ensemble MBO optimized ELM, and SC ensemble MBO optimized MLP yield 87%, 86.2%, and 73.34% respectively. In the breast cancer dataset, the proposed SineCosine ensemble MBO optimized CNN yields 97.93% accuracy, while SineCosine ensemble MBO optimized KELM, SineCosine ensemble MBO optimized ELM, and SineCosine ensembled MBO optimized MLP yield 95.8%, 93.35%, and 91.76%, respectively. In the ovarian cancer dataset, the proposed SC-MBO-CNN yields 99.6% accuracy, while SineCosine ensembled MBO-KELM, SineCosine ensembled MBO-ELM, and SineCosine ensembled MBO-MLP yield 95%, 92.7%, and 89%, respectively.

Tab. 6. Classification accuracy between the standard machine learning model and suggested method in five high-dimensional cancerous datasets

Dataset	Methods used	Accuracy%	Sensitivity%	Specificity%
GSE13159	SC-MBO-CNN	91.6	90.85	89.62
	SC_MBO-KELM	89.6	88.2	87.65
	SC_MBO-ELM	89	87.83	88.75
	SC-MBO-MLP	76.34	73.67	72.36
GSE15061	SC-MBO-CNN	90.22	90.18	90.2
	SC_MBO-KELM	86.9	86.92	85.5
	SC_MBO-ELM	83.6	81.32	82.75
	SC-MBO-MLP	77.4	74.71	76.32
GSE13204	SC-MBO-CNN	91.9	91.82	90.64
	SC_MBO-KELM	87	86.32	87.15
	SC_MBO-ELM	86.2	85.37	86.06
	SC-MBO-MLP	73.34	72.79	73.21
Breast Cancer	SC-MBO-CNN	97.93	96.65	97.72
	SC_MBO-KELM	95.8	93.2	94.65
	SC_MBO-ELM	93.35	91.35	92.95
	SC-MBO-MLP	91.76	90.69	91.46
Ovarian cancer	SC-MBO-CNN	99.6	99.12	97.53
	SC_MBO-KELM	95	94.2	95.05
	SC_MBO-ELM	92.7	90.83	91.58
	SC-MBO-MLP	89	88.42	89.36

A GPU time (in seconds) comparison of KFS-based Optimized CNN and FS-based Basic CNN on five high-dimensional cancer datasets. From table 8, it can be seen that the KFS-based SC-MBO CNN model takes more time overall than the FS-based SC-MBO CNN model. Since the specificity, accuracy and sensitivity of the KFS-based SC-MBO-CNN approach are higher than other standard machine learning models, it can be considered as the best approach for classifying high-dimensional cancer data.

Tab. 7. A GPU time (in second) comparison of KFS based optimized CNN and FS based basic CNN in five high-dimensional cancerous datasets

Dataset	Methods used	GPU Time (in Sec)
GSE13159	KFS-SC-MBO-CNN	2.32
	FS-SC-MBO-CNN	2.22
GSE15061	KFS-SC-MBO-CNN	2.07
	FS-SC-MBO-CNN	1.98
GSE13204	KFS-SC-MBO-CNN	2.68
	FS-SC-MBO-CNN	2.62
Breast cancer	KFS-SC-MBO-CNN	1.62
	FS-SC-MBO-CNN	1.58
Ovarian Cancer	KFS-SC-MBO-CNN	1.82
	FS-SC-MBO-CNN	1.75

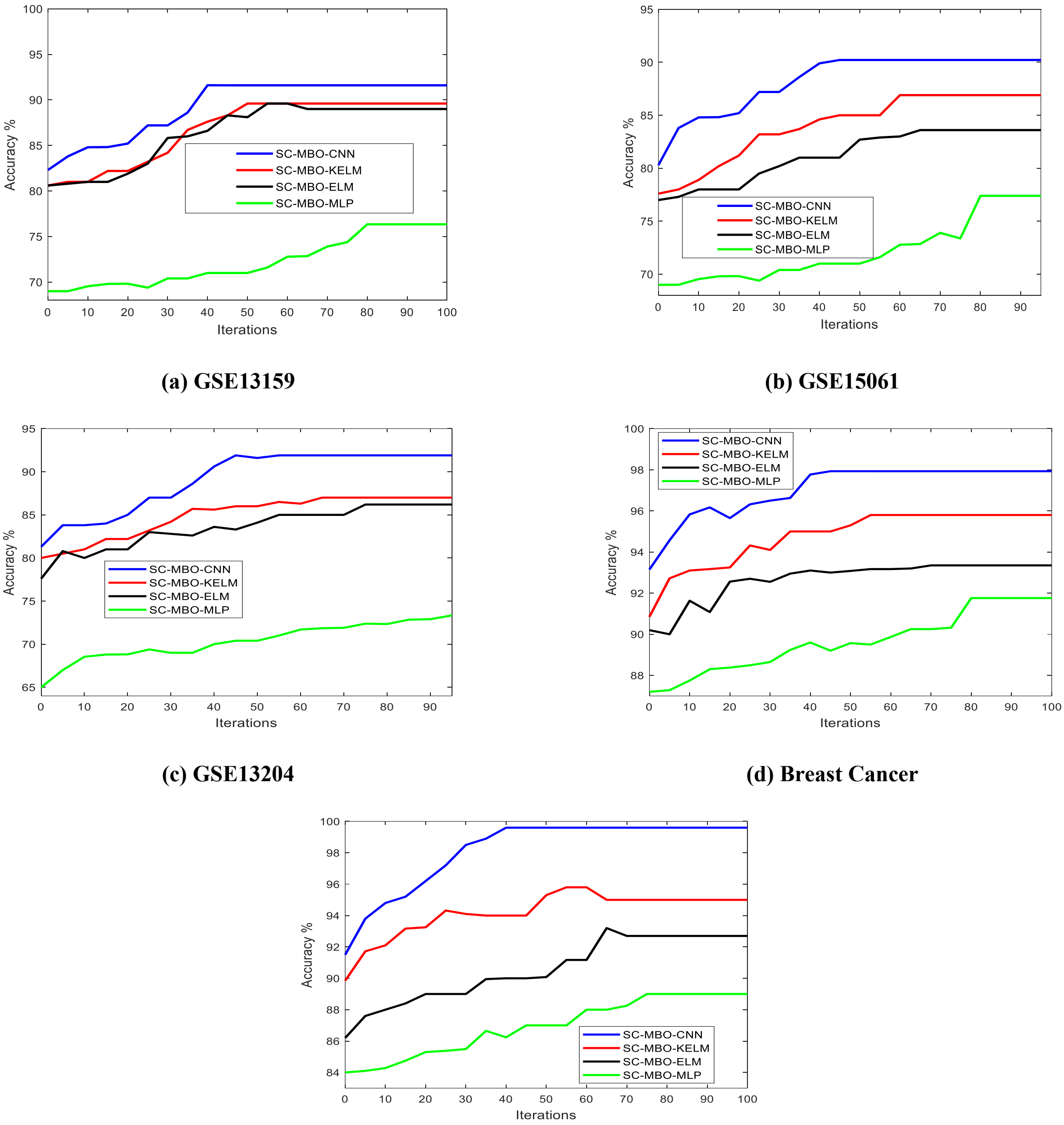
Table 8 shows the SC-MBO optimized hyperparameter values of the CNN on five high-dimensional datasets. The hyperparameter values in table 8 are rounded to four decimal places. In GSE13159 dataset, the proposed model uses 0.0008 as Learning Rate value, 0.2625 as Dropout 1, 0.2667 as Dropout 2, 100 as Batch Size and 4Layers and gives 91.5% accuracy. In GSE15061 dataset, the proposed model uses 0.0009 as Learning Rate value, 0.2234 as Dropout 1, 0.2625 as Dropout 2, 50 as Batch Size and 3Layers and gives 90.22% accuracy. In GSE13204 dataset, the proposed model uses 0.0015 as learning rate value, 0.2667 as dropout 1, 0.4233 as dropout 2, 150 as batch size and 4Layers and gives 91.9% accuracy. In breast cancer dataset, the proposed model uses 0.0005 as learning rate value, 0.2667 as dropout 1, 0.4367 as dropout 2, 100 as batch size and 3Layers and yields 97.33% accuracy. In ovarian cancer dataset, the proposed model uses 0.0013 as learning rate value, 0.1668 as dropout 1, 0.4327 as dropout 2, 250 as batch size and 4 layers and gives 99.6% accuracy.

Tab 8. SC-MBO optimized CNN Hyper-parameters values in 5 datasets

Datasets	Learning rate	Dropout 1	Dropout 2	Batch size	Layer	Acc%
GSE13159	.0008	0.2625	0.2667	100	4	91.5
GSE15061	.0009	0.2234	0.2625	50	3	90.22
GSE13204	.0015	0.2667	0.4233	150	4	91.9
Breast cancer	.0005	0.2667	0.4367	100	3	97.93
Ovarian Cancer	.0013	0.1668	0.4327	250	4	99.6

In this experimental analysis, Fig. 5. shows the converging frame of the of the Sine Cosine ensemble MBO optimized CNN, SineCosine ensemble MBO optimized KELM, SineCosine ensemble MBO optimized ELM, and SineCosine ensemble MBO optimized MLP. methods on 5 high-dimensional cancer datasets. These plots show how accuracy improved over 100 iterations on five datasets. GSE13159 Accuracy of the data set is converging near the 39th, 50th, 64th and 79th iteration in SineCosine ensemble MBO optimized CNN, SineCosine ensemble MBO optimized KELM, SineCosine ensemble MBO optimized ELM, and SineCosine ensemble MBO optimized MLP methods in Fig. 5(a), respectively.

The accuracy of the GSE15061 dataset converges near the 48th, 59th, 65th and 80th iteration in the SineCosine ensembled MBO optimized CNN, SineCosine ensembled MBO optimized KELM, SineCosine ensembled MBO optimized ELM, and SineCosine ensembled MBO optimized MLP methods in Fig. 5(b), respectively. The accuracy of the GSE13204 dataset converges at the 54th, 66th, 75th and after the 100th iteration in the SineCosine ensembled MBO optimized CNN, SineCosine ensemble MBO optimized KELM, SineCosine ensemble MBO optimized ELM, and SineCosine ensemble MBO optimized MLP methods in Fig. 5(c), respectively.



(a) GSE13159

(b) GSE15061

(c) GSE13204

(d) Breast Cancer

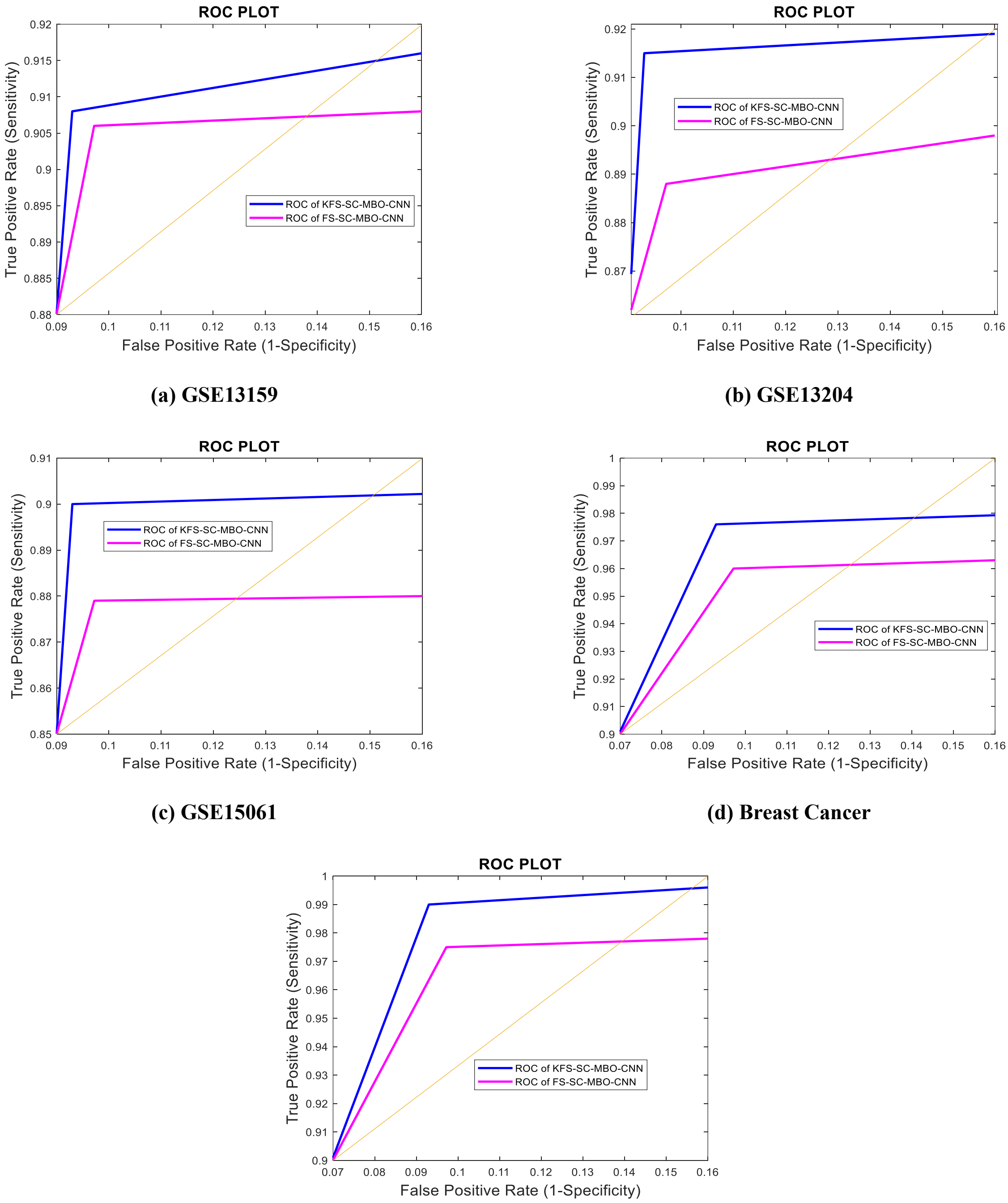
(e) Ovarian Cancer

Fig. 5. Convergence graph of five high dimensional cancerous datasets

The accuracy of the breast cancer dataset converges at the 43rd, 56th, 72nd and 78th iteration in the SineCosine ensemble MBO optimized CNN, SineCosine ensemble MBO optimized KELM, SineCosine ensemble MBO optimized ELM, and SineCosine ensemble MBO optimized MLP methods in Fig. 5(d), respectively. The accuracy of the ovarian dataset converges near the 40th, 61st, 67th and 73rd iteration in the sinecosine ensemble MBO-optimized CNN, sinecosine ensemble MBO-optimized KELM, sinecosine ensemble MBO-optimized ELM, and sinecosine ensemble MBO-optimized MLP methods in Fig. 5(e), respectively. The above convergence graphs clearly show that the SC ensemble MBO optimized CNN method converges significantly faster than other methods. This accelerated convergence can be achieved by integrating Sine Cosine (SC) into the Monarch Butterfly algorithm.

In addition, figure 6 shows the Receiver Operating Curves (ROC) between the specificity and sensitivity values observed from the KFScore-SC-MBO-CNN and FS-SC-MBO-CNN methods on five cancer datasets. ROCcurves illustrate the trade-off between true positive rate (TPR) and false positive rate (FPR) at different threshold settings, which is useful in high-dimensional data where optimal thresholds are not apparent.

Based on the results shown in figure 6, it's clear that the proposed ensemble model, KFScore-SC-MBO-CNN), outperforms other approaches.



(a) GSE13159

(b) GSE13204

(c) GSE15061

(d) Breast Cancer

(e) Ovarian Cancer

Fig. 6. ROC graph of five high dimensional cancerous datasets

Table 9 presents a clear view that the proposed KFScore based SC-MBO-CNN outperforms in GSE13159 (91.6% accuracy), GSE13204 (91.9% accuracy) and breast cancer (97.93% accuracy). However, in the Ovarian Cancer dataset, the Fisher Score applied WCGWO-Mr based PNN method gives an accuracy of 100%, but the KFScore SC-MBO-CNN model gives 99.6% accuracy with a smaller number of extracted significant attributes.

Tab. 9. A qualitative comparison among the presented ensembled method with other recent methods (The '-' sign indicates missing data.)

Methods	Datasets				
	GSE13159	GSE13204	GSE15061	Breast Cancer	Ovarian cancer
Friedman-mrKNN (Kumar et al., 2022)	79.5 (17593)	84.8 (1225)	73.1 (54318)	-	-
Friedman-mrPSVM (Kumar & Rath, 2015)	80.4 (6000)	-	83.4 (4000)	-	-
ReliefF-WCGWO-mrPNN (Baliarsingh et al., 2020)	83.78 (300)	84.28 (300)	65.51 (700)	80.00 (20)	99.2 (200)
InceptionV3 (Radhakrishnan et al., 2024)	-	-	-	-	97.96
KW-mrKNN (Kumar et al., 2022)	80.3 (36897)	83.9 (1427)	73.4 (9741)	-	-
Decision Tree, (Botlagunta et al., 2023)	-	-	-	83	-
Fisher score- WCGWO-mrPNN (Baliarsingh et al., 2020)	87.56 (500)	65.51 (700)	90.07 (300)	88.88 (20)	100 (150)
Optimized stacking ensemble learning (Kumar, et al., 2022)	-	-	-	99.45	-
KW-mrPSVM (Kumar & Rath, 2015)	81.0 (15000)	-	–83.4 (6000)	72.7 (90)	-
ANOVA-mrKNN (Baliarsingh et al., 2020)	80.8 (37016)	83.8 (1423)	71.7 (6786)	72.7 (30)	-
ANOVA-mrPSVM (Baliarsingh et al., 2020)	81.1 (14000)	-	84.1 (5000)	-	-
KFS-SC-MBO-CNN	91.6 (500)	91.9 (300)	90.22 (150)	97.93 (20)	99.6 (100)

7. CONCLUSIONS

In this experimental research work, we have presented an ensemble deep learning approach to deal with high-dimensional cancer datasets. The presented model is characterized by the following specific objectives:

(i) At the outset, the KFScore algorithm is used to select key genomes. A key advantage of using KFS is that it removes insignificant genes from the high-dimensional input feature space by transforming the data set using a kernel function.

(ii) In this study, we used the SC ensemble MBO approach to optimize the random variables of the CNN. A significant advantage of this algorithm lies in its reduced computational complexity and time.

Moreover, the presented strategy is evaluated by its accuracy, number of extracted prominent genomes, sensitivity, specificity and ROC curve. The effectiveness of the proposed methods is also compared with that of other existing models. According to the experimental evaluation, the proposed scheme is reliable, accurate and robust. Consequently, the presented approach can be interpreted as a trustworthy basis for the analysis of large genomic data.

Author Contributions

1. *Prajna Paramita Debata: Written Proposed Algorithm 1-4 and Implementation of proposed model.*
2. *Alakananda Tripathy: Literature Review.*
3. *Pournamasi Parhi: Written Introduction part.*
4. *Smruti Rekha Das: Prepared Fig. 1-3.*

Conflicts of Interest

The authors declare that there is no conflict of interest for this manuscript.

REFERENCES

Alshamlan, H. M., Badr, G. H., & Alohali, Y. A. (2015). Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification. *Computational Biology and Chemistry*, *56*, 49–60. https://doi.org/10.1016/j.compbiolchem.2015.03.001

Ang, J. C., Mirzal, A., Haron, H., & Hamed, H. N. A. (2015). Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *13*(5), 971–989. https://doi.org/10.1109/tcbb.2015.2478454

Aziz, R., Verma, C., & Srivastava, N. (2016). A fuzzy based feature selection from independent component subspace for machine learning classification of microarray data. *Genomics Data*, *8*, 4–15. https://doi.org/10.1016/j.gdata.2016.02.012

Baliarsingh, S. K., Vipsita, S., Gandomi, A. H., Panda, A., Bakshi, S., & Ramasubbareddy, S. (2020). Analysis of high-dimensional genomic data using MapReduce based probabilistic neural network. *Computer Methods and Programs in Biomedicine*, *195*, 105625. https://doi.org/10.1016/j.cmpb.2020.105625

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

Botlagunta, M., Botlagunta, M. D., Myneni, M. B., Lakshmi, D., Nayyar, A., Gullapalli, J. S., & Shah, M. A. (2023). Classification and diagnostic prediction of breast cancer metastasis on clinical data using machine learning algorithms. *Scientific Reports*, *13*(1), 485. https://doi.org/10.1038/s41598-023-27548-w

Chen, Y.-W., & Lin, C.-J. (2006). Combining SVMs with various feature selection strategies. In I. Guyon, M. Nikravesh, S. Gunn, & L. A. Zadeh (Eds.), *Feature Extraction* (Vol. 207, pp. 315–324). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-35488-8_13

Debata, P. P., & Mohapatra, P. (2022). Selection of informative genes from high-dimensional cancerous data employing an improvised meta-heuristic algorithm. *Evolutionary Intelligence*, *15*, 1841-1859. https://doi.org/10.1007/s12065-021-00593-y

Díaz-Uriarte, R., & De Andrés, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, *7*, 3. https://doi.org/10.1186/1471-2105-7-3

Kilicarslan, S., Adem, K., & Celik, M. (2020). Diagnosis and classification of cancer using hybrid model based on ReliefF and convolutional neural network. *Medical Hypotheses*, *137*, 109577. https://doi.org/10.1016/j.mehy.2020.109577

Kumar, M., & Rath, S. K. (2015). Classification of microarray using MapReduce based proximal support vector machine classifier. *Knowledge-Based Systems*, *89*, 584–602. https://doi.org/10.1016/j.knosys.2015.09.005

Kumar, M., Singhal, S., Shekhar, S., Sharma, B., & Srivastava, G. (2022). Optimized stacking ensemble learning model for breast cancer detection and classification using machine learning. *Sustainability*, *14*(21), 13998. https://doi.org/10.3390/su142113998

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436-444. https://doi.org/10.1038/nature14539

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

Liu, J., Wang, X., Cheng, Y., & Zhang, L. (2017). Tumor gene expression data classification via sample expansion-based deep learning. *Oncotarget*, *8*(65), 109646–109660. https://doi.org/10.18632/oncotarget.22762

Ludwig, S. A., Jakobovic, D., & Picek, S. (2015). Analyzing gene expression data: Fuzzy decision tree algorithm applied to the classification of cancer data. *2015 IEEE international conference on fuzzy systems (FUZZ-IEEE)* (pp. 1-8). IEEE. https://doi.org/10.1109/FUZZ-IEEE.2015.7337854

Medjahed, S. A., Saadi, T. A., Benyettou, A., & Ouali, M. (2017). Kernel-based learning and feature selection analysis for cancer diagnosis. *Applied Soft Computing*, *51*, 39-48. https://doi.org/10.1016/j.asoc.2016.12.010

Mohapatra, P., Chakravarty, S., & Dash, P. (2016). Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system. *Swarm and Evolutionary Computation*, *28*, 144–160. https://doi.org/10.1016/j.swevo.2016.02.002

Panda, M. (2017). Elephant search optimization combined with deep neural network for microarray data analysis. *Journal of King Saud University - Computer and Information Sciences*, *32*(8), 940–948. https://doi.org/10.1016/j.jksuci.2017.12.002

Polat, K., & Güneş, S. (2009). A new feature selection method on classification of medical datasets: Kernel F-score feature selection. *Expert Systems With Applications*, *36*(7), 10367–10373. https://doi.org/10.1016/j.eswa.2009.01.041

Radhakrishnan, M., Sampathila, N., Muralikrishna, H., & Swathi, K. S. (2024). Advancing ovarian cancer diagnosis through deep learning and eXplainable AI: A multiclassification approach. *IEEE Access*, *12*, 116968-116986. https://doi.org/10.1109/ACCESS.2024.3448219

Sharma, S., Saha, A. K., Roy, S., Mirjalili, S., & Nama, S. (2022). A mixed sine cosine butterfly optimization algorithm for global optimization and its application. *Cluster Computing*, *25*, 4573-4600. https://doi.org/10.1007/s10586-022-03649-5

Shilaskar, S., Ghatol, A., & Chatur, P. (2017). Medical decision support system for extremely imbalanced datasets. *Information Sciences*, *384*, 205–219. https://doi.org/10.1016/j.ins.2016.08.077

Wang, G. G., Deb, S., & Cui, Z. (2019a). Monarch butterfly optimization. *Neural computing and Applications*, *31*, 1995-2014. https://doi.org/10.1007/s00521-015-1923-y

Wang, Y., Yang, X.-G., & Lu, Y. (2019b). Informative gene selection for microarray classification via adaptive elastic net with conditional mutual information. *Applied Mathematical Modelling*, *71*, 286-297. https://doi.org/10.1016/j.apm.2019.01.044

Yoo, J.-H., Yoon, H.-I., Kim, H.-G., Yoon, H.-S., & Han, S.-S. (2019). Optimization of hyper-parameter for CNN model using genetic algorithm. *2019 1st International conference on electrical, control and instrumentation engineering (ICECIE)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICECIE47765.2019.8974762

Zeebaree, D. Q., Haron, H., & Abdulazeez, A. M. (2018). Gene selection and classification of microarray data using convolutional neural network. *2018 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 145-150). IEEE. https://doi.org/10.1109/ICOASE.2018.8548836